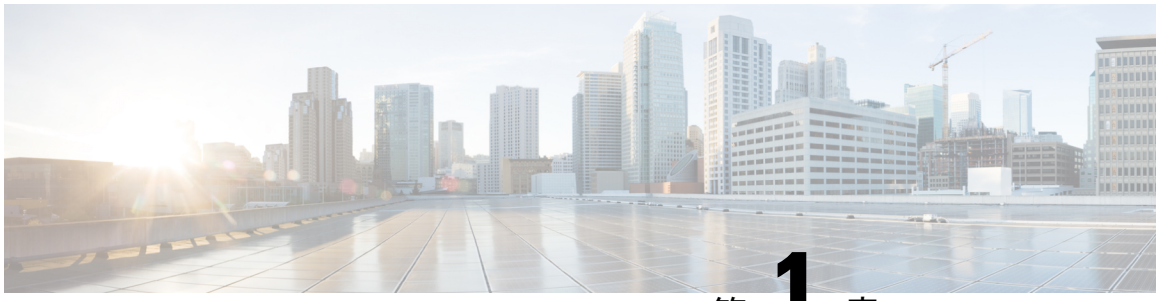
Cisco IOS XE Gibraltar 16.12.x（Catalyst 9300 スイッチ）IP ルーティング コンフィギュレーション ガイド

初版：2019 年 7 月 31 日

最終更新：2019 年 7 月 29 日

シスコシステムズ合同会社
〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー
http://www.cisco.com/jp
お問い合わせ先：シスコ コンタクトセンター
0120-092-255 （フリーコール、携帯・PHS含む）
電話受付時間：平日 10:00～12:00、13:00～17:00
http://www.cisco.com/jp/go/contactcenter/

【注意】シスコ製品をご使用になる前に、安全上の注意（**www.cisco.com/jp/go/safety_warning/**）をご確認ください。本書は、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。



第 1 章

双方向フォワーディング検出の設定

双方向フォワーディング検出

このマニュアルでは、双方向フォワーディング検出（BFD）プロトコルを有効にする方法について説明します。BFD はあらゆるメディア タイプ、カプセル化、トポロジ、およびルーティング プロトコルの高速転送パス障害検出時間を提供するように設計された検出プロトコルです。

BFDは高速転送パス障害検出に加えて、ネットワーク管理者向けの整合性のある障害検出方法を提供します。ネットワーク管理者はBFDを使用して、ルーティングプロコル毎に異なるhelloメカニズムの多様な検出時間でなく、一定の検出時間で転送パスの障害を検出できるため、ネットワーク プロファイリングおよびプランニングが容易になります。また、再コンバージェンス時間の整合性が保たれ、予測可能になります。

双方向フォワーディング検出の前提条件

- Cisco Express Forwarding および IP ルーティングが、関連するすべてのスイッチで有効になっている必要があります。
- BFD をスイッチに展開する前に、BFD でサポートされている IP ルーティングプロトコルのいずれかを設定する必要があります。使用しているルーティング プロトコルの高速コンバージェンスを実装する必要があります。高速コンバージェンスの設定については、お使いのバージョンの Cisco IOS ソフトウェアの IP ルーティングのマニュアルを参照してください。Cisco IOS ソフトウェアの BFD ルーティングプロトコルのサポートの詳細については、「双方向フォワーディング検出の制約事項」の項を参照してください。

双方向フォワーディング検出の制約事項

- BFD は直接接続されたネイバーだけに対して動作します。BFD のネイバーは 1 ホップ以内に限られます。BFD はマルチホップ設定をサポートしていません。

- プラットフォームおよびインターフェイスによっては、BFDサポートを利用できないものがあります。特定のプラットフォームまたはインターフェイスで BFD がサポートされているかどうか確認し、プラットフォームとハードウェアの正確な制約事項を入手するには、お使いのソフトウェアバージョンの Cisco IOS ソフトウェアのリリースノートを参照してください。
- 自己生成パケットの QoS ポリシーは BFD パケットと一致しません。
- **class class-default** コマンドは BFD パケットと一致します。そのため、適切な帯域幅の可用性を確認して、オーバーサブスクリプションによる BFD パケットのドロップを防ぐ必要があります。
- BFD HA はサポートされていません。

双方向フォワーディング検出について

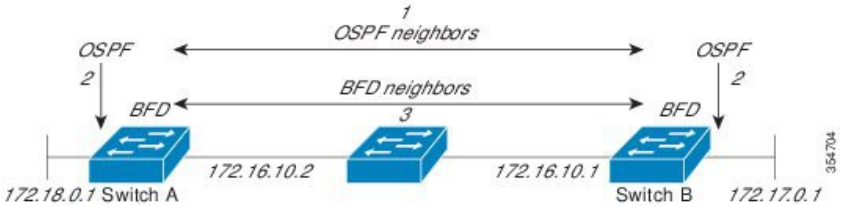
BFD の動作

BFD は、2 つの隣接デバイス間の転送パスで、オーバーヘッドの少ない短期間の障害検出方法を提供します。これらのデバイスには、インターフェイス、データリンク、および転送プレーンが含まれます。

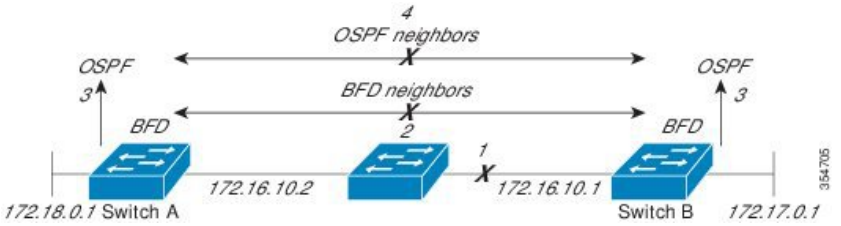
BFD はインターフェイス レベルおよびルーティング プロトコル レベルで有効にする検出プロトコルです。シスコでは、BFD 非同期モードをサポートしています。BFD 非同期モードは、デバイス間の BFD ネイバー セッションをアクティブにして維持するための、2 台のシステム間の BFD 制御パケットの送信に依存します。したがって、BFD セッションを作成するには、両方のシステム（または BFD ピア）で BFD を設定する必要があります。BFD が適切なルーティングプロトコルに対してインターフェイスおよびデバイスレベルで有効になると、BFD セッションが作成されます。BFD タイマーがネゴシエーションされ、BFD ピアはネゴシエーションされた間隔で BFD 制御パケットの相互送信を開始します。

ネイバー関係

BFDは、高速BFDピア障害検出時間を個別に提供します。これは、すべてのメディアタイプ、カプセル化、トポロジ、ルーティングプロトコル（BGP、EIGRP、IS-IS、OSPF など）から独立しています。BFDは、ローカルルータのルーティングプロトコルに高速障害検出通知を送信して、ルーティングテーブル再計算プロセスを開始します。これにより BFD は、ネットワーク コンバージェンス時間全体を大幅に短縮できます。下の図に、OSPF と BFD を実行する 2 台のルータがある単純なネットワークを示します。OSPFがネイバー（1）を検出すると、ローカル BFD プロセスに要求を送信します。OSPF ネイバールータとの BFD ネイバーセッションが開始されます（2）。OSPF ネイバー ルータでの BFD ネイバー セッションが確立されます（3）。



以下の図に、ネットワークで障害が発生した場合を示します（1）。OSPF ネイバー ルータでの BFD ネイバー セッションが停止されます（2）。BFD はローカル OSPF プロセスに BFD ネイバーに接続できなくなったことを通知します（3）。ローカル OSPF プロセスは OSPF ネイバー関係を解除します（4）。代替パスが使用可能な場合、ルータはただちにそのパスでコンバージェンスを開始します。



ルーティングプロトコルは、取得したネイバーそれぞれについて、BFDに登録する必要があります。ネイバーが登録されると、セッションがまだ存在していない場合、BFDによって、ネイバーとのセッションが開始されます。

次のとき、OSPF では、BFD を使用して登録が行われます。

- ネイバーの有限状態マシン（FSM）は、Full ステートに移行します。
- OSPF BFD と BFD の両方が有効にされます。

ブロードキャスト インターフェイスでは、OSPF によって、指定ルータ（DR）とバックアップ指定ルータ（BDR）とともにのみ、BFD セッションが確立されます。セッションは、DROTHER ステートのすべての 2 台のルータ間では確立されません

BFD の障害検出

BFD セッションが確立され、タイマー否定が完了すると、BFD ピアは BFD 制御パケットを送信します。パケットは、より高速なレートである点を除き、IGP hello プロトコルと同じように動作して活性を検出します。次の点に注意する必要があります。

- BFD はフォワーディング パスの障害検出プロトコルです。BFD は障害を検出しますが、ルーティングプロトコルが障害が発生したピアをバイパスするように機能する必要があります。
- Cisco IOS XE Denali 16.3.1 以降、シスコ デバイスは BFD バージョン 0 をサポートしています。実装では、デバイスが複数のクライアントプロトコルに 1 つの BFD セッションを使用します。たとえば、同じピアへの同じリンクを介してネットワークで OSPF および EIGRP を実行している場合、1 つの BFD セッションだけが確立されます。BFD は両方のルーティングプロトコルとセッション情報を共有します。

BFD バージョンの相互運用性

デフォルトでは、すべての BFD セッションがバージョン 1 で実行され、バージョン 0 と相互運用可能です。システムで自動的に FD バージョン検出が実行される場合、ネイバー間の BFD セッションがネイバー間の最も一般的な BFD バージョンで実行されます。たとえば、BFD ネイバーが BFD バージョン 0 を実行し、他の BFD ネイバーがバージョン 1 を実行している場合、セッションで BFD バージョン 0 が実行されます。**show bfd neighbors** [**details**] コマンドの出力で、BFD ネイバーが実行している BFD バージョンを確認できます。

BFDバージョンの検出の例については、エコーモードがデフォルトで有効になったEIGRPネットワークでの BFD の設定の例を参照してください。

BFD セッション数の上限値

Cisco IOS XE Denali 16.3.1 以降、作成できる BFD セッションの数が 100 に増えました。

非ブロードキャスト メディア インターフェイスに対する BFD サポート

Cisco IOS XE Denali 16.3.1 以降、BFD 機能は、ルーテッド SVI と L3 ポートチャネルでサポートされます。**bfd interval** コマンドは、BFD モニタリングを開始するインターフェイスで設定する必要があります。

ステートフル スイッチオーバーでのノンストップ フォワーディングの BFD サポート

通常、ネットワーキング デバイスを再起動すると、そのデバイスのすべてのルーティング ピアがデバイスの終了および再起動を検出します。この遷移によってルーティング フラップが発生し、そのために複数のルーティング ドメインに分散される可能性があります。ルーティングの再起動によって発生したルーティングフラップによって、ルーティングが不安定になります。これはネットワーク全体のパフォーマンスに悪影響を及ぼします。ノンストップ フォワーディング（NSF）は、ステートフル スイッチオーバー（SSO）が有効になっているデバイスのルーティングフラップを抑制するのに役立ち、そのためネットワークの不安定さが減少します。

NSF では、ルーティングプロトコル情報がスイッチオーバー後に保存されるとき、既知のルータでデータパケットのフォワーディングを継続できます。NSF を使用すると、ピア ネットワーキング デバイスでルーティング フラップが発生しません。データ トラフィックはインテリジェント ラインカードまたはデュアル フォワーディング プロセッサを介して転送されますが、スタンバイ RP では、スイッチオーバー中に障害が発生したアクティブな RP からの制御と見なされます。NSF の動作の重要な点の 1 つは、ラインカードとフォワーディングプロセッサがスイッチオーバー中も稼働状態を維持できることです。これらは、アクティブ RP の転送情報ベース（FIB）で最新の状態を維持します。

デュアル RP をサポートするデバイスでは、SSO が RP の 1 つをアクティブなプロセッサとして確立し、他の RP はスタンバイプロセッサに割り当てられます。SSO は、アクティブプロセッサとスタンバイプロセッサの間で情報を同期します。アクティブ RP に障害が発生したとき、アクティブ RP がネットワーキングデバイスから削除されたとき、またはメンテナンスのために手動で停止されたときに、アクティブプロセッサからスタンバイプロセッサへのスイッチオーバーが発生します。

ステートフル スイッチオーバーの BFD サポート

BFD プロトコルでは、隣接するフォワーディング エンジン間でパスに短期間の障害検出が行われます。デュアル RP ルータまたはスイッチ（冗長性のため）を使用するネットワーク導入では、ルータにグレースフル リスタート メカニズムがあります。このメカニズムは、アクティブな RP とスタンバイ RP の間のスイッチオーバー時にフォワーディング状態を保護します。

ハードウェアの通信障害を検出する機能に応じて、デュアル RP のスイッチオーバー回数が異なります。BFD が RP で稼働している場合、一部のプラットフォームでは BFD プロトコルがタイムアウトになる前にスイッチオーバーを検出することはできません。このようなプラットフォームは低速スイッチオーバー プラットフォームと呼ばれます。

スタティック ルーティングの BFD サポート

OSPF や BGP などの動的なルーティング プロトコルとは異なり、スタティック ルーティングにはピア検出の方法がありません。したがって、BFDが設定されると、ゲートウェイの到達可能性は指定されたネイバーへの BFD セッションの状態に依存します。BFD セッションが開始されない限り、スタティックルートのゲートウェイは到達不能で、影響を受けるルートが適切なルーティング情報ベース（RIB）にインストールされません。

BFD セッションを正常に確立するには、ピアのインターフェイスで BFD を設定する必要があります。BFD ネイバーのアドレスのピアに BFD クライアントが登録されている必要があります。インターフェイスがダイナミック ルーティング プロトコルで使用される場合、後者の要件は、BFD の各ネイバーでルーティング プロトコル インスタンスを設定することによって満たされます。インターフェイスがスタティック ルーティングに排他的に使用される場合、この要件はピア上でスタティック ルートを設定することによって満たす必要があります。

BFD セッションが起動状態のときに BFD 設定がリモート ピアから削除された場合、BFD セッションの最新状態が IPv4 スタティックに送信されません。その結果、スタティックルートが RIB に残ります。唯一の回避策は、IPv4 スタティック BFD ネイバー設定を削除して、スタティック ルートが BFD セッション状態を追跡しないようにすることです。また、シリアル インターフェイスのカプセル化のタイプを BFD でサポートされていないタイプに変更する場合、このインターフェイスで BFD がダウン状態になります。回避策はインターフェイスをシャットダウンし、サポートされているカプセル化のタイプに変更してから、BFDを再設定することです。

IPv4 スタティック クライアントでは 1 つの BFD セッションを使用して、特定のインターフェイスを通るネクスト ホップの到達可能性を追跡できます。一連の BFD 追跡対象スタティック ルートに対して BFD グループを割り当てることができます。各グループには 1 つのアクティブスタティック BFD 設定、1 つ以上のパッシブ BFD 構成、および対応する BFD 追跡対象スタティックルートが必要です。nongroupエントリは、BFD グループが割り当てられていない BFD 追跡対象スタティック ルートです。BFD グループは、さまざまな VRF の一部として構成可能なスタティック BFD 設定に対応する必要があります。実際には、パッシブ スタティック BFD 設定は、アクティブな設定と同じ VRF に構成する必要はありません。

BFD グループごとに存在するアクティブなスタティック BFD セッションは 1 つだけです。スタティック BFD 設定とその BFD 設定を使用する対応のスタティック ルートを追加して、アクティブBFDセッションを設定できます。アクティブなスタティック BFD構成とそのスタティック BFD 設定を使用するスタティック ルートがある場合にのみ、グループの BFD セッションが

作成されます。アクティブなスタティック BFD 設定またはアクティブなスタティック ルートが BFD グループから削除されると、パッシブなスタティック ルートがすべて RIB から削除されます。実際には、すべてのパッシブなスタティック ルートは、アクティブなスタティック BFD 設定と、アクティブな BFD セッションで追跡されるスタティック ルートがグループで設定されるまでは非アクティブです。

同様に、BFD グループごとに 1 つ以上のパッシブなスタティック BFD 設定と、対応する BFD 追跡対象スタティック ルートが存在します。パッシブなスタティック セッション ルートは、アクティブな BFD セッション状態が到達可能であるときだけ有効です。グループのアクティブな BFD セッション状態が到達可能であっても、対応するインターフェイスの状態がアップである場合にのみ、パッシブなスタティック ルートが RIB に追加されます。パッシブな BFD セッションがグループから削除されると、アクティブな BFD セッション（存在する場合）や BFD グループの到達可能性ステータスには影響しません。

障害検出に BFD を使用することの利点

機能を導入するときは、あらゆる代替策を検討し、トレードオフに注意することが重要です。

通常の導入で BFD に最も近い代替策は、EIGRP、IS-IS、および OSPF ルーティングプロトコルの障害検出メカニズムを修正することです。

EIGRP の hello およびホールド タイマーを絶対最小値に設定する場合、EIGRP の障害検出速度が 1～2 秒程度に下がります。IS-IS または OSPF などの Interior Gateway Protocol（IGP）プロトコルに fast hello を使用する場合、これらによって障害検出メカニズムが最小 1 秒に減少します。

BFD を実装する方が、ルーティングプロトコルのタイマー値を減らすよりも、いくつかの点で優れています。

- EIGRP、IS-IS、および OSPF タイマーによって 1 秒または 2 秒の最小検出タイマーを実現できますが、障害検出が 1 秒未満になる場合もあります。
- BFD は特定のルーティング プロトコルに関連付けられていないため、EIGRP、IS-IS、および OSPF の汎用の整合性のある障害検出メカニズムとして使用できます。
- BFD の一部をデータ プレーンに分散できるため、コントロール プレーンに全体が存在する分散 EIGRP、IS-IS、および OSPF タイマーよりも CPU の負荷を軽くすることができます。

双方向フォワーディング検出の設定方法

インターフェイスでの BFD セッション パラメータの設定

インターフェイスで BFD を設定するには、BFD セッションの基本パラメータを設定する必要があります。BFD ネイバーに対して BFD セッションを実行するインターフェイスごとに、この手順を繰り返します。

次の手順は、物理インターフェイスの BFD 設定手順を示しています。SVI とイーサチャネルにそれぞれ対応する BFD タイマー値を使用してください。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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	次のいずれかの手順を実行します。
	• **ip address** *ipv4-address mask*
	• **ipv6 address** *ipv6-address*/*mask*
	例：
	インターフェイスの IPv4 アドレスの設定：
	`Device(config-if)#ip address 10.201.201.1 255.255.255.0`
	インターフェイスの IPv6 アドレスの設定：
	`Device(config-if)#ipv6 address 2001:db8:1:1::1/32`
		インターフェイスに IP アドレスを設定します。

ステップ 4	**bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier* 例： `Device(config-if)#bfd interval 100 min_rx 100 multiplier 3`	インターフェイスで BFD を有効にします。 BFD interval 設定は、それを設定したサブインターフェイスが削除されたときに削除されます。 BFD interval 設定は次のような場合には削除されません。 • IPv4 アドレスがインターフェイスから削除された場合 • IPv6 アドレスがインターフェイスから削除された場合 • IPv6 がインターフェイスで無効にされた場合 • インターフェイスがシャットダウンされた場合 • インターフェイスで IPv4 CEF がグローバルまたはローカルで無効にされた場合 • インターフェイスで IPv6 CEF がグローバルまたはローカルで無効にされた場合

	コマンドまたはアクション	目的
ステップ 5	**end** 例： `Device(config-if)#end`	インターフェイス 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。

ダイナミック ルーティング プロトコルに対する BFD サポートの設定

次のセクションでは、ダイナミック ルーティング プロトコルの BFD サポートに関する設定について説明します。

eBGP に対する BFD サポートの設定

ここでは、BGP の BFD サポートを設定する手順について説明します。これにより、BGP が BFD の登録プロトコルとなり、BFD から転送パスの検出障害メッセージを受信できます。

始める前に

eBGP は、関連するすべてのルータで実行する必要があります。

BFD セッションを BFD ネイバーに対して実行するインターフェイスで、BFD セッションの基本パラメータを設定します。詳細については、「インターフェイスでの BFD セッション パラメータの設定」の項を参照してください。

（注） **show bfd neighbors details** コマンドの出力には、設定された間隔が表示されます。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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** *as-tag* 例： `Device(config)#router bgp tag1`	BGP プロセスを指定し、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* **fall-over bfd** 例： `Device(config-router)#neighbor 172.16.10.2 fall-over bfd`	フェールオーバーに対するBFDサポートを有効にします。
ステップ 5	**end** 例： `Device(config-router)#end`	ルータ コンフィギュレーション モードを終了して、特権 EXEC モードに戻ります。
ステップ 6	**show bfd neighbors** [**details**] 例： `Device#show bfd neighbors detail`	（任意）BFDネイバーがアクティブで、BFDが登録したルーティングプロトコルが表示されることを確認します。
ステップ 7	**show ip bgp neighbor** 例： `Device#show ip bgp neighbor`	（任意）ネイバーへの BGP および TCP 接続についての情報を表示します。

EIGRP に対する BFD サポートの設定

ここでは、EIGRP の BFD サポートを設定する手順について説明します。これにより、EIGRP が BFD の登録プロトコルとなり、BFD から転送パスの検出障害メッセージを受信できます。EIGRP に対する BFD サポートを有効にするには、2 つの方法があります。

- ルータ コンフィギュレーション モードで **bfd all-interfaces** コマンドを使用して、EIGRP がルーティングしているすべてのインターフェイスに対して BFD を有効にできます。
- ルータ設定モードで **bfd interface** *type number* コマンドを使用して、EIGRP がルーティングしているインターフェイスのサブセットに対して BFD を有効にできます。

始める前に

- EIGRP は、関連するすべてのルータで実行する必要があります。
- BFD セッションを BFD ネイバーに対して実行するインターフェイスで、BFD セッションの基本パラメータを設定します。詳細については、「インターフェイスでのBFDセッションパラメータの設定」の項を参照してください。

（注） **show bfd neighbors details** コマンドの出力には、設定された間隔が表示されます。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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 eigrp** *as-number* 例： `Device(config)#router eigrp 123`	EIGRP ルーティング プロセスを設定し、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	次のいずれかを実行します。
	• **bfd all-interfaces**
	• **bfd interface** *type number*
	例：
	`Device(config-router)#bfd all-interfaces`
	例：
	`Device(config-router)#bfd interface GigabitFastEthernet 1/0/1`
		EIGRP ルーティングプロセスに関連付けられたすべてのインターフェイスで、BFD をグローバルに有効にします。
		または
		EIGRP ルーティングプロセスに関連付けられた 1 つ以上のインターフェイスに対して、インターフェイスごとに BFD を有効にします。

ステップ 5	**end** 例： `Device(config-router) end`	ルータ コンフィギュレーションモードを終了して、特権 EXEC モードに戻ります。
ステップ 6	**show bfd neighbors** [**details**] 例： `Device#show bfd neighbors details`	（任意）BFD ネイバーがアクティブで、BFD が登録したルーティング プロトコルが表示されることを確認します。
ステップ 7	**show ip eigrp interfaces** [*type number*] [*as-number*] [**detail**] 例： `Device#show ip eigrp interfaces detail`	（任意）EIGRP に対する BFD サポートが有効になっているインターフェイスを表示します。

IS-IS に対する BFD サポートの設定

ここでは、IS-IS が BFD の登録プロトコルとなり、BFD から転送パスの検出障害メッセージを受信するように、IS-IS に対する BFD サポートを設定する手順について説明します。IS-IS に対する BFD サポートをイネーブルにするには、2 つの方法があります。

- ルータ コンフィギュレーション モードで **bfd all-interfaces** コマンドを使用して、IS-IS が IPv4 ルーティングをサポートしているすべてのインターフェイスに対して BFD を有効にできます。次にインターフェイス コンフィギュレーション モードで **isis bfd disable** コマンドを使用すると、1 つ以上のインターフェイスに対して BFD を無効にできます。
- インターフェイス コンフィギュレーション モードで **isis bfd** コマンドを使用すると、IS-IS がルーティングしているインターフェイスのサブセットに対して BFD を有効にできます。

IS-IS に対する BFD サポートを設定するには、次のいずれかの手順に従います。

前提条件

- IS-IS は、関連するすべてのデバイスで実行する必要があります。
- BFD セッションを BFD ネイバーに対して実行するインターフェイスで、BFD セッションの基本パラメータを設定する必要があります。詳細については、「インターフェイスでの BFD セッションパラメータの設定」の項を参照してください。

（注） **show bfd neighbors details** コマンドの出力には、設定された間隔が表示されます。ハードウェア オフロードされた BFD セッションが 50 ms の倍数でない Tx および Rx 間隔で設定されていたために変更された間隔は出力に表示されません。

すべてのインターフェイスの *IS-IS* に対する *BFD* サポートの設定

IPv4 ルーティングをサポートするすべての IS-IS インターフェイスで BFD を設定するには、この項の手順に従います。

手順の概要

1. **enable**
2. **configure terminal**
3. **router isis** *area-tag*
4. **bfd all-interfaces**
5. **exit**
6. **interface** *type number*
7. **ip router isis** [*tag*]
8. **isis bfd** [**disable**]
9. **end**
10. **show bfd neighbors** [**details**]
11. **show clns interface**

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： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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 isis** *area-tag* 例： `Device(config)#router isis tag1`	IS-IS プロセスを指定し、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	**bfd all-interfaces** 例： `Device(config-router)#bfd all-interfaces`	IS-IS ルーティング プロセスに関連付けられたすべてのインターフェイスで、BFD をグローバルにイネーブルにします。
ステップ 5	**exit** 例： `Device(config-router)#exit`	（任意）ルータでグローバル コンフィギュレーション モードに戻ります。
ステップ 6	**interface** *type number* 例： `Device(config)#interface fastethernet 6/0`	（任意）インターフェイス コンフィギュレーション モードを開始します。
ステップ 7	**ip router isis** [*tag*] 例： `Device(config-if)#ip router isis tag1`	（任意）インターフェイスで IPv4 ルーティングのサポートをイネーブルにします。
ステップ 8	**isis bfd** [**disable**] 例： `Device(config-if)#isis bfd`	（任意）IS-IS ルーティング プロセスに関連付けられた 1 つ以上のインターフェイスに対して、インターフェイスごとに BFD をイネーブルまたはディセーブルにします。

	コマンドまたはアクション	目的
		（注） コンフィギュレーション モードで **bfd all-interfaces** コマンドを使用して IS-IS が関連付けられたすべてのインターフェイスで以前に BFD を有効にしていた場合にのみ、**disable** キーワードを使用する必要があります。
ステップ 9	**end** 例： `Device(config-if)#end`	インターフェイス コンフィギュレーション モードを終了して、特権 EXEC モードに戻ります。
ステップ 10	**show bfd neighbors** [**details**] 例： `Device#show bfd neighbors details`	（任意）BFD ネイバーがアクティブで、BFD が登録したルーティング プロトコルが表示されるかどうかの検証に使用できる情報を表示します。
ステップ 11	**show clns interface** 例： `Device#show clns interface`	（任意）IS-IS に対する BFD が、関連付けられた特定の IS-IS インターフェイスに対してイネーブルになっているかどうかを検証するために使用できる情報を表示します。

1 つ以上のインターフェイスの *IS-IS* に対する *BFD* サポートの設定

1 つ以上の IS-IS インターフェイスだけに BFD を設定するには、この項の手順に従います。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip router isis** [*tag*]
5. **isis bfd [disable]**
6. **end**
7. **show bfd neighbors** [**details**]
8. **show clns interface**

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： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	**configure terminal** 例： `Device#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	**interface** *type number* 例： `Device(config)#interface fastethernet 6/0`	インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**ip router isis** [*tag*] 例： `Device(config-if)#ip router isis tag1`	インターフェイスで IPv4 ルーティングのサポートをイネーブルにします。
ステップ 5	**isis bfd [disable]** 例： `Device(config-if)#isis bfd`	IS-IS ルーティング プロセスに関連付けられた 1 つ以上のインターフェイスに対して、インターフェイスごとに BFD をイネーブルまたはディセーブルにします。 （注） コンフィギュレーション モードで **bfd all-interfaces** コマンドを使用して IS-IS が関連付けられたすべてのインターフェイスで BFD を有効にした場合にのみ、**disable** キーワードを使用する必要があります。
ステップ 6	**end** 例： `Device(config-if)#end`	インターフェイス コンフィギュレーション モードを終了して、特権 EXEC モードに戻ります。
ステップ 7	**show bfd neighbors** [**details**] 例： `Device#show bfd neighbors details`	（任意）BFD ネイバーがアクティブで、BFD が登録したルーティング プロトコルが表示されるかどうかの検証に使用できる情報を表示します。
ステップ 8	**show clns interface** 例： `Device#show clns interface`	（任意）IS-IS に対する BFD が、関連付けられた特定の IS-IS インターフェイスに対してイネーブルになっているかどうかを検証するために使用できる情報を表示します。

OSPF に対する BFD サポートの設定

ここでは、OSPF が BFD の登録プロトコルとなり、BFD から転送パスの検出障害メッセージを受信するように、OSPF に対する BFD サポートを設定する手順について説明します。すべてのインターフェイスでグローバルに OSPF に対する BFD を設定するか、または 1 つ以上のインターフェイスで選択的に設定することができます。

OSPF に対する BFD サポートを有効にするには、2 つの方法があります。

- ルータ コンフィギュレーション モードで **bfd all-interfaces** コマンドを使用して、OSPF がルーティングしているすべてのインターフェイスに対して BFD を有効にできます。インターフェイス コンフィギュレーション モードで **ip ospf bfd** [**disable**] コマンドを使用して、個々のインターフェイスで BFD サポートを無効にできます。
- インターフェイス コンフィギュレーション モードで **ip ospf bfd** コマンドを使用すると、OSPF がルーティングしているインターフェイスのサブセットに対して BFD を有効にできます。

OSPF に対する BFD サポートのタスクについては、次の項を参照してください。

すべてのインターフェイスの *OSPF* に対する *BFD* サポートの設定

すべての OSPF インターフェイスに BFD を設定するには、この項の手順に従います。

すべての OSPF インターフェイスに対して BFD を設定するのではなく、特定の 1 つ以上のインターフェイスに対して BFD サポートを設定する場合は、「1 つ以上のインターフェイスの OSPF に対する BFD サポートの設定」の項を参照してください。

始める前に

- OSPF は、関連するすべてのルータで実行する必要があります。
- BFD セッションを BFD ネイバーに対して実行するインターフェイスで、BFD セッションの基本パラメータを設定する必要があります。詳細については、「インターフェイスでの BFD セッションパラメータの設定」の項を参照してください。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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。

	コマンドまたはアクション	目的
ステップ 3	**router ospf** *process-id* 例： `Device(config)#router ospf 4`	OSPF プロセスを指定し、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	**bfd all-interfaces** 例： `Device(config-router)#bfd all-interfaces`	OSPF ルーティング プロセスに関連付けられたすべてのインターフェイスで、BFD をグローバルに有効にします。
ステップ 5	**exit** 例： `Device(config-router)#exit`	（任意）デバイスでグローバル コンフィギュレーション モードに戻ります。ステップ 7 を実行して 1 つ以上のインターフェイスに対して BFD を無効にする場合にだけ、このコマンドを入力します。
ステップ 6	**interface** *type number* 例： `Device(config)#interface fastethernet 6/0`	（任意）インターフェイス コンフィギュレーション モードを開始します。ステップ 7 を実行して 1 つ以上のインターフェイスに対して BFD を無効にする場合にだけ、このコマンドを入力します。
ステップ 7	**ip ospf bfd** [**disable**] 例： `Device(config-if)#ip ospf bfd disable`	（任意）OSPF ルーティング プロセスに関連付けられた 1 つ以上のインターフェイスに対して、インターフェイスごとに BFD を無効にします。 （注） コンフィギュレーション モードで **bfd all-interfaces** コマンドを使用して OSPF が関連付けられたすべてのインターフェイスで BFD を有効にした場合にのみ、**disable** キーワードを使用する必要があります。
ステップ 8	**end** 例： `Device(config-if)#end`	インターフェイス コンフィギュレーション モードを終了して、特権 EXEC モードに戻ります。
ステップ 9	**show bfd neighbors** [**details**] 例： `Device#show bfd neighbors detail`	（任意）BFD ネイバーがアクティブで、BFD が登録したルーティング プロトコルが表示されるかどうかの検証に使用できる情報を表示します。
ステップ 10	**show ip ospf** 例： `Device#show ip ospf`	（任意）OSPF に対して BFD が有効になっているかどうかを検証するために使用できる情報を表示します。

1 つ以上のインターフェイスの *BFD over IPv4* に対する *OSPF* サポートの設定

1 つ以上の OSPF インターフェイスで BFD を設定するには、この項の手順に従います。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip ospf bfd** [**disable**]
5. **end**
6. **show bfd neighbors** [**details**]
7. **show ip ospf**

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： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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	**interface** *type number* 例： `Device(config)#interface fastethernet 6/0`	インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**ip ospf bfd** [**disable**] 例： `Device(config-if)#ip ospf bfd`	OSPF ルーティング プロセスに関連付けられた 1 つ以上のインターフェイスに対して、インターフェイスごとに BFD を有効または無効にします。 （注） ルータ コンフィギュレーション モードで **bfd all-interfaces** コマンドを使用して OSPF が関連付けられたすべてのインターフェイスで BFD を有効にした場合にのみ、**disable** キーワードを使用します。
ステップ 5	**end** 例： `Device(config-if)#end`	インターフェイス コンフィギュレーション モードを終了して、デバイスが特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
ステップ 6	**show bfd neighbors** [**details**] 例： `Device#show bfd neighbors details`	（任意）BFD ネイバーがアクティブで、BFD が登録したルーティング プロトコルが表示されるかどうかの検証に使用できる情報を表示します。 （注） ハードウェア オフロードされた BFD セッションが、50 ms の倍数でない Tx および Rx 間隔で設定されると、ハードウェア間隔が変更されます。ただし、**show bfd neighbors details** コマンドの出力には、変更された間隔ではなく、設定された間隔値のみが表示されます。
ステップ 7	**show ip ospf** 例： `Device#show ip ospf`	（任意）OSPF に対して BFD サポートが有効になっているかどうかを検証するために使用できる情報を表示します。

HSRP に対する BFD サポートの設定

ホットスタンバイ ルータ プロトコル（HSRP）の BFD サポートをイネーブルにするには、次の作業を実行します。この手順のステップは、HSRP ピアに BFD セッションを実行する各インターフェイスで行ってください。

デフォルトでは、HSRP は BFD をサポートします。BFD に対する HSRP サポートが手動でディセーブルになっている場合、ルータ レベルで再びイネーブルにして、すべてのインターフェイスに対してグローバルに BFD サポートをイネーブルにするか、またはインターフェイス レベルでインターフェイスごとにイネーブルにすることができます。

始める前に

- HSRP は、関連するすべてのルータで実行する必要があります。
- シスコ エクスプレス フォワーディングをイネーブルにする必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例：	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
	`Device#configure terminal`	
ステップ 3	**ip cef** [**distributed**] 例： `Device(config)#ip cef`	シスコ エクスプレス フォワーディングまたは分散型シスコ エクスプレス フォワーディングをイネーブルにします。
ステップ 4	**interface** *type number* 例： `Device(config)#interface FastEthernet 6/0`	インターフェイス コンフィギュレーション モードを開始します。
ステップ 5	**ip address** *ip-address mask* 例： `Device(config-if)#ip address 10.1.0.22 255.255.0.0`	インターフェイスに IP アドレスを設定します。
ステップ 6	**standby** [*group-number*] **ip** [*ip-address* [**secondary**]] 例： `Device(config-if)#standby 1 ip 10.0.0.11`	HSRP をアクティブにします。
ステップ 7	**standby bfd** 例： `Device(config-if)#standby bfd`	（任意）インターフェイスで BFD に対する HSRP をイネーブルにします。
ステップ 8	**exit** 例： `Device(config-if)#exit`	インターフェイス コンフィギュレーション モードを終了します。
ステップ 9	**standby bfd all-interfaces** 例： `Device(config)#standby bfd all-interfaces`	（任意）すべてのインターフェイスで BFD に対する HSRP をイネーブルにします。
ステップ 10	**exit** 例： `Device(config)#exit`	グローバル コンフィギュレーション モードを終了します。
ステップ 11	**show standby neighbors** 例：	（任意）BFD に対する HSRP サポートについての情報を表示します。

	コマンドまたはアクション	目的
	`Device#show standby neighbors`	

スタティック ルーティングに対する BFD サポートの設定

スタティック ルーティングのための BFD サポートを設定するには、このタスクを実行します。各 BFD ネイバーに対してこの手順を繰り返します。詳細については、「例：スタティック ルーティングに対する BFD サポートの設定」の項を参照してください。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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	**interface** *type number* 例： `Device(config)#interface serial 2/0`	インターフェイスを設定し、インターフェイス コンフィギュレーション モードを開始します。

ステップ 4	次のいずれかの手順を実行します。
	• **ip address** *ipv4-address mask*
	• **ipv6 address** *ipv6-address*/*mask*
	例：
	インターフェイスの IPv4 アドレスの設定：
	`Device(config-if)#ip address 10.201.201.1 255.255.255.0`
	インターフェイスの IPv6 アドレスの設定：
	`Device(config-if)#ipv6 address 2001:db8:1:1::1/32`
		インターフェイスに IP アドレスを設定します。

ステップ 5	**bfd interval** *milliseconds* **mix_rx** *milliseconds* **multiplier** *interval-multiplier* 例：	インターフェイスで BFD を有効にします。 bfd interval 設定は、それを設定したサブインターフェイスが削除されたときに削除されます。

	コマンドまたはアクション	目的
	`Device(config-if)#bfd interval 500 min_rx 500 multiplier 5`	bfd interval 設定は次のような場合には削除されません。 • IPv4 アドレスがインターフェイスから削除された場合 • IPv6 アドレスがインターフェイスから削除された場合 • IPv6 がインターフェイスからディセーブルにされた場合 • インターフェイスがシャットダウンされた場合 • インターフェイスで IPv4 CEF がグローバルまたはローカルでディセーブルにされた場合 • インターフェイスで IPv6 CEF がグローバルまたはローカルでディセーブルにされた場合
ステップ 6	**exit** 例： `Device(config-if)#exit`	インターフェイス 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 7	**ip route static bfd** *interface-type interface-number ip-address* [**group** *group-name* [**passive**]] 例： `Device(config)#ip route static bfd TenGigabitEthernet1/0/1 10.10.10.2 group group1 passive`	スタティック ルートの BFD ネイバーを指定します。 • BFD が直接接続されたネイバーだけでサポートされているため、*interface-type*、*interface-number*、および *ip-address* 引数は必須です。
ステップ 8	**ip route** [**vrf** *vrf-name*] *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [**dhcp**] [*distance*] [**name** *next-hop-name*] [**permanent** | **track** *number*] [**tag** *tag*] 例： `Device(config)#ip route 10.0.0.0 255.0.0.0`	スタティック ルートの BFD ネイバーを指定します。
ステップ 9	**exit** 例： `Device(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 モードに戻ります。

	コマンドまたはアクション	目的
ステップ 10	**show ip static route** 例： `Device#show ip static route`	（任意）スタティック ルート データベース情報を表示します。
ステップ 11	**show ip static route bfd** 例： `Device#show ip static route bfd`	（任意）設定された BFD グループおよび nongroup エントリからスタティック BFD の設定に関する情報を表示します。
ステップ 12	**exit** 例： `Device#exit`	特権 EXEC モードを終了し、ユーザー EXEC モードに戻ります。

BFD エコー モードの設定

デフォルトでは BFD エコー モードが有効になっていますが、方向ごとに個別に実行できるように、無効にすることもできます。

BFD エコー モードは非同期 BFD で動作します。エコー パケットはフォワーディング エンジンによって送信され、検出を実行するために、同じパスで転送されます。反対側の BFD セッションはエコー パケットの実際のフォワーディングに関与しません。エコー機能およびフォワーディング エンジンが検出プロセスを処理するため、2 つの BFD ネイバー間で送信される BFD 制御パケットの数が減少します。また、フォワーディング エンジンが、リモート システムを介さずにリモート（ネイバー）システムの転送パスをテストするため、パケット間の遅延のばらつきが向上する可能性があり、それによって BFD バージョン 0 を BFD セッションの BFD 制御パケットで使用する場合に、障害検出時間を短縮できます。

エコー モードを両端で実行している（両方の BFD ネイバーがエコー モードを実行している）場合は、非対称性がないと表現されます。

前提条件

- BFD は、参加しているすべてのデバイスで実行されている必要があります。
- CPU 使用率の上昇を避けるために、BFD エコーモードを使用する前に、**no ip redirects** コマンドを入力して、Internet Control Message Protocol（ICMP）リダイレクトメッセージの送信を無効にする必要があります。
- BFD セッションを BFD ネイバーに対して実行するインターフェイスで、BFD セッションの基本パラメータを設定する必要があります。詳細については、「インターフェイスでの BFD セッション パラメータの設定」の項を参照してください。

機能制限

BFDエコーモードは、ユニキャストリバースパス転送（uRPF）の設定との組み合わせでは動作しません。BFDエコーモードとuRPFの設定がイネーブルの場合、セッションはフラップします。

非対称性のない BFD エコー モードの無効化

この手順では、非対称性のない BFD エコーモードを無効化にする方法を示します。デバイスからはエコーパケットが送信されず、デバイスはネイバーデバイスから受信する BFD エコーパケットを転送しません。

各 BFD デバイスに対してこの手順を繰り返します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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	**no bfd echo** 例： `Device(config)#no bfd echo`	BFD エコー モードを無効にします。 • **no** 形式を使用すると、BFD エコーモードを無効にできます。
ステップ 4	**end** 例： `Device(config)#end`	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 モードに戻ります。

BFD テンプレートの作成と設定

シングルホップ テンプレートは一連の BFD 間隔値を指定するために設定できます。BFD テンプレートの一部として指定されるBFD間隔値は、1つのインターフェイスに限定されるものではありません。

（注） BDF テンプレートを設定すると、エコーモードが無効になります。

シングルホップ テンプレートの設定

BFD シングルホップ テンプレートを作成し、BFD インターバル タイマーを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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	**bfd-template single-hop** *template-name* 例： `Device(config)#bfd-template single-hop bfdtemplate1`	シングルホップ BFD テンプレートを作成し、BFD 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	**interval min-tx** *milliseconds* **min-rx** *milliseconds* **multiplier** *multiplier-value* 例： `Device(bfd-config)#interval min-tx 120 min-rx 100 multiplier 3`	BFD パケット間での送受信間隔を設定し、ピアが使用不能であると BFD が宣言する前に損失される連続的な BFD 制御パケット数を指定します。
ステップ 5	**end** 例： `Device(bfd-config)#end`	BFD コンフィギュレーション モードを終了し、デバイスを特権 EXEC モードに戻します。

BFD のモニタリングとトラブルシューティング

ここでは、維持とトラブルシューティングのために BFD 情報を取得する方法について説明します。これらのタスクのコマンドを必要に応じて任意の順序で入力できます。

ここでは、次の Cisco プラットフォームに対する BFD のモニタリングとトラブルシューティングについて説明します。

BFD のモニタリングとトラブルシューティング

BFD のモニタリングまたはトラブルシューティングを実行するには、この項の 1 つ以上の手順に従います。

手順の概要

1. **enable**
2. **show bfd neighbors** [**details**]
3. **debug bfd** [**packet** | **event**]

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： `Device>enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**show bfd neighbors** [**details**] 例： `Device#show bfd neighbors details`	（任意）BFD 隣接関係データベースを表示します。 **details** キーワードを指定すると、すべての BFD プロトコルパラメータとネイバーごとにタイマーが表示されます。
ステップ 3	**debug bfd** [**packet**	**event**] 例： `Device#debug bfd packet`	（任意）BFD パケットのデバッグ情報を表示します。

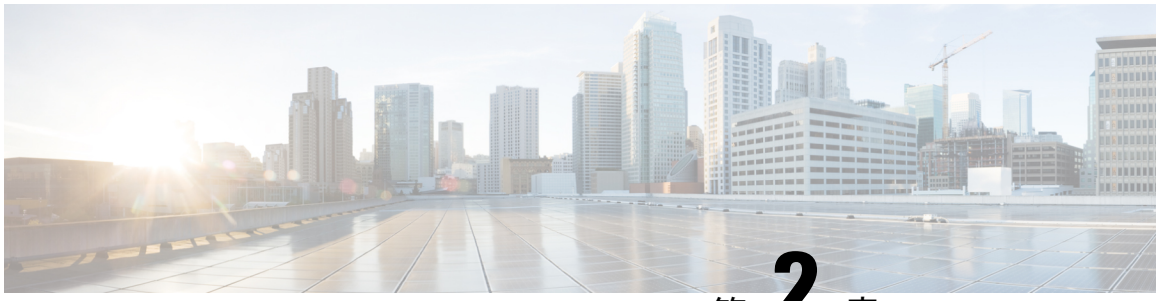
双方向フォワーディング検出の設定例

ここでは、双方向フォワーディング検出の設定例を示します。

双方向フォワーディング検出に関する機能情報

表 *1:* 双方向フォワーディング検出に関する機能情報

リリース	機能情報
Cisco IOS XE Everest 16.5.1a	この機能が導入されました



第 2 章

EIGRP IPv6 に対する BFD サポートの設定

EIGRP IPv6 に対する BFD サポートの前提条件

EIGRP IPv6 セッションには、ルータ、アドレスファミリ、およびアドレスファミリ インターフェイス コンフィギュレーション モードでのシャットダウンオプションがあります。EIGRP IPv6 セッションでの BFD サポートを有効にするには、これらのモードでルーティングプロセスを no shut モードにする必要があります。

EIGRP IPv6 に対する BFD サポートに関する制約事項

- EIGRP IPv6 に対する BFD サポートの機能は、EIGRP 名前付きモードでのみサポートされます。
- EIGRP は、シングルホップの Bidirectional Forwarding Detection（BFD）のみをサポートしています。
- EIGRP IPv6 に対する BFD サポートの機能は、パッシブインターフェイスではサポートされません。

EIGRP IPv6 に対する BFD サポートに関する情報

EIGRP IPv6 に対する BFD サポート機能は、Enhanced Interior Gateway Routing Protocol（EIGRP）IPv6 セッションに対する Bidirectional Forwarding Detection（BFD）サポートを提供します。これにより、EIGRPIPv6 トポロジでの迅速な障害検出と代替パスの選択が容易になります。BFD は、一貫した障害検出方式をネットワーク管理者に提供する検出プロトコルです。ネットワーク管理者は、BFD を使用することで、さまざまなルーティングプロトコルの「Hello」メカニズムの変動速度ではなく一定速度で転送パス障害を検出できます。この障害検出方式により、ネットワークのプロファイリングとプランニングが容易になり、再コンバージェンス時間も一貫性のある予測可能なものになります。このガイドでは、EIGRP IPv6 ネットワークの BFD サポートに関する情報を提供し、EIGRPIPv6 ネットワークで BFD サポートを設定する方法について説明します。

EIGRP IPv6 に対する BFD サポートの設定方法

ここでは、1 つのインターフェイスおよびすべてのインターフェイスでの EIGRP IPv6 に対する BFD サポートの設定について説明します。

すべてのインターフェイスでの BFD サポートの設定

次の手順は、すべてのインターフェイスで BFD サポートを設定する方法を示しています。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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	**ipv6 unicast-routing** 例： `Device(config)# ipv6 unicast-routing`	IPv6 ユニキャスト データグラムの転送を有効にします。
ステップ 4	**interface***type number* 例： `Device(config)# interface ethernet0/0`	インターフェイスのタイプと番号を指定し、インターフェイス コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 5	**ipv6 address** *ipv6-address/prefix-length* 例： `Device(config-if)# ipv6 address 2001:DB8:A:B::1/64`	IPv6 アドレスを設定します。
ステップ 6	**bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier* 例： `Device(config-if)# bfd interval 50 min_rx 50 multiplier 3`	インターフェイスのベースライン BFD セッション パラメータを設定します。
ステップ 7	**exit** 例： `Device(config-if)# exit`	インターフェイス 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 8	**router eigrp** *virtual-name* 例： `Device(config)# router eigrp name`	EIGRP ルーティング プロセスを指定し、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	**address-family ipv6 autonomous-system***as-number* 例： `Device(config-router)# address-family ipv6 autonomous-system 3`	IPv6 のアドレス ファミリ コンフィギュレーション モードを開始して、EIGRP ルーティング インスタンスを設定します。
ステップ 10	**eigrp router-id***ip-address* 例： `Device(config-router-af)# eigrp router-id 172.16.1.3`	EIGRP ピアがネイバーと通信する際に EIGRP がこのアドレスファミリに関して使用するデバイス ID を設定します。
ステップ 11	**af-interface default** 例： `Device(config-router-af)# af-interface default`	EIGRP 名前付きモード設定においてアドレスファミリに属するすべてのインターフェイスでインターフェイス固有のコマンドを設定します。アドレス ファミリ インターフェイス コンフィギュレーション モードを開始します。
ステップ 12	**bfd** 例： `Device(config-router-af-interface)# bfd`	すべてのインターフェイスでBFDを有効にします。
ステップ 13	**End** 例： `Device(config-router-af-interface)# end`	アドレスファミリ インターフェイス 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。

	コマンドまたはアクション	目的
ステップ 14	**show eigrp address-family ipv6 neighbors detail** 例： `Device# show eigrp address-family ipv6 neighbors detail`	（任意）インターフェイスで BFD が有効になっている EIGRP によって検出されたネイバーに関する詳細情報を表示します。
ステップ 15	**show bfd neighbors** 例： `Device# show bfd neighbors`	（任意）BFD 情報をネイバーに表示します。

インターフェイスでの BFD サポートの設定

次の手順は、インターフェイスで BFD サポートを設定する方法を示しています。

手順の概要

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface***type number*
5. **ipv6 address** *ipv6-address /prefix-length*
6. **bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier*
7. **exit**
8. **router eigrp** *virtual-name*
9. **address-family ipv6 autonomous-system***as-number*
10. **eigrp router-id***ip-address*
11. **af-interface** *interface-type interface-number*
12. **bfd**
13. **end**
14. **show eigrp address-family ipv6 neighbors**
15. **show bfd neighbors**

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： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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。

	コマンドまたはアクション	目的
ステップ 3	**ipv6 unicast-routing** 例： `Device(config)# ipv6 unicast-routing`	IPv6 ユニキャスト データグラムの転送を有効にします。
ステップ 4	**interface***type number* 例： `Device(config)# interface ethernet0/0`	インターフェイスのタイプと番号を指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 5	**ipv6 address** *ipv6-address /prefix-length* 例： `Device(config-if)# ipv6 address 2001:DB8:A:B::1/64`	IPv6 アドレスを設定します。
ステップ 6	**bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier* 例： `Device(config-if)# bfd interval 50 min_rx 50 multiplier 3`	インターフェイスのベースライン BFD セッション パラメータを設定します。
ステップ 7	**exit** 例： `Device(config-if)# exit`	インターフェイス 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 8	**router eigrp** *virtual-name* 例： `Device(config)# router eigrp name`	EIGRP ルーティング プロセスを指定し、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	**address-family ipv6 autonomous-system***as-number* 例： `Device(config-router)# address-family ipv6 autonomous-system 3`	IPv6 のアドレス ファミリ コンフィギュレーション モードを開始して、EIGRP ルーティング インスタンスを設定します。
ステップ 10	**eigrp router-id***ip-address* 例： `Device(config-router-af)# eigrp router-id 172.16.1.3`	EIGRP ピアがネイバーと通信する際に EIGRP がこのアドレス ファミリに関して使用するデバイス ID を設定します。
ステップ 11	**af-interface** *interface-type interface-number* 例： `Device(config-router-af)# af-interface ethernet0/0`	EIGRP 名前付きモード設定においてアドレス ファミリに属するインターフェイスでインターフェイス固有のコマンドを設定します。アドレス ファミリ インターフェイス コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 12	**bfd** 例： `Device(config-router-af-interface)# bfd`	指定されたインターフェイス上で BFD をイネーブルにします。
ステップ 13	**end** 例： `Device(config-router-af-interface)# end`	アドレスファミリ インターフェイス 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 eigrp address-family ipv6 neighbors** 例： `Device# show eigrp address-family ipv6 neighbors`	（任意）BFD が有効になっているネイバーを表示します。
ステップ 15	**show bfd neighbors** 例： `Device# show bfd neighbors`	（任意）BFD 情報をネイバーに表示します。

EIGRP IPv6 に対する BFD サポートの設定例

ここでは、EIGRP に対する BFD サポートの設定例を示します。

例：すべてのインターフェイスでの BFD サポートの設定

```
Device> enable
Device# configure terminal
Device(config)# ipv6 unicast-routing
Device(config)# interface Ethernet0/0
Device(config-if)# ipv6 address 2001:0DB8:1::12/64
Device(config-if)# bfd interval 50 min_rx 50 multiplier 3
Device(config-if)# exit
Device(config)# router eigrp name
Device(config-router)# address-family ipv6 unicast autonomous-system 1
Device(config-router-af)# eigrp router-id 172.16.0.1
Device(config-router-af)# af-interface default
Device(config-router-af-interface)# bfd
Device(config-router-af-interface)# end
```

次に、**show eigrp address-family ipv6 neighbors detail** コマンドの出力例を示します。

```
Device# show eigrp address-family ipv6 neighbors detail
EIGRP-IPv6 VR(test) Address-Family Neighbors for AS(5)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
0   Link-local address:     Et0/0                    14 00:02:04    1  4500  0  4
    FE80::10:2
   Version 23.0/2.0, Retrans: 2, Retries: 0, Prefixes: 1
   Topology-ids from peer - 0
   Topologies advertised to peer:   base

Max Nbrs: 0, Current Nbrs: 0
```

```
BFD sessions
NeighAddr          Interface
FE80::10:2             Ethernet0/0
```

次に、**show bfd neighbor** コマンドの出力例を示します。

```
Device# show bfd neighbors

IPv6 Sessions
NeighAddr                                LD/RD         RH/RS     State     Int
FE80::10:2                                2/0          Down      Down      Et0/0
```

例：インターフェイスでの BFD サポートの設定

次に、インターフェイスで BFD サポートを設定する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# ipv6 unicast-routing
Device(config)# Ethernet0/0
Device(config-if)# ipv6 address 2001:DB8:A:B::1/64
Device(config-if)# bfd interval 50 min_rx 50 multiplier 3
Device(config-if)# exit
Device(config)# router eigrp name
Device(config-router)# address-family ipv6 autonomous-system 3
Device(config-router-af)# af-interface Ethernet0/0
Device(config-router-af-interface)# bfd
Device(config-router-af-interface)# end
```

その他の参考資料

関連資料

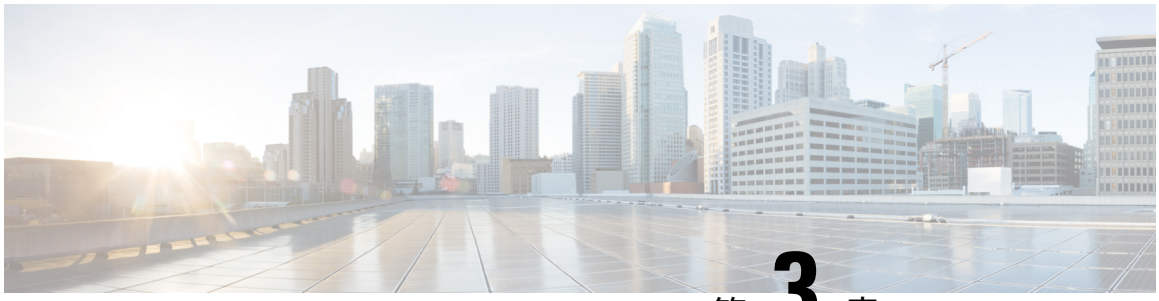
関連項目	マニュアル タイトル
BFD コマンド：コマンド構文、コマンドモード、コマンド履歴、デフォルト、使用に関する注意事項、および例。	次のドキュメントの IP ルーティングに関する項を参照してください：*Command Reference (Catalyst 9300 Series Switches)*
EIGRP コマンド：コマンド構文の詳細、コマンドモード、コマンド履歴、デフォルト設定、使用に関する注意事項、および例	次のドキュメントの IP ルーティングに関する項を参照してください：*Command Reference (Catalyst 9300 Series Switches)*
EIGRP の設定	次のドキュメントのルーティングに関する項を参照してください：*Software Configuration Guide (Catalyst 9300 Switches)*

EIGRP IPv6 に対する BFD サポートの機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。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 *2 : EIGRP IPv6* に対する *BFD* サポートの機能情報

機能名	リリース	機能情報
EIGRP IPv6 に対する BFD サポート	Cisco IOS XE Gibraltar 16.11.x	この機能が導入されました。



第 3 章

MSDP の設定

MSDP の設定について

このセクションでは、スイッチに Multicast Source Discovery Protocol（MSDP）を設定する方法について説明します。MSDP によって、複数の Protocol-Independent Multicast Sparse-Mode（PIM-SM）ドメインが接続されます。

このソフトウェア リリースでは、MSDP と連携して動作する Multicast Border Gateway Protocol（MBGP）がサポートされていないため、MSDP は完全にはサポートされていません。ただし、MBGP が動作していない場合、MSDP と連携して動作するデフォルト ピアを作成できます。

（注） この機能を使用するには、アクティブ スイッチ上で Network Advantage フィーチャ セットが稼働している必要があります。

MSDP の概要

MSDP を使用すると、さまざまなドメイン内のすべてのランデブーポイント（RP）に、グループのマルチキャスト送信元を通知できます。各 PIM-SM ドメインでは独自の RP が使用され、他のドメインの RP には依存しません。RP は伝送制御プロトコル（TCP）を通じて MSDP を実行し、他のドメイン内のマルチキャスト送信元を検出します。

PIM-SM ドメイン内の RP は、他のドメイン内の MSDP 対応デバイスと MSDP ピアリング関係にあります。ピアリング関係は TCP 接続を通じて発生します。主に、マルチキャスト グループを送信する送信元のリストを交換します。RP 間の TCP 接続は、基本的なルーティング シス

テムによって実現されます。受信側の RP では、送信元リストを使用して送信元のパスが確立されます。

このトポロジの目的は、ドメインから、他のドメイン内のマルチキャスト送信元を検出することです。マルチキャスト送信元がレシーバーのあるドメインを対象としている場合、マルチキャストデータはPIM-SMの通常の送信元ツリー構築メカニズムを通じて配信されます。MSDPは、グループを送信する送信元のアナウンスにも使用されます。これらのアナウンスは、ドメインの RP で発信する必要があります。

MSDP のドメイン間動作は、Border Gateway Protocol（BGP）または MBGP に大きく依存します。ドメイン内の RP（インターネットへのアナウンス対象であるグローバル グループを送信する送信元用の RP）で、MSDP を実行してください。

MSDP の動作

送信元が最初のマルチキャスト パケットを送信すると、送信元に直接接続された先頭ホップ ルータ（指定ルータまたは RP）によって RP に PIM 登録メッセージが送信されます。RP は登録メッセージを使用し、アクティブな送信元を登録したり、ローカル ドメイン内の共有ツリーの下方向にマルチキャスト パケットを転送します。MSDP が設定されている場合は、Source-Active（SA）メッセージも、すべての MSDP ピアに転送します。送信元、送信元からの送信先であるグループ、および RP のアドレスまたは発信元 ID（RP アドレスとして使用されるインターフェイスの IP アドレス）が設定されている場合は、SA メッセージによってこれらが識別されます。

各 MSDP ピアは SA メッセージを発信元の RP から受信して転送し、ピア Reverse-Path Forwarding（RPF）フラッディングを実現します。MSDP デバイスは、BGP または MBGP ルーティング テーブルを調べ、どのピアが SA メッセージの発信元 RP へのネクスト ホップであるかを検出します。このようなピアは *RPF* ピアと呼ばれます。MSDP デバイスでは、RPF ピア以外のすべての MSDP ピアにメッセージが転送されます。BGP および MBGP がサポートされていない場合に MSDP を設定する方法については、デフォルトの MSDP ピアの設定 （38 ページ）を参照してください。

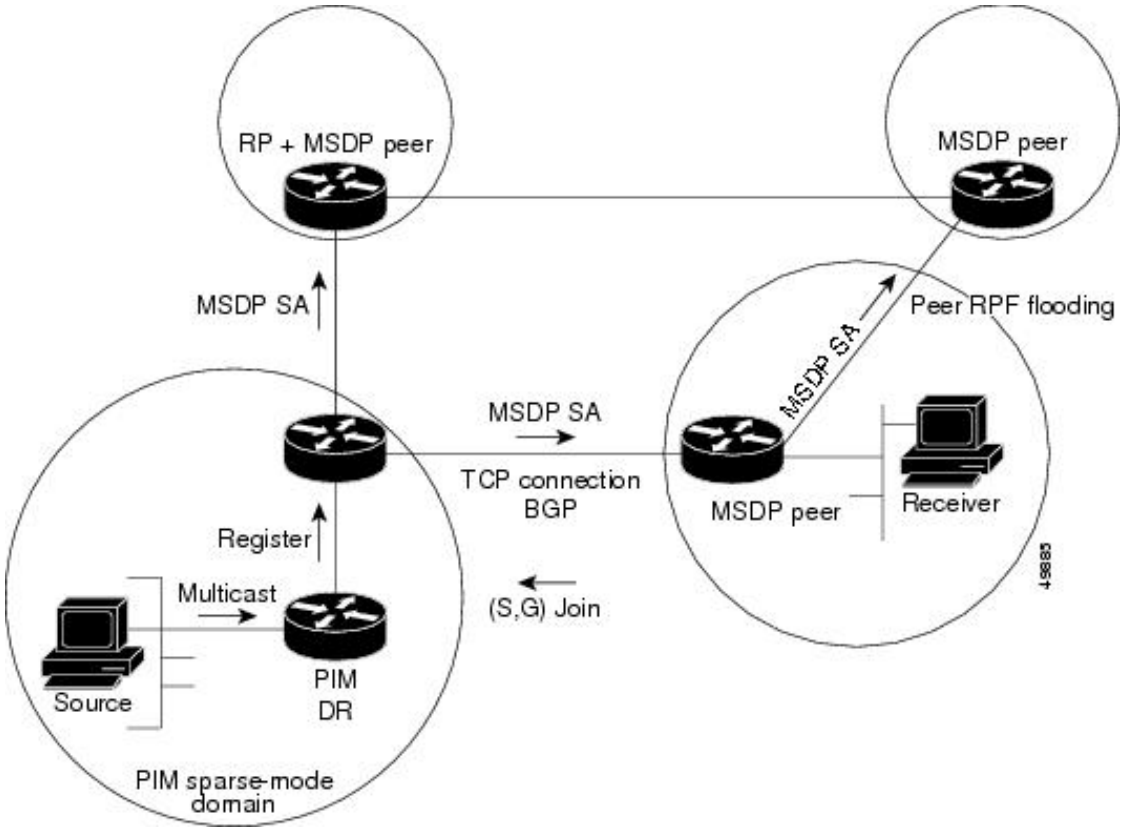
MSDP ピアは、非 RPF ピアから発信元 RP へ向かう同じ SA メッセージを受信すると、そのメッセージをドロップします。それ以外の場合、すべての MSDP ピアにメッセージが転送されます。

ドメインの RP ピアは MSDP ピアから SA メッセージを受信します。この RP が SA メッセージに記述されているグループへの加入要求を持ち、空でない発信インターフェイス リストに（*,G）エントリが含まれている場合、そのグループはドメインの対象となり、RP から送信元方向に（S,G）Join メッセージが送信されます。（S,G）Join メッセージが送信元の DR に到達してからは、送信元からリモート ドメイン内の RP への送信元ツリーのブランチが構築されています。この結果、マルチキャスト トラフィックを送信元から送信元ツリーを経由して RP へ、そしてリモート ドメイン内の共有ツリーを下ってレシーバへと送信できます。

図 *1 : RP* ピア間で動作する *MSDP*

この図に、2 つの MSDP ピアの間での MSDP の動作を示します。PIM では、ドメインの RP に送信元を登録するための標準メカニズムとして、MSDP が使用されます。MSDP が設定されて

いる場合は、次のシーケンスが発生します。



デフォルトでは、スイッチで受信されたSAメッセージ内の送信元やグループのペアは、キャッシュに格納されません。また、MSDP SA 情報が転送される場合、この情報はメモリに格納されません。したがって、ローカル RP で SA メッセージが受信された直後にメンバーがグループに加入した場合、そのメンバーは、その次の SA メッセージによって送信元に関する情報が取得されるまで、待機する必要があります。この遅延は加入遅延と呼ばれます。

ローカルRPでは、SA要求を送信し、指定されたグループに対するすべてのアクティブな送信元の要求をすぐに取得できます。デフォルトでは、新しいメンバーがグループに加入してマルチキャスト トラフィックを受信する必要が生じた場合、スイッチはMSDP ピアに SA 要求メッセージを送信しません。新しいメンバーは次の定期的な SA メッセージを受信する必要があります。

グループへの送信元である接続PIM SM ドメイン内のアクティブなマルチキャスト送信元を、グループの新しいメンバーが学習する必要がある場合は、新しいメンバーがグループに加入したときに、指定されたMSDP ピアにSA要求メッセージを送信するようにスイッチを設定します。

MSDP の利点

MSDP には次の利点があります。

- 共有されたマルチキャスト配信ツリーが分割され、共有ツリーがドメインに対してローカルになるように設定できます。ローカル メンバーはローカル ツリーに加入します。共有ツリーへの Join メッセージはドメインから脱退する必要はありません。
- PIM SM ドメインは独自の RP だけを信頼するため、他のドメインの RP に対する信頼度が低下します。このため、送信元の情報がドメイン外部に漏れないようにでき、セキュリティが高まります。
- レシーバーだけが配置されているドメインは、グループ メンバーシップをグローバルにアドバタイズしなくても、データを受信できます。
- グローバルな送信元マルチキャスト ルーティング テーブル ステートが不要になり、メモリが削減されます。

MSDP の設定方法

MSDP のデフォルト設定

MSDP はイネーブルになっていません。デフォルトの MSDP ピアはありません。

デフォルトの MSDP ピアの設定

始める前に

MSDP ピアを設定します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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 msdp default-peer** *ip-address* | *name* [**prefix-list** *list*] 例：	すべての MSDP SA メッセージの受信元となるデフォルト ピアを定義します。

	コマンドまたはアクション	目的
	`Device(config)#ip msdp default-peer 10.1.1.1 prefix-list site-a`	• *ip-address* | *name* には、MSDP デフォルト ピアの IP アドレスまたはドメイン ネーム システム（DNS）サーバー名を入力します。 • （任意） **prefix-list** *list* を指定する場合は、リスト内のプレフィックス専用のデフォルトピアとなるピアを指定するリスト名を入力します。プレフィックス リストがそれぞれ関連付けられている場合は、複数のアクティブなデフォルト ピアを設定できます。 **prefix-list** キーワードが指定された **ip msdp default-peer** コマンドを複数入力すると、複数の RP プレフィックスに対してすべてのデフォルトピアが同時に使用されます。この構文は通常、スタブ サイト クラウドに接続されたサービス プロバイダ クラウドで使用されます。 **prefix-list** キーワードを指定せずに **ip msdp default-peer** コマンドを複数入力すると、単一のアクティブピアですべての SA メッセージが受信されます。このピアに障害がある場合は、次の設定済みデフォルト ピアですべての SA メッセージが受信されます。この構文は通常、スタブ サイトで使用されます。
ステップ 4	**ip prefix-list** *name* [**description** *string*] | **seq** *number* {**permit** | **deny**} *network length* 例： `Device(config)#prefix-list site-a seq 3 permit 12 network length 128`	（任意）ステップ 2 で指定された名前を使用し、プレフィックス リストを作成します。 • （任意）**description** *string* を指定する場合は、このプレフィックスリストを説明する 80 文字以下のテキストを入力します。 • **seq** *number* には、エントリのシーケンス番号を入力します。指定できる範囲は 1 ～ 4294967294 です。 • **deny** キーワードを指定すると、条件が一致した場合にアクセスが拒否されます。 • **permit** キーワードを指定すると、条件が一致した場合にアクセスが許可されます。 • *network length* には、許可または拒否されているネットワークの番号およびネットワーク マスク長（ビット単位）を指定します。

	コマンドまたはアクション	目的
ステップ 5	**ip msdp description** {*peer-name* | *peer-address*} *text* 例： `Device(config)#ip msdp description peer-name site-b`	（任意）設定内で、または **show** コマンド出力内で簡単に識別できるように、指定されたピアの説明を設定します。 デフォルトでは、MSDP ピアに説明は関連付けられていません。
ステップ 6	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 7	**show running-config** 例： `Device#show running-config`	入力を確認します。
ステップ 8	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

SA ステートのキャッシング

メモリを消費して送信元情報の遅延を短縮する場合は、SA メッセージをキャッシュに格納するようにデバイスを設定できます。送信元とグループのペアのキャッシングをイネーブルにするには、次の手順を実行します。

送信元とグループのペアのキャッシングをイネーブルにするには、次の手順に従います。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例：	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
	`Device#configure terminal`	
ステップ 3	**ip msdp cache-sa-state** [**list** *access-list-number*] 例： `Device(config)#ip msdp cache-sa-state 100`	送信元とグループのペアのキャッシングをイネーブルにします（SA ステートを作成します）。アクセス リストを通過したこれらのペアがキャッシュに格納されます。 **list** *access-list-number* の範囲は 100 ～ 199 です。 （注） このコマンドの代わりに、**ip msdp sa-reques** グローバル コンフィギュレーション コマンドを使用できます。この代替コマンドを使用すると、グループの新しいメンバがアクティブになった場合に、SA 要求メッセージがデバイスから MSDP ピアに送信されます。
ステップ 4	**access-list** *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* 例： `Device(config)#access-list 100 permit ip 171.69.0.0 0.0.255.255 224.2.0.0 0.0.255.255`	IP拡張アクセス リストを作成します。必要な回数だけこのコマンドを繰り返します。 • *access-list-number* の範囲は 100 ～ 199 です。ステップ 2 で作成した番号と同じ値を入力します。 • **deny** キーワードは、条件が一致した場合にアクセスを拒否します。**permit** キーワードは、条件が一致した場合にアクセスを許可します。 • *protocol* には、プロトコル名として **ip** を入力します。 • *source* には、パケットの送信元であるネットワークまたはホストの番号を入力します。 • *source-wildcard* には、送信元に適用するワイルドカード ビットをドット付き 10 進表記で入力します。無視するビット位置には 1 を設定します。 • *destination* には、パケットの送信先であるネットワークまたはホストの番号を入力します。 • *destination-wildcard* には、宛先に適用するワイルドカード ビットをドット付き 10 進表記で入力します。無視するビット位置には 1 を設定します。

	コマンドまたはアクション	目的
		アクセスリストの末尾には、すべてに対する暗黙の拒否ステートメントが常に存在することに注意してください。
ステップ 5	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 6	**show running-config** 例： `Device#show running-config`	入力を確認します。
ステップ 7	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーションファイルに設定を保存します。

MSDP ピアからの送信元情報の要求

グループへの送信元である接続 PIM SM ドメイン内のアクティブなマルチキャスト送信元を、グループの新しいメンバが学習する必要がある場合は、新しいメンバがグループに加入したときに、指定された MSDP ピアに SA 要求メッセージがデバイスから送信されるようにこのタスクを実行します。ピアは SA キャッシュ内の情報に応答します。ピアにキャッシュが設定されていない場合、このコマンドを実行しても何も起こりません。この機能を設定すると加入遅延は短縮されますが、メモリが消費されます。

新しいメンバがグループに加入し、マルチキャストトラフィックを受信する必要が生じた場合、MSDP ピアに SA 要求メッセージを送信するようにデバイスを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	**configure terminal** 例： `Device#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 msdp sa-request** {*ip-address* | *name*} 例： `Device(config)#ip msdp sa-request 171.69.1.1`	指定された MSDP ピアに SA 要求メッセージを送信するようにデバイスを設定します。 *ip-address* | *name* を指定する場合は、グループの新しいメンバーがアクティブになるときにローカルデバイスの SA メッセージの要求元になる MSDP ピアの IP アドレス、または名前を入力します。 SA メッセージを送信する必要がある MSDP ピアごとに、このコマンドを繰り返します。
ステップ 4	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `Device#show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

スイッチから発信される送信元情報の制御

デバイスから発信されるマルチキャスト送信元情報を制御できます。

- アドバタイズ対象の送信元（送信元ベース）
- 送信元情報のレシーバー（要求元認識ベース）

詳細については、送信元の再配信 （44 ページ）およびSA 要求メッセージのフィルタリング（46 ページ）を参照してください。

送信元の再配信

SA メッセージは、送信元が登録されている RP で発信されます。デフォルトでは、RP に登録されているすべての送信元がアドバタイズされます。送信元が登録されている場合は、RP に *A* フラグが設定されています。このフラグは、フィルタリングされる場合を除き、送信元が SA に格納されてアドバタイズされることを意味します。

アドバタイズされる登録済みの送信元をさらに制限するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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 msdp redistribute** [**list** *access-list-name*] [**asn** *aspath-access-list-number*] [**route-map** *map*] 例： `Device(config)#ip msdp redistribute list 21`	SA メッセージに格納されてアドバタイズされる、マルチキャスト ルーティング テーブル内の（S, G）エントリを設定します。 デフォルトでは、ローカル ドメイン内の送信元だけがアドバタイズされます。 • （任意） **list** *access-list-name*：IP 標準または IP 拡張アクセスリストの名前または番号を入力します。標準アクセス リストの範囲は 1 ～ 99、拡張アクセス リストの範囲は 100 ～ 199 です。アクセスリストによって、アドバタイズされるローカルな送信元、および送信されるグループが制御されます。 • （任意） **asn** *aspath-access-list-number*：1 ～ 199 の範囲の IP 標準または IP 拡張アクセスリスト番号を入力します。このアクセスリスト番号は、**ip as-path access-list** コマンドでも設定する必要があります。 • （任意） **route-map** *map*：1 ～ 199 の範囲の IP 標準または IP 拡張アクセスリスト番号を入力します。このアクセスリスト番号は、**ip as-path**

	コマンドまたはアクション	目的
		access-list コマンドでも設定する必要があります。 アクセスリストまたは自律システムパスアクセスリストに従って、デバイスが（S, G）ペアをアドバタイズします。
ステップ 4	次のいずれかを使用します。 • `access-list`*access-list-number* {`deny`|`permit`} *source* [*source-wildcard*] • `access-list`*access-list-number* {`deny`|`permit`} *protocol source source-wildcard destination destination-wildcard* **例**： `Device(config)#`**`access list 21 permit 194.1.22.0`** または `Device(config)#`**`access list 21 permit ip 194.1.22.0 1.1.1.1 194.3.44.0 1.1.1.1`**	IP 標準アクセスリストを作成します。必要な回数だけこのコマンドを繰り返します。 または IP 拡張アクセスリストを作成します。必要な回数だけこのコマンドを繰り返します。 • *access-list-number*：ステップ 2 で作成した同じ番号を入力します。標準アクセスリストの範囲は 1 ～ 99、拡張アクセス リストの範囲は 100 ～ 199 です。 • **deny**：条件に合致している場合、アクセスを拒否します。**permit** キーワードは、条件が一致した場合にアクセスを許可します。 • *protocol*：プロトコル名として **ip** を入力します。 • *source*：パケットの送信元であるネットワークまたはホストの番号を入力します。 • *source-wildcard*：送信元に適用されるワイルドカード ビットをドット付き 10 進表記で入力します。無視するビット位置には 1 を設定します。 • *destination*：パケットの宛先であるネットワークまたはホストの番号を入力します。 • *destination-wildcard*：宛先に適用されるワイルドカード ビットをドット付き 10 進表記で入力します。無視するビット位置には 1 を設定します。 アクセス リストの末尾には、すべてに対する暗黙の拒否ステートメントが常に存在することに注意してください。
ステップ 5	**end** **例**：	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
	`Device(config)#end`	
ステップ 6	**show running-config** 例： `Device#show running-config`	入力を確認します。
ステップ 7	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

SA 要求メッセージのフィルタリング

デフォルトでは、SA情報をキャッシングしているデバイスだけが、SA要求に応答できます。このようなデバイスでは、デフォルトでMSDP ピアからのすべてのSA 要求メッセージが採用され、アクティブな送信元の IP アドレスが取得されます。

ただし、MSDP ピアからの SA 要求をすべて無視するように、デバイスを設定できます。標準アクセス リストに記述されたグループのピアからのSA要求メッセージだけを採用することもできます。アクセス リスト内のグループが指定された場合は、そのグループのピアからの SA 要求メッセージが受信されます。他のグループのピアからの他のメッセージは、すべて無視されます。

デフォルト設定に戻すには、 **no ip msdp filter-sa-request** {*ip-address*| *name*} グローバル コンフィギュレーション コマンドを使用します。

これらのオプションのいずれかを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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。

	コマンドまたはアクション	目的
ステップ 3	次のいずれかを使用します。 • `ip msdp filter-sa-request` {*ip-addressname*} • `ip msdp filter-sa-request` {*ip-addressname*} `list` *access-list-number* 例： `Device(config)#`**`ip msdp filter sa-request 171.69.2.2`**	指定された MSDP ピアからの SA 要求メッセージをすべてフィルタリングします。 または 標準アクセスリストを通過したグループに対して、指定された MSDP ピアからの SA 要求メッセージをフィルタリングします。アクセスリストには、複数のグループ アドレスが記述されています。 access-list-number の範囲は 1 ～ 99 です。
ステップ 4	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] 例： `Device(config)#`**`access-list 1 permit 192.4.22.0 0.0.0.255`**	IP 標準アクセスリストを作成します。必要な回数だけこのコマンドを繰り返します。 • *access-list-number* の範囲は 1 ～ 99 です。 • **deny** キーワードは、条件が一致した場合にアクセスを拒否します。**permit** キーワードは、条件が一致した場合にアクセスを許可します。 • *source* には、パケットの送信元であるネットワークまたはホストの番号を入力します。 •（任意）*source-wildcard* には、source に適用されるワイルドカード ビットをドット付き 10 進表記で入力します。無視するビット位置には 1 を設定します。 アクセス リストの末尾には、すべてに対する暗黙の拒否ステートメントが常に存在することに注意してください。
ステップ 5	**end** 例： `Device(config)#`**`end`**	特権 EXEC モードに戻ります。
ステップ 6	**show running-config** 例： `Device#`**`show running-config`**	入力を確認します。
ステップ 7	**copy running-config startup-config** 例：	（任意）コンフィギュレーション ファイルに設定を保存します。

	コマンドまたはアクション	目的
	`Device#copy running-config startup-config`	

スイッチで転送される送信元情報の制御

デフォルトでは、デバイスで受信されたすべての SA メッセージが、すべての MSDP ピアに転送されます。ただし、フィルタリングするか、または存続可能時間（TTL）値を設定し、発信メッセージがピアに転送されないようにできます。

フィルタの使用法

フィルタを作成すると、次のいずれかの処理を実行できます。

- すべての送信元とグループのペアのフィルタリング
- 特定の送信元とグループのペアだけが通過するように、IP 拡張アクセス リストを指定
- ルート マップの一致条件に基づくフィルタリング

フィルタを適用するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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	次のいずれかを使用します。 • **ip msdp sa-filter out** {*ip-address name*} • **ip msdp sa-filter out** {*ip-address name*} `list` *access-list-number* • **ip msdp sa-filter out**	• 指定された MSDP ピアへの SA メッセージをフィルタリングします。 • 指定したピアに対する IP 拡張アクセス リストを通過した SA メッセージのみを渡します。拡張アクセスリスト番号の範囲は 100 ～ 199 です。 **list** と **route-map** の両方のキーワードを使用すると、すべての条件に一致しなければ、発信 SA

	コマンドまたはアクション	目的

	{*ip-address name*}
	`route-map` *map-tag*
	例：
	`Device(config)#ip msdp sa-filter out switch.cisco.com`
	または
	`Device(config)#ip msdp sa-filter out list 100`
	または
	`Device(config)#ip msdp sa-filter out switch.cisco.com route-map 22`
		メッセージ内のいずれの（S,G）ペアも通過できません。
		• 指定された MSDP ピアへのルート マップ *map-tag* で一致基準を満たす SA メッセージのみを渡します。
		すべての一致基準に当てはまる場合、ルート マップの **permit** がフィルタを通してルートを通過します。**deny** はルートをフィルタ処理します。

ステップ 4	**access-list** *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* 例： `Device(config)#access list 100 permit ip 194.1.22.0 1.1.1.1 194.3.44.0 1.1.1.1`	（任意）IP 拡張アクセス リストを作成します。必要な回数だけこのコマンドを繰り返します。 • *access-list-number* には、ステップ 2 で指定した番号を入力します。 • **deny** キーワードは、条件が一致した場合にアクセスを拒否します。**permit** キーワードは、条件が一致した場合にアクセスを許可します。 • *protocol* には、プロトコル名として **ip** を入力します。 • *source* には、パケットの送信元であるネットワークまたはホストの番号を入力します。 • *source-wildcard* には、送信元に適用するワイルドカード ビットをドット付き 10 進表記で入力します。無視するビット位置には 1 を設定します。 • *destination* には、パケットの送信先であるネットワークまたはホストの番号を入力します。 • *destination-wildcard* には、宛先に適用するワイルドカード ビットをドット付き 10 進表記で入力します。無視するビット位置には 1 を設定します。 アクセス リストの末尾には、すべてに対する暗黙の拒否ステートメントが常に存在することに注意してください。

	コマンドまたはアクション	目的
ステップ 5	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 6	**show running-config** 例： `Device#show running-config`	入力を確認します。
ステップ 7	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

SA メッセージに格納されて送信されるマルチキャスト データの TTL による制限

TTL 値を使用して、各送信元の最初の SA メッセージにカプセル化されるデータを制御できます。IP ヘッダー TTL 値が *ttl* 引数以上であるマルチキャスト パケットだけが、指定された MSDP ピアに送信されます。たとえば、内部トラフィックの TTL 値を 8 に制限できます。他のグループを外部に送信する場合は、これらのパケットの TTL を 8 より大きく設定して送信する必要があります。

TTL しきい値を確立するには、次の手順に従います。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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。

	コマンドまたはアクション	目的
ステップ 3	**ip msdp ttl-threshold** {*ip-address* | *name*} *ttl* 例： `Device(config)#ip msdp ttl-threshold switch.cisco.com 0`	指定された MSDP ピア宛ての最初の SA メッセージにカプセル化されるマルチキャストデータを制限します。 • *ip-address*|*name* には、TTL の制限が適用される MSDP ピアの IP アドレスまたは名前を入力します。 • *ttl* には、TTL 値を入力します。デフォルトは 0 です。この場合、すべてのマルチキャストデータ パケットは、TTL がなくなるまでピアに転送されます。指定できる範囲は 0 ～ 255 です。
ステップ 4	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `Device#show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

スイッチで受信される送信元情報の制御

デフォルトでは、デバイスは、MSDP の RPF ピアによって送信されたすべての SA メッセージを受信します。ただし、着信 SA メッセージをフィルタリングし、MSDP ピアから受信する送信元情報を制御できます。つまり、特定の着信 SA メッセージを受信しないようにデバイスを設定できます。

次のいずれかの処理を実行できます。

- MSDP ピアからのすべての着信 SA メッセージのフィルタリング
- 特定の送信元とグループのペアが通過するように、IP 拡張アクセス リストを指定
- ルート マップの一致条件に基づくフィルタリング

フィルタを適用するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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	次のいずれかを使用します。
	• **ip msdp sa-filter in** {*ip-address name*}
	• **ip msdp sa-filter in** {*ip-address name*} `list` *access-list-number*
	• **ip msdp sa-filter in** {*ip-address name*} `route-map` *map-tag*
	例：
	`Device(config)#ip msdp sa-filter in switch.cisco.com`
	または
	`Device(config)#ip msdp sa-filter in list 100`
	または
	`Device(config)#ip msdp sa-filter in switch.cisco.com route-map 22`
		• 指定された MSDP ピアへの SA メッセージをフィルタリングします。
		• IP 拡張アクセス リストを通過する、指定されたピアからの SA メッセージのみを通過させます。拡張アクセス リスト *access-list-number* の範囲は 100 ～ 199 です。
		list と **route-map** の両方のキーワードを使用すると、すべての条件に一致しなければ、発信 SA メッセージ内のいずれの（S,G）ペアも通過できません。
		• ルート マップ *map-tag* 内の一致条件を満たす、指定された MSDP ピアからの SA メッセージのみを通過させます。
		すべての一致基準に当てはまる場合、ルート マップの **permit** がフィルタを通してルートを通過します。**deny** はルートをフィルタ処理します。

ステップ 4	**access-list** *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* 例： `Device(config)#access list 100 permit ip 194.1.22.0 1.1.1.1 194.3.44.0 1.1.1.1`	（任意）IP 拡張アクセス リストを作成します。必要な回数だけこのコマンドを繰り返します。 • *Access-list-number* には、ステップ 2 で指定した番号を入力します。 • **deny** キーワードは、条件が一致した場合にアクセスを拒否します。**permit** キーワードは、条件が一致した場合にアクセスを許可します。

	コマンドまたはアクション	目的
		• *protocol* には、プロトコル名として **ip** を入力します。 • *source*には、パケットの送信元であるネットワークまたはホストの番号を入力します。 • *source-wildcard* には、送信元に適用するワイルドカード ビットをドット付き 10 進表記で入力します。無視するビット位置には 1 を設定します。 • *destination* には、パケットの送信先であるネットワークまたはホストの番号を入力します。 • *destination-wildcard* には、宛先に適用するワイルドカード ビットをドット付き 10 進表記で入力します。無視するビット位置には 1 を設定します。 アクセス リストの末尾には、すべてに対する暗黙の拒否ステートメントが常に存在することに注意してください。
ステップ 5	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 6	**show running-config** 例： `Device#show running-config`	入力を確認します。
ステップ 7	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

MSDP メッシュ グループの設定

MSDP メッシュ グループは、MSDP によって完全なメッシュ型に相互接続された MSDP スピーカーのグループです。メッシュ グループ内のピアから受信された SA メッセージは、同じメッシュ グループ内の他のピアに転送されません。したがって、SA メッセージのフラッディング

が削減され、ピア RPF フラッディングが簡素化されます。ドメイン内に複数の RP がある場合は、**ip msdp mesh-group** グローバル コンフィギュレーション コマンドを使用します。特に、ドメインを越えて SA メッセージを送信する場合に使用します。単一のデバイスに複数のメッシュグループを（異なる名前で）設定できます。

メッシュ グループを作成するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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 msdp mesh-group** *name* {*ip-address* | *name*} 例： `Devic(config)#ip msdp mesh-group 2 switch.cisco.com`	MSDP メッシュ グループを設定し、そのメッシュ グループに属する MSDP ピアを指定します。 デフォルトでは、MSDP ピアはメッシュ グループに属しません。 • *name* には、メッシュ グループの名前を入力します。 • *ip-address* | *name* には、メッシュ グループのメンバーになる MSDP ピアの IP アドレスまたは名前を入力します。 グループ内の MSDP ピアごとに、この手順を繰り返します。
ステップ 4	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `Device#show running-config`	入力を確認します。

	コマンドまたはアクション	目的
ステップ 6	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

MSDP ピアのシャットダウン

複数の MSDP コマンドが設定された単一のピアをアクティブにしない場合は、ピアをシャットダウンしてから、あとで起動できます。ピアがシャットダウンすると、TCP 接続が終了し、再起動されません。ピアの設定情報を保持したまま、MSDP セッションをシャットダウンすることもできます。

ピアをシャットダウンするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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 msdp shutdown** {*peer-name* | *peer address*} 例： `Device(config)#ip msdp shutdown switch.cisco.com`	設定情報を保持したまま、指定された MSDP ピアをシャットダウン状態にします。 *peer-name* | *peer address* を指定する場合は、シャットダウンする MSDP ピアの IP アドレスまたは名前を入力します。
ステップ 4	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例：	入力を確認します。

	コマンドまたはアクション	目的
	`Device#show running-config`	
ステップ 6	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

境界 PIM デンス モード領域の MSDP への包含

デンスモード（DM）領域と PIM スパースモード（SM）領域の境界となるデバイスに MSDP を設定します。デフォルトでは、DM 領域のアクティブな送信元は MSDP に加入しません。

（注） **ip msdp border sa-address** グローバル コンフィギュレーション コマンドの使用は推奨できません。DM ドメイン内の送信元が SM ドメイン内の RP にプロキシ登録されるように SM ドメイン内の境界ルータを設定し、標準 MSDP 手順でこれらの送信元をアドバタイズするように SM ドメインを設定してください。

ip msdp originator-idグローバル コンフィギュレーション コマンドを実行すると、RP アドレスとして使用されるインターフェイスも識別されます。**ip msdp border sa-address** および **ip msdp originator-id** グローバル コンフィギュレーション コマンドの両方が設定されている場合、**ip msdp originator-id** コマンドから取得されたアドレスが RP アドレスを指定します。

DM 領域でアクティブな送信元の SA メッセージを MSDP ピアに送信するように境界ルータを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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。

	コマンドまたはアクション	目的
ステップ 3	**ip msdp border sa-address** *interface-id* 例： `Device(config)#ip msdp border sa-address 0/1`	DM 領域内のアクティブな送信元に関する SA メッセージを送信するように、DM 領域と SM 領域の境界スイッチを設定します。 *interface-id* には、SA メッセージ内の RP アドレスとして使用される、IP アドレスの配信元となるインターフェイスを指定します。 インターフェイスの IP アドレスは、SA メッセージ内の RP フィールド [Originator-ID] の値として使用されます。
ステップ 4	**ip msdp redistribute** [**list** *access-list-name*] [**asn** *aspath-access-list-number*] [**route-map** *map*] 例： `Device(config)#ip msdp redistribute list 100`	SA メッセージに格納されてアドバタイズされる、マルチキャスト ルーティング テーブル内の（S, G）エントリを設定します。詳細については、送信元の再配信 （44 ページ）を参照してください。
ステップ 5	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 6	**show running-config** 例： `Device#show running-config`	入力を確認します。
ステップ 7	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

RP アドレス以外の発信元アドレスの設定

SA メッセージの発信元である MSDP スピーカーで、インターフェイスの IP アドレスを SA メッセージ内の RP アドレスとして使用する場合は、送信元 ID を変更します。次のいずれかの場合に送信元 ID を変更できます。

- MSDP メッシュ グループ内の複数のデバイス上で、論理 RP を設定する場合。
- PIM SM ドメインと DM ドメインの境界となるデバイスがある場合。サイトの DM ドメインの境界となるデバイスがあり、SM がその外部で使用されている場合は、DM の送信元

を外部に通知する必要があります。このデバイスはRPでないため、SAメッセージで使用される RP アドレスはありません。したがって、このコマンドではインターフェイスのアドレスを指定し、RP アドレスを提供します。

ip msdp border sa-address および **ip msdp originator-id** グローバル コンフィギュレーション コマンドの両方が設定されている場合、**ip msdp originator-id** コマンドから取得されたアドレスが RP アドレスを指定します。

SA メッセージの発信元である MSDP スピーカーで、インターフェイスの IP アドレスを SA メッセージ内の RP アドレスとして使用できるようにするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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 msdp originator-id** *interface-id* 例： `Device(config)#ip msdp originator-id 0/1`	発信元デバイスのインターフェイスのアドレスとなるように、SA メッセージ内の RP アドレスを設定します。 *interface-id* には、ローカルデバイスのインターフェイスを指定します。
ステップ 4	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `Device#show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例：	（任意）コンフィギュレーション ファイルに設定を保存します。

	コマンドまたはアクション	目的
	`Device#copy running-config startup-config`	

MSDP のモニタリングおよびメンテナンス

MSDP SA メッセージ、ピア、状態、ピアのステータスをモニターするコマンドは以下のとおりです。

表 *3 : MSDP* のモニターおよびメンテナンスのためのコマンド

コマンド	目的
debug ip msdp [*peer-address* | *name*] [**detail**] [**routes**]	MSDP アクティビティをデバッグします。
debug ip msdp resets	MSDP ピアのリセット原因をデバッグします。
show ip msdp count [*autonomous-system-number*]	SA メッセージに格納され、各自律システムから発信された送信元およびグループの個数を表示します。**ip msdp cache-sa-state** コマンドは、このコマンドによって出力が生成されるように設定する必要があります。
show ip msdp peer [*peer-address* | *name*]	MSDP ピアに関する詳細情報を表示します。
show ip msdp sa-cache [*group-address* | *source-address* | *group-name* | *source-name*] [*autonomous-system-number*]	MSDP ピアから学習した（S,G）ステートを表示します。
show ip msdp summary	MSDP ピア ステータスおよび SA メッセージ数を表示します。

MSDP 接続、統計情報、SA キャッシュ エントリをクリアするコマンドは以下のとおりです。

表 *4 : MSDP* 接続、統計情報、または *SA* キャッシュ エントリをクリアするためのコマンド

コマンド	目的
clear ip msdp peer *peer-address* | *name*	指定された MSDP ピアへの TCP 接続をクリアし、すべての MSDP メッセージ カウンタをリセットします。
clear ip msdp statistics [*peer-address* | *name*]	セッションをリセットせずに、1 つまたはすべての MSDP ピア統計情報カウンタをクリアします。

コマンド	目的
clear ip msdp sa-cache [*group-address* | *name*]	すべてのエントリの SA キャッシュ エントリ、特定のグループのすべての送信元、または特定の送信元とグループのペアのすべてのエントリをクリアします。

MSDP の設定例

デフォルト MSDP ピアの設定：例

次に、ルータ A およびルータ C の部分的な設定の例を示します。これらの ISP にはそれぞれに複数のカスタマー（カスタマーと同様）がおり、デフォルトのピアリング（BGP または MBGP なし）を使用しています。この場合、両方の ISP で類似した設定となります。つまり、両方の ISP では、対応するプレフィックス リストで SA が許可されている場合、デフォルト ピアからの SA だけが受信されます。

ルータ A

```
Device(config)#ip msdp default-peer 10.1.1.1
Device(config)#ip msdp default-peer 10.1.1.1 prefix-list site-a
Device(config)#ip prefix-list site-b permit 10.0.0.0/1
```

ルータ C

```
Device(config)#ip msdp default-peer 10.1.1.1 prefix-list site-a
Device(config)#ip prefix-list site-b permit 10.0.0.0/1
```

SA ステートのキャッシング：例

次に、グループ 224.2.0.0/16 への送信元である 171.69.0.0/16 のすべての送信元のキャッシュ ステートをイネーブルにする例を示します。

```
Device(config)#ip msdp cache-sa-state 100
Device(config)#access-list 100 permit ip 171.69.0.0 0.0.255.255 224.2.0.0 0.0.255.255
```

MSDP ピアからの送信元情報の要求：例

次に、171.69.1.1 の MSDP ピアに SA 要求メッセージを送信するように、スイッチを設定する例を示します。

```
Device(config)#ip msdp sa-request 171.69.1.1
```

スイッチから発信される送信元情報の制御：例

次に、171.69.2.2 の MSDP ピアからの SA 要求メッセージをフィルタリングするように、スイッチを設定する例を示します。ネットワーク 192.4.22.0 の送信元からの SA 要求メッセージはアクセス リスト 1 に合格して、受信されます。その他のすべてのメッセージは無視されます。

```
Device(config)#ip msdp filter sa-request 171.69.2.2 list 1
Device(config)#access-list 1 permit 192.4.22.0 0.0.0.255
```

スイッチから転送される送信元情報の制御：例

次に、アクセス リスト 100 を通過する（S,G）ペアだけが SA メッセージに格納され、*switch.cisco.com* という名前のピアに転送されるように設定する例を示します。

```
Device(config)#ip msdp peer switch.cisco.com connect-source gigabitethernet1/0/1
Device(config)# ip msdp sa-filter out switch.cisco.com list 100
Device(config)#access-list 100 permit ip 171.69.0.0 0.0.255.255 224.20 0 0.0.255.255
```

スイッチで受信される送信元情報の制御：例

次に、*switch.cisco.com* という名前のピアからのすべての SA メッセージをフィルタリングする例を示します。

```
Device(config)#ip msdp peer switch.cisco.com connect-source gigabitethernet1/0/1
Device(config)#ip msdp sa-filter in switch.cisco.com
```

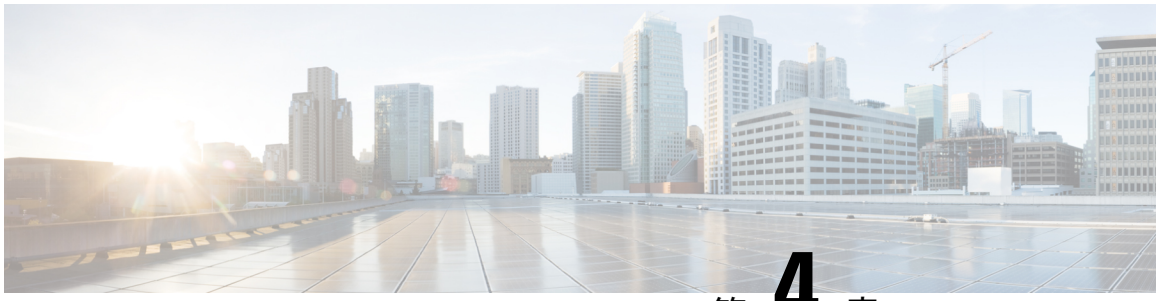
Multicast Source Discovery Protocol の機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS XE Everest 16.5.1a	MSDP	MSDP を使用すると、さまざまなドメイン内のすべてのランデブー ポイント（RP）に、グループのマルチキャスト送信元を通知できます。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn [英語] からアクセスします。



第 4 章

IP ユニキャスト ルーティングの設定

IP ユニキャストルーティングの制約事項

- IP ルーティングを有効にすると、SVI として設定されている VLAN は、他の宛先へのブロードキャスト ARP 要求も学習します。
- スイッチは、ユニキャストルーテッドトラフィックのトンネルインターフェイスをサポートしません。
- 設定できるルーテッドポートおよび SVI の個数は 2000 です。推奨個数と実装されている機能の数量を超えると、ハードウェアによって制限されるため、CPU利用率が影響を受けることがあります。
- このデバイスでは、サブネットワーク アクセス プロトコル（SNAP）アドレス解決はサポートされていません。

IP ユニキャスト ルーティングの設定に関する情報

このモジュールでは、スイッチで IP Version 4（IPv4）ユニキャスト ルーティングを設定する方法について説明します。

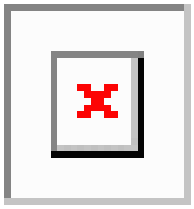
（注） IPv4 トラフィックに加えて、スイッチまたはスイッチスタックが Network Essentials または Network Advantage ライセンスを実行している場合、I 6（IPv6）ユニキャストルーティングをイネーブルにし、IPv6 トラフィックを転送するようにインターフェイスを設定できます。

IP ルーティングに関する情報

一部のネットワーク環境で、VLAN（仮想LAN）は各ネットワークまたはサブネットワークに関連付けられています。IP ネットワークで、各サブネットワークは 1 つの VLAN に対応しています。VLAN を設定すると、ブロードキャスト ドメインのサイズを制御し、ローカル トラフィックをローカル内にとどめることができます。ただし、異なる VLAN 内のネットワークデバイスが相互に通信するには、VLAN間でトラフィックをルーティング（VLAN間ルーティング）するレイヤ3デバイス（ルータ）が必要です。VLAN間ルーティングでは、適切な宛先 VLAN にトラフィックをルーティングするため、1 つまたは複数のルータを設定します。

図 *2:* ルーティング トポロジの例

次の図に基本的なルーティング トポロジを示します。スイッチ A は VLAN 10 内、スイッチ B は VLAN 20 内にあります。ルータには各 VLAN のインターフェイスが備わっています。



VLAN 10 内のホスト A が VLAN 10 内のホスト B と通信する場合、ホスト A はホスト B 宛にアドレス指定されたパケットを送信します。スイッチAはパケットをルータに送信せず、ホスト B に直接転送します。

ホスト A から VLAN 20 内のホスト C にパケットを送信する場合、スイッチ A はパケットをルータに転送し、ルータは VLAN 10 インターフェイスでトラフィックを受信します。ルータはルーティング テーブルを調べて正しい発信インターフェイスを判別し、VLAN 20 インターフェイスを経由してパケットをスイッチ B に送信します。スイッチ B はパケットを受信し、ホスト C に転送します。

ルーティング タイプ

ルータおよびレイヤ 3 スイッチは、次の方法でパケットをルーティングできます。

- デフォルト ルーティング
- 事前にプログラミングされているトラフィックのスタティック ルートの使用

デフォルトルーティングとは、宛先がルータにとって不明であるトラフィックをデフォルトの出口または宛先に送信することです。

スタティック ユニキャスト ルーティングの場合、パケットは事前に設定されたポートから単一のパスを通り、ネットワークの内部または外部に転送されます。スタティック ルーティングは安全で、帯域幅をほとんど使用しません。ただし、リンク障害などのネットワークの変更には自動的に対応しないため、パケットが宛先に到達しないことがあります。ネットワークが拡大するにつれ、スタティック ルーティングの設定は煩雑になります。

ルータでは、トラフィックを転送する最適ルートを動的に計算するため、ダイナミック ルーティング プロトコルが使用されます。ダイナミック ルーティング プロトコルには次の 2 つのタイプがあります。

- ディスタンスベクトル プロトコルを使用するルータでは、ネットワーク リソースの距離の値を使用してルーティング テーブルを保持し、これらのテーブルをネイバーに定期的に渡します。ディスタンスベクトルプロトコルは1つまたは複数のメトリックを使用し、最適なルートを計算します。これらのプロトコルは、簡単に設定、使用できます。
- リンクステート プロトコルを使用するルータでは、ルータ間のリンクステート アドバタイズメント（LSA）の交換に基づき、ネットワーク トポロジに関する複雑なデータベースを保持します。LSA はネットワークのイベントによって起動され、コンバージェンス時間、またはこれらの変更への対応時間を短縮します。リンクステートプロトコルはトポロジの変更にすばやく対応しますが、ディスタンスベクトルプロトコルよりも多くの帯域幅およびリソースが必要になります。

スイッチでサポートされているディスタンスベクトルプロトコルは、Routing Information Protocol（RIP）および Border Gateway Protocol（BGP）です。RIP は最適パスを決定するために単一の距離メトリック（コスト）を使用し、BGP はパス ベクトル メカニズムを追加します。また、Open Shortest Path First（OSPF）リンクステート プロトコル、および従来の Interior Gateway Routing Protocol（IGRP）にリンクステート ルーティング機能の一部を追加して効率化を図った Enhanced IGRP（EIGRP）もサポートされています。

（注） スイッチまたはスイッチ スタックでサポートされるプロトコルは、アクティブ スイッチ上で稼働しているソフトウェアによって決まります。アクティブ スイッチ上で Network Essentials ライセンスで稼働している場合は、デフォルトのルーティング、スタティック ルーティング、および RIP だけがサポートされます。他のすべてのルーティング プロトコルには、Network Advantage ライセンスが必要です。

IP ルーティングおよびスイッチ スタック

スタックのスイッチがルーティング ピアに接続されているかどうかに関係なく、スイッチ スタックはネットワークからは単一のスイッチとして認識されます。

アクティブ スイッチにより、次の機能が実行されます。

- distributed Cisco Express Forwarding（dCEF）データベースを生成および維持し、すべてのスタック メンバーに配信します。このデータベースに基づいて、スタック内のすべてのスイッチにルートがプログラミングされます。
- アクティブ スイッチの MAC アドレスはスタック全体のルータ MAC アドレスとして使用され、すべての外部デバイスはこのアドレスを使用して IP パケットをスタックに送信します。
- ソフトウェア転送またはソフトウェア処理を必要とするすべての IP パケットは、アクティブ スイッチの CPU を通ります。

スタック メンバーは、次に示す機能を実行します。

- ルーティング スタンバイ スイッチとして機能します。アクティブ スイッチに障害が発生し、新規アクティブ スイッチとして選択された場合に、処理を引き継ぐことができます。
- ルートをハードウェアにプログラムします。

アクティブ スイッチに障害が発生すると、スタックはアクティブ スイッチがダウンしていることを検出し、スタック メンバの 1 つを新規アクティブ スイッチとして選択します。この期間中に、ハードウェアは一時的な中断を除き、アクティブなプロトコルがない状態でパケットの転送を継続します。

ただし、スイッチ スタックが障害のあとハードウェア ID を維持していても、アクティブ スイッチの再起動前の短い中断の間にルータ ネイバーのルーティング プロトコルがフラップすることがあります。OSPF や EIGRP などのルーティング プロトコルは、ネイバーの移行を認識する必要があります。ルータは、次の 2 つのレベルの Nonstop Forwarding（NSF）を使用して、スイッチオーバーの検出、ネットワーク トラフィックの転送の継続、およびピア デバイスから情報の回復を行います。

- NSF 認識ルータによるネイバー ルータ障害の許容。ネイバー ルータの再起動後、NSF 認識ルータは要求を受けて自身のステート情報とルートの隣接情報を提供します。
- NSF 対応ルータによる NSF のサポート。NSF 対応ルータは、アクティブ スイッチの変更を検出した場合、NSF 認識ネイバーまたは NSF 対応ネイバーからの情報でルーティング情報を再構築します。再起動を待つことはしません。

スイッチ スタックは NSF 対応ルーティングを OSPF および EIGRP に対してサポートします。

新規アクティブ スイッチは、選択されたときに次の機能を実行します。

- ルーティング アップデートの生成、受信、および処理を開始します。
- ルーティング テーブルを構築し、CEF データベースを生成して、スタック メンバーに配信します。
- ルータ MAC アドレスとして自身の MAC アドレスを使用します。新規 MAC アドレスのネットワーク ピアに通知するために、新規ルータ MAC アドレスを使用して余分の ARP 応答を定期的に（5 分間の間、数秒おきに）送信します。

（注） 固定 MAC アドレス機能をスタックに設定していて、アクティブ スイッチに変更があった場合、設定された時間スタック MAC アドレスは変更されません。この期間に前のアクティブ スイッチがメンバスイッチとしてスタックに再加入する場合、スタック MAC アドレスは前のアクティブ スイッチの MAC アドレスのままになります。

- ARP 要求をプロキシ ARP IP アドレスに送信し、ARP 応答を受信して、各プロキシ ARP エントリの到達可能性を判別しようとします。到達可能なプロキシ ARP IP アドレスごとに、新規ルータ MAC アドレスを使用して gratuitous ARP 応答を生成します。このプロセスは、新規アクティブ スイッチが選択されたあと、5 分間繰り返されます。

（注） アクティブなスイッチで Network Advantage ライセンスを実行している場合、スタックは Enhanced IGRP（EIGRP）や Border Gateway Protocol（BGP）など、サポートされているすべてのプロトコルを実行できます。アクティブ スイッチに障害が発生し、新規に選択されたアクティブ スイッチ上で Network Essentials ライセンスが稼働している場合、これらのプロトコルはスタック内で稼働しなくなります。

注意 スイッチ スタックを複数のスタックに分割すると、ネットワークが適切に動作しなくなる場合があります。

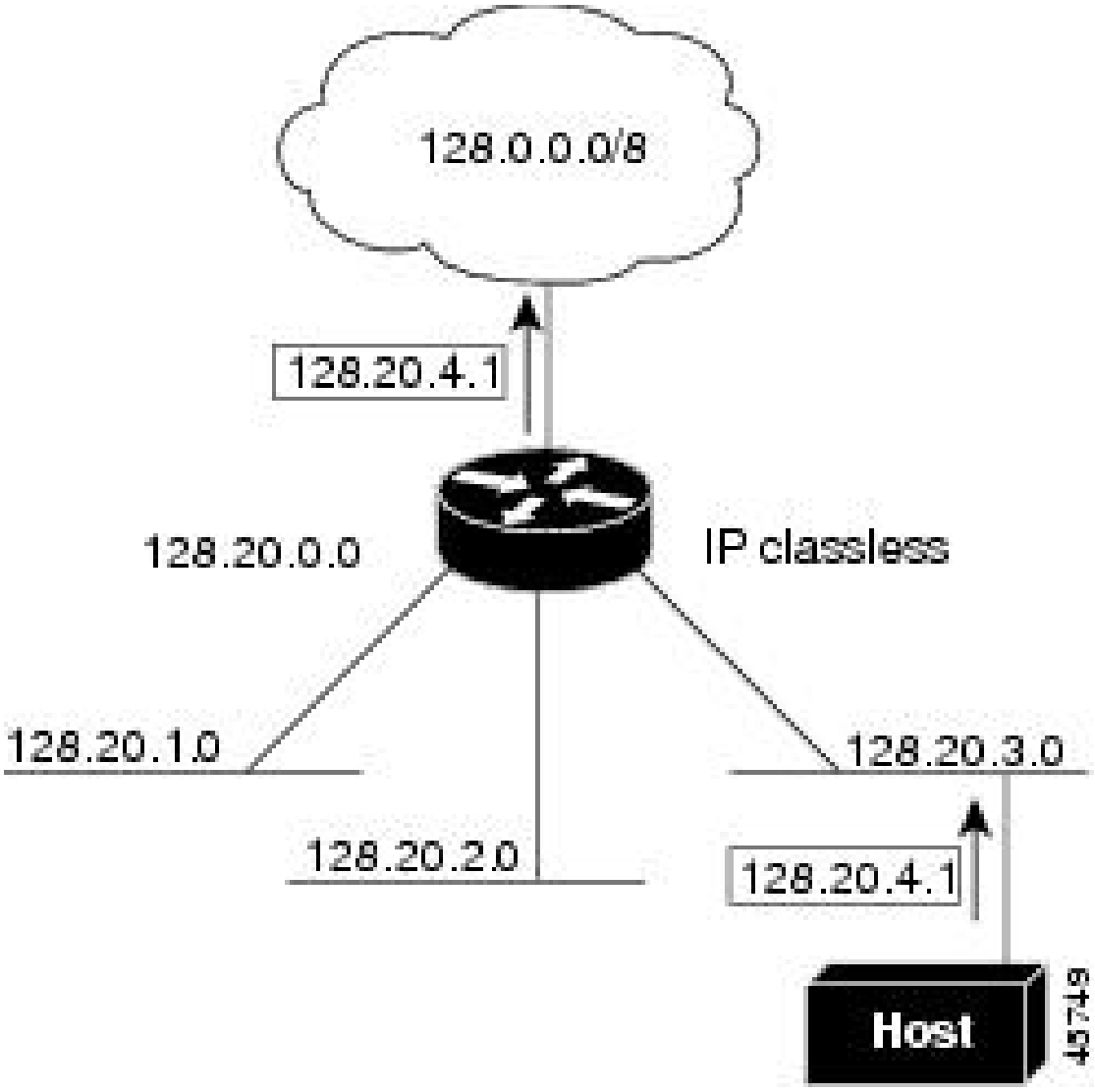
スイッチがリロードされると、NSF/SSO機能である場合でも、そのスイッチのポートがすべてダウンし、ルーティングに関わるインターフェイスにトラフィックの損失が発生します。

クラスレス ルーティング

ルーティングを行うように設定されたデバイスで、クラスレスルーティング動作はデフォルトで有効となっています。クラスレス ルーティングがイネーブルの場合、デフォルト ルートがないネットワークのサブネット宛てパケットをルータが受信すると、ルータは最適なスーパーネット ルートにパケットを転送します。スーパーネットは、単一の大規模アドレス空間をシミュレートするために使用されるクラスCアドレス空間の連続ブロックで構成されています。スーパーネットは、クラス B アドレス空間の急速な枯渇を回避するために設計されました。

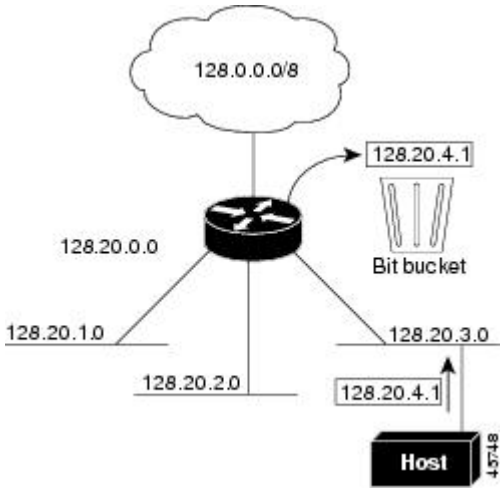
図では、クラスレスルーティングがイネーブルとなっています。ホストがパケットを128.20.4.1に送信すると、ルータはパケットを廃棄せずに、最適なスーパーネットルートに転送します。クラスレス ルーティングがディセーブルの場合、デフォルト ルートがないネットワークのサブネット宛てパケットを受信したルータは、パケットを廃棄します。

図 *3: IP* クラスレス ルーティングがイネーブルの場合



図では、ネットワーク 128.20.0.0 のルータはサブネット 128.20.1.0、128.20.2.0、128.20.3.0 に接続されています。ホストがパケットを 128.20.4.1 に送信した場合、ネットワークのデフォルトルートが存在しないため、ルータはパケットを廃棄します。

図 *4 : IP* クラスレス ルーティングがディセーブルの場合



デバイスが認識されないサブネット宛てのパケットを最適なスーパーネットルートに転送しないようにするには、クラスレスルーティング動作を無効にします。

アドレス解決

インターフェイス固有の IP 処理方法を制御するには、アドレス解決を行います。IP を使用するデバイスには、ローカルセグメントまたはLAN上のデバイスを一意に定義するローカル アドレス（MAC アドレス）と、デバイスが属するネットワークを特定するネットワーク アドレスがあります。

（注） スイッチスタックでは、スタックの単一のMACアドレスおよびIPアドレスを使用して、ネットワーク通信を行います。

ローカルアドレス（MAC アドレス）は、パケット ヘッダーのデータ リンク層（レイヤ 2）セクションに格納されて、データ リンク（レイヤ2）デバイスによって読み取られるため、データ リンク アドレスと呼ばれます。ソフトウェアがイーサネット上のデバイスと通信するには、デバイスの MAC アドレスを学習する必要があります。IP アドレスから MAC アドレスを学習するプロセスを、アドレス解決と呼びます。MAC アドレスから IP アドレスを学習するプロセスを、逆アドレス解決と呼びます。

デバイスでは、次の形式のアドレス解決を行うことができます。

- ARP：IP アドレスを MAC アドレスと関連付けるために使用されます。ARP は IP アドレスを入力と解釈し、対応する MAC アドレスを学習します。次に、IP アドレス/MAC アドレス アソシエーションを ARP キャッシュにストアし、すぐに取り出せるようにします。その後、IPデータグラムがリンク層フレームにカプセル化され、ネットワークを通じて送信されます。

- プロキシ ARP：ルーティング テーブルを持たないホストで、他のネットワークまたはサブネット上のホストの MAC アドレスを学習できるようにします。デバイス（ルータ）が送信者と異なるインターフェイス上のホストに宛てた ARP 要求を受信した場合、そのルータに他のインターフェイスを経由してそのホストに至るすべてのルートが格納されていれば、ルータは自身のローカル データ リンク アドレスを示すプロキシ ARP パケットを生成します。ARP 要求を送信したホストはルータにパケットを送信し、ルータはパケットを目的のホストに転送します。

デバイスでは、ARP と同様の機能（ローカル MAC アドレスでなく IP アドレスを要求する点を除く）を持つ Reverse Address Resolution Protocol（RARP）を使用することもできます。RARP を使用するには、ルータ インターフェイスと同じネットワーク セグメント上に RARP サーバーを設置する必要があります。サーバーを識別するには、**ip rarp-server** *address* インターフェイス コンフィギュレーション コマンドを使用します。

プロシキ ARP

プロキシ ARP は、他のルートを学習する場合の最も一般的な方法です。プロキシ ARP を使用すると、ルーティング情報を持たないイーサネットホストと、他のネットワークまたはサブネット上のホストとの通信が可能になります。このホストでは、すべてのホストが同じローカルイーサネット上にあり、ARP を使用して MAC アドレスを学習すると想定されています。デバイスが送信元と異なるネットワーク上にあるホストに宛てた ARP 要求を受信した場合、デバイスはそのホストへの最適なルートがあるかどうかを調べます。最適なルートがある場合、デバイスは自身のイーサネット MAC アドレスが格納された ARP 応答パケットを送信します。要求の送信元ホストはパケットをデバイスに送信し、スイッチは目的のホストにパケットを転送します。プロキシ ARP は、すべてのネットワークをローカルな場合と同様に処理し、IP アドレスごとに ARP 要求を実行します。

ICMP Router Discovery Protocol

ルータディスカバリを使用すると、デバイスは ICMP Router Discovery Protocol（IRDP）を使用し、他のネットワークへのルートを動的に学習します。ホストは IRDP を使用し、ルータを特定します。クライアントとして動作しているデバイスは、ルータディスカバリパケットを生成します。ホストとして動作しているデバイスは、ルータディスカバリパケットを受信します。デバイスは Routing Information Protocol（RIP）ルーティングのアップデートを受信し、この情報を使用してルータの場所を推測することもできます。ルーティングデバイスによって送信されたルーティングテーブルは、実際にはデバイスにストアされません。どのシステムがデータを送信しているのかが記録されるだけです。IRDP を使用する利点は、プライオリティと、パケットが受信されなくなってからデバイスがダウンしていると見なされるまでの期間の両方をルータごとに指定できることです。

検出された各デバイスは、デフォルト ルータの候補となります。現在のデフォルト ルータがダウンしたと宣言された場合、または再送信が多すぎて TCP 接続がタイムアウトになりつつある場合、プライオリティが上位のルータが検出されると、最も高いプライオリティを持つ新しいルータが選択されます。

IP ルーティングの有効化または無効化中は、IRDP パケットは送信されません。インターフェイスのシャットダウン中は、最後の IRDP メッセージに有効期間がありません。すべてのルータで 0 になります。

UDP ブロードキャスト パケットおよびプロトコル

ユーザー データグラム プロトコル（UDP）は IP のホスト間レイヤ プロトコルで、TCP と同様です。UDP はオーバーヘッドが少ない、コネクションレスのセッションを 2 つのエンド システム間に提供しますが、受信されたデータグラムの確認応答は行いません。場合に応じてネットワーク ホストは UDP ブロードキャストを使用し、アドレス、コンフィギュレーション、名前に関する情報を検索します。このようなホストが、サーバーを含まないネットワーク セグメント上にある場合、通常 UDP ブロードキャストは転送されません。この状況を改善するには、特定のクラスのブロードキャストをヘルパー アドレスに転送するように、ルータのインターフェイスを設定します。インターフェイスごとに、複数のヘルパー アドレスを使用できます。

UDP 宛先ポートを指定し、転送される UDP サービスを制御できます。複数の UDP プロトコルを指定することもできます。旧式のディスクレス Sun ワークステーションおよびネットワーク セキュリティ プロトコル SDNS で使用される Network Disk（ND）プロトコルも指定できます。

ヘルパー アドレスがインターフェイスに定義されている場合、デフォルトでは UDP と ND の両方の転送がイネーブルになっています。

ブロードキャスト パケットの処理

IP インターフェイスアドレスを設定したあとで、ルーティングをイネーブルにしたり、1 つまたは複数のルーティングプロトコルを設定したり、ネットワーク ブロードキャストへのデバイスの応答方法を設定したりできます。ブロードキャストは、物理ネットワーク上のすべてのホスト宛てのデータ パケットです。デバイスでは、2 種類のブロードキャストがサポートされています。

- ダイレクト ブロードキャスト パケット：特定のネットワークまたは一連のネットワークに送信されます。ダイレクト ブロードキャスト アドレスには、ネットワークまたはサブネット フィールドが含まれます。
- フラッディング ブロードキャスト パケット：すべてのネットワークに送信されます。

（注） **storm-control**インターフェイス コンフィギュレーション コマンドを使用して、トラフィック抑制レベルを設定し、レイヤ 2 インターフェイスでブロードキャスト、ユニキャスト、マルチキャスト トラフィックを制限することもできます。

ルータはローカル ケーブルまでの範囲を制限して、ブロードキャスト ストームを防ぎます。ブリッジ（インテリジェントなブリッジを含む）はレイヤ 2 デバイスであるため、ブロードキャストはすべてのネットワーク セグメントに転送され、ブロードキャスト ストームを伝播します。ブロードキャスト ストーム問題を解決する最善の方法は、ネットワーク上で単一のブ

ロードキャスト アドレス方式を使用することです。最新の IP 実装機能ではほとんどの場合、アドレスをブロードキャストアドレスとして使用するように設定できます。デバイスの場合も含めて、多くの実装機能では、ブロードキャストメッセージを転送するためのアドレス方式が複数サポートされています。

IP ブロードキャストのフラッディング

IPブロードキャストをインターネットワーク全体に、制御可能な方法でフラッディングできるようにするには、ブリッジング STP で作成されたデータベースを使用します。この機能を使用すると、ループを回避することもできます。この機能を使用できるようにするには、フラッディングが行われるインターフェイスごとにブリッジングを設定する必要があります。ブリッジングが設定されていないインターフェイス上でも、ブロードキャストを受信できます。ただし、ブリッジングが設定されていないインターフェイスでは、受信したブロードキャストが転送されません。また、異なるインターフェイスで受信されたブロードキャストを送信する場合、このインターフェイスは使用されません。

IPヘルパーアドレスのメカニズムを使用して単一のネットワーク アドレスに転送されるパケットを、フラッディングできます。各ネットワーク セグメントには、パケットのコピーが 1 つだけ送信されます。

フラッディングを行う場合、パケットは次の条件を満たす必要があります（これらの条件は、IP ヘルパー アドレスを使用してパケットを転送するときの条件と同じです）。

- パケットは MAC レベルのブロードキャストでなければなりません。
- パケットは IP レベルのブロードキャストでなければなりません。
- パケットは Trivial File Transfer Protocol（TFTP）、ドメインネームシステム（DNS）、Time、NetBIOS、ND、または BOOTP パケット、または **ip forward-protocol udp** グローバル コンフィギュレーション コマンドで指定された UDP でなければなりません。
- パケットの存続可能時間（TTL）値は 2 以上でなければなりません。

フラッディングされた UDP データグラムには、出力インターフェイスで **ip broadcast-address** インターフェイス コンフィギュレーション コマンドによって指定された宛先アドレスが表示されます。宛先アドレスを、任意のアドレスに設定できます。このため、データグラムがネットワーク内に伝播されるにつれ、宛先アドレスが変更されることもあります。送信元アドレスは変更されません。TTL 値が減ります。

フラッディングされた UDP データグラムがインターフェイスから送信されると（場合によっては宛先アドレスが変更される）、データグラムは通常の IP 出力ルーチンに渡されます。このため、出力インターフェイスにアクセス リストがある場合、データグラムはその影響を受けます。

スイッチでは、パケットの大部分がハードウェアで転送され、スイッチの CPU を経由しません。CPUに送信されるパケットの場合は、ターボフラッディングを使用し、スパニングツリーベースの UDP フラッディングを約 4 ～ 5 倍高速化します。この機能は、ARP カプセル化用に設定されたイーサネット インターフェイスでサポートされています。

IP ルーティング設定時の注意事項

デバイス上で、IPルーティングはデフォルトでディセーブルとなっているため、ルーティングを行う前に、IP ルーティングをイネーブルにする必要があります。

次の手順では、次に示すレイヤ 3 インターフェイスの 1 つを指定する必要があります。

- ルーテッドポート：**no switchport** インターフェイス コンフィギュレーション コマンドを使用し、レイヤ 3 ポートとして設定された物理ポートです。
- スイッチ仮想インターフェイス（SVI）：**interface vlan** *vlan_id* グローバル コンフィギュレーション コマンドによって作成された VLAN インターフェイス。デフォルトではレイヤ 3 インターフェイスです。
- レイヤ 3 モードの Etherchannel ポートチャネル： **interface port-channel** *port-channel-number* グローバル コンフィギュレーション コマンドを使用し、イーサネット インターフェイスをチャネルグループにバインドして作成されたポートチャネル論理インターフェイスです。

ルーティングが発生するすべてのレイヤ3インターフェイスに、IPアドレスを割り当てる必要があります。

（注） スイッチは、各ルーテッド ポートおよび SVI に割り当てられた IP アドレスを持つことができます。

ルーティングを設定するための主な手順は次のとおりです。

- VLAN インターフェイスをサポートするために、スイッチまたはスイッチ スタックで VLAN を作成および設定し、レイヤ 2 インターフェイスに VLAN メンバーシップを割り当てます。詳細については、「VLAN の設定」の章を参照してください。
- レイヤ 3 インターフェイスを設定します。
- スイッチ上で IP ルーティングをイネーブルに設定します。
- レイヤ 3 インターフェイスに IP アドレスを割り当てます。
- 選択したルーティング プロトコルをスイッチ上でイネーブルにします。
- ルーティング プロトコル パラメータを設定します（任意）。

IP アドレッシングの設定方法

IP ルーティングを設定するには、レイヤ 3 ネットワーク インターフェイスに IP アドレスを割り当ててインターフェイスをイネーブルにし、IPを使用するインターフェイスを経由してホストとの通信を許可する必要があります。次の項では、さまざまな IP アドレス指定機能の設定

方法について説明します。IPアドレスをインターフェイスに割り当てる手順は必須ですが、その他の手順は任意です。

- アドレス指定のデフォルト設定
- ネットワーク インターフェイスへの IP アドレスの割り当て
- アドレス解決方法の設定
- IP ルーティングがディセーブルの場合のルーティング支援機能
- ブロードキャスト パケットの処理方法の設定
- IP アドレスのモニターリングおよびメンテナンス

IP アドレス指定のデフォルト設定

表 *5:* アドレス指定のデフォルト設定

機能	デフォルト設定
IP アドレス	未定義
ARP	ARP キャッシュに永続的なエントリはありません カプセル化：標準イーサネット形式の ARP 14400 秒（4 時間）
IP ブロードキャスト アドレス	255.255.255.255（すべて 1）
IP クラスレス ルーティング	イネーブル。
IP デフォルト ゲートウェイ	ディセーブル。
IP ダイレクト ブロードキャスト	ディセーブル（すべての IP ダイレクト ブロードキャストがドロップされます）
IP ドメイン	ドメイン リスト：ドメイン名は未定義 ドメイン検索：イネーブル ドメイン名：イネーブル

機能	デフォルト設定
IP 転送プロトコル	ヘルパー アドレスが定義されているか、またはユーザー データグラム プロフラッディングが設定されている場合、デフォルト ポートでは UDP 転送がります ローカル ブロードキャスト：ディセーブル スパニングツリー プロトコル（STP）：ディセーブル ターボフラッディング：ディセーブル
IP ヘルパー アドレス	ディセーブル。
IP ホスト	ディセーブル。
ICMP Router Discovery Protocol（IRDP）	ディセーブル。 イネーブルの場合のデフォルト： • ブロードキャスト IRDP アドバタイズメント • アドバタイズメント間の最大インターバル：600 秒 • アドバタイズ間の最小インターバル：最大インターバルの 0.75 倍 • プリファレンス：0
IP プロキシ ARP	イネーブル。
IP ルーティング	ディセーブル。
IP サブネットゼロ	ディセーブル

ネットワーク インターフェイスへの IP アドレスの割り当て

IP アドレスは IP パケットの送信先を特定します。一部の IP アドレスは特殊な目的のために予約されていて、ホスト、サブネット、またはネットワーク アドレスには使用できません。RFC 1166 の『Internet Numbers』には IP アドレスに関する公式の説明が記載されています。

インターフェイスには、1 つのプライマリ IP アドレスを設定できます。マスクで、IP アドレス中のネットワーク番号を示すビットが識別できます。マスクを使用してネットワークをサブネット化する場合、そのマスクをサブネット マスクと呼びます。割り当てられているネットワーク番号については、インターネット サービス プロバイダにお問い合わせください。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例：	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
	`Device>enable`	
ステップ 2	**configure terminal** 例： `Device#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	**interface** *interface-id* 例： `Device(config)#interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ3インターフェイスを指定します。
ステップ 4	**no switchport** 例： `Device(config-if)#no switchport`	レイヤ 2 コンフィギュレーション モードからインターフェイスを削除します（物理インターフェイスの場合）。
ステップ 5	**ip address** *ip-address subnet-mask* 例： `Device(config-if)#ip address 10.1.5.1 255.255.255.0`	IP アドレスおよび IP サブネット マスクを設定します。
ステップ 6	**no shutdown** 例： `Device(config-if)#no shutdown`	物理インターフェイスをイネーブルにします。
ステップ 7	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 8	**show ip route** 例： `Device#show ip route`	入力を確認します。
ステップ 9	**show ip interface** [*interface-id*] 例： `Device#show ip interface gigabitethernet 1/0/1`	入力を確認します。

	コマンドまたはアクション	目的
ステップ 10	**show running-config** 例： `Device#show running-config`	入力を確認します。
ステップ 11	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

サブネット ゼロの使用

サブネット アドレスがゼロであるサブネットを作成しないでください。同じアドレスを持つネットワークおよびサブネットがある場合に問題が発生することがあります。たとえば、ネットワーク 131.108.0.0 のサブネットが 255.255.255.0 の場合、サブネット ゼロは 131.108.0.0 と記述され、ネットワーク アドレスと同じとなってしまいます。

すべてが 1 のサブネット（131.108.255.0）は使用可能です。また、IP アドレス用にサブネット スペース全体が必要な場合は、サブネット ゼロの使用をイネーブルにできます（ただし推奨できません）。

デフォルトに戻して、サブネットゼロの使用を無効にするには、**no ip subnet-zero** グローバル コンフィギュレーション コマンドを使用します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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 subnet-zero** 例： `Device(config)#ip subnet-zero`	インターフェイス アドレスおよびルーティングのアップデート時にサブネット ゼロの使用をイネーブルにします。

	コマンドまたはアクション	目的
ステップ 4	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `Device#show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

クラスレス ルーティングのディセーブル化

デバイスが認識されないサブネット宛てのパケットを最適なスーパーネットルートに転送しないようにするには、クラスレスルーティング動作を無効にします。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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	**no ip classless** 例： `Device(config)#no ip classless`	クラスレスルーティング動作をディセーブルにします。
ステップ 4	**end** 例：	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
	`Device(config)#end`	
ステップ 5	**show running-config** 例： `Device#show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

アドレス解決方法の設定

アドレス解決を設定するために必要な作業は次のとおりです。

スタティック ARP キャッシュの定義

ARP および他のアドレス解決プロトコルを使用すると、IP アドレスと MAC アドレス間をダイナミックにマッピングできます。ほとんどのホストではダイナミック アドレス解決がサポートされているため、通常の場合、スタティック ARP キャッシュ エントリを指定する必要はありません。静的 ARP キャッシュエントリを定義する必要がある場合は、グローバルに行うことができます。グローバルに定義すると、IP アドレスを MAC アドレスに変換するためにデバイスが使用する ARP キャッシュに永続的なエントリをインストールします。また、指定された IP アドレスに属しているかのように、デバイスが ARP 要求に応答するように指定することもできます。ARP エントリを永続的なエントリにしない場合は、ARP エントリのタイムアウト期間を指定できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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。

	コマンドまたはアクション	目的
ステップ 3	**arp** *ip-address hardware-address type* 例： `Device(config)#ip 10.1.5.1 c2f3.220a.12f4 arpa`	ARP キャッシュ内で IP アドレスを MAC（ハードウェア）アドレスに関連付け、次に示すカプセル化タイプのいずれかを指定します。 • **arpa**：ARP カプセル化（イーサネット インターフェイス用） • **sap**：HP の ARP タイプ
ステップ 4	**arp** *ip-address hardware-address type* [**alias**] 例： `Device(config)#ip 10.1.5.3 d7f3.220d.12f5 arpa alias`	（任意）指定された IP アドレスがスイッチに属する場合と同じ方法で、スイッチが ARP 要求に応答するように指定します。
ステップ 5	**interface** *interface-id* 例： `Device(config)#interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するインターフェイスを指定します。
ステップ 6	**arp** *timeout seconds* 例： `Device(config-if)#arp 20000`	（任意）ARP キャッシュ エントリがキャッシュに保持される期間を設定します。デフォルト値は 14400 秒（4 時間）です。指定できる範囲は 0 ～ 2147483 秒です。
ステップ 7	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 8	**show interfaces** [*interface-id*] 例： `Device#show interfaces gigabitethernet 1/0/1`	すべてのインターフェイスまたは特定のインターフェイスで使用される ARP のタイプおよびタイムアウト値を確認します。
ステップ 9	**show arp** 例： `Device#show arp`	ARP キャッシュの内容を表示します。
ステップ 10	**show ip arp** 例： `Device#show ip arp`	ARP キャッシュの内容を表示します。

	コマンドまたはアクション	目的
ステップ 11	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ARP のカプセル化の設定

IP インターフェイスでは、イーサネット ARP カプセル化（**arpa** キーワードで表される）がデフォルトで有効に設定されています。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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	**interface** *interface-id* 例： `Device(config)#interface gigabitethernet 1/0/2`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 4	**arp arpa** 例： `Device(config-if)#arp arpa`	ARP カプセル化方式を指定します。 **no arp arpa** コマンドを使用して、ARP カプセル化方式を無効にします。
ステップ 5	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 6	**show interfaces** [*interface-id*] 例：	すべてのインターフェイスまたは指定されたインターフェイスの ARP カプセル化設定を確認します。

	コマンドまたはアクション	目的
	`Device#show interfaces`	
ステップ 7	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

プロキシ ARP のイネーブル化

デフォルトでは、プロキシ ARP がデバイスで使用されます。ホストが他のネットワークまたはサブネット上のホストの MAC アドレスを学習できるようにするためです。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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	**interface** *interface-id* 例： `Device(config)#interface gigabitethernet 1/0/2`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 4	**ip proxy-arp** 例： `Device(config-if)#ip proxy-arp`	インターフェイス上でプロキシ ARP をイネーブルにします。
ステップ 5	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
ステップ 6	**show ip interface** [*interface-id*] 例： `Device#show ip interface gigabitethernet 1/0/2`	指定されたインターフェイスまたはすべてのインターフェイスの設定を確認します。
ステップ 7	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

IP ルーティングがディセーブルの場合のルーティング支援機能

次のメカニズムを使用することで、デバイスは、IP ルーティングが有効でない場合、別のネットワークへのルートを学習できます。

- 『Proxy ARP』
- デフォルト ゲートウェイ
- ICMP Router Discovery Protocol（IRDP）

プロシキ ARP

プロキシ ARP は、デフォルトでイネーブルに設定されています。ディセーブル化されたプロキシ ARP をイネーブルにするには、「プロキシ ARP のイネーブル化」の項を参照してください。プロキシ ARP は、他のルータでサポートされているかぎり有効です。

デフォルト ゲートウェイ

ルートを特定するもう 1 つの方法は、デフォルト ルータ、つまりデフォルト ゲートウェイを定義する方法です。ローカルでないすべてのパケットはこのルータに送信されます。このルータは適切なルーティングを行う、または IP 制御メッセージ プロトコル（ICMP）リダイレクト メッセージを返信するという方法で、ホストが使用するローカル ルータを定義します。デバイスはリダイレクト メッセージをキャッシュに格納し、各パケットをできるだけ効率的に転送します。この方法には、デフォルト ルータがダウンした場合、または使用できなくなった場合に、検出が不可能となる制限があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例：	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
	Device>**enable**	
ステップ 2	**configure terminal** 例： Device#**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 default-gateway** *ip-address* 例： Device(config)#ip default gateway 10.1.5.1	デフォルト ゲートウェイ（ルータ）を設定します。
ステップ 4	**end** 例： Device(config)#**end**	特権 EXEC モードに戻ります。
ステップ 5	**show ip redirects** 例： Device#show ip redirects	設定を確認するため、デフォルト ゲートウェイ ルータのアドレスを表示します。
ステップ 6	**copy running-config startup-config** 例： Device#**copy running-config startup-config**	（任意）コンフィギュレーション ファイルに設定を保存します。

ICMP Router Discovery Protocol（IRDP）

インターフェイスでIRDPルーティングを行う場合は、インターフェイスでIRDP処理をイネーブルにしてください。IRDP 処理をイネーブルにすると、デフォルトのパラメータが適用されます。

これらのパラメータを変更することもできます。**maxadvertinterval** 値を変更すると、**holdtime** 値および **minadvertinterval** 値も変更されます。最初に **maxadvertinterval** 値を変更し、次に **holdtime** 値または **minadvertinterval** 値のどちらかを手動で変更することが重要です。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例：	特権 EXEC モードを有効にします。

	コマンドまたはアクション	目的
	`Device>enable`	• パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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	**interface** *interface-id* 例： `Device(config)#interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ3インターフェイスを指定します。
ステップ 4	**ip irdp** 例： `Device(config-if)#ip irdp`	インターフェイスでIRDP処理をイネーブルにします。
ステップ 5	**ip irdp multicast** 例： `Device(config-if)#ip irdp multicast`	（任意）IP ブロードキャストの代わりとして、マルチキャスト アドレス（224.0.0.1）にIRDP アドバタイズを送信します。 （注） このコマンドを使用すると、IRDP パケットをマルチキャストとして送信するサン マイクロシステムズ社の Solaris との互換性を維持できます。実装機能の中には、これらのマルチキャストを受信できないものも多くあります。このコマンドを使用する前に、エンドホストがこの機能に対応していることを確認してください。
ステップ 6	**ip irdp holdtime** *seconds* 例： `Device(config-if)#ip irdp holdtime 1000`	（任意）アドバタイズが有効であるIRDP期間を設定します。デフォルトは **maxadvertinterval** 値の 3 倍です。**maxadvertinterval** 値よりも大きな値（9000 秒以下）を指定する必要があります。**maxadvertinterval**値を変更すると、この値も変更されます。
ステップ 7	**ip irdp maxadvertinterval** *seconds* 例： `Device(config-if)#ip irdp maxadvertinterval 650`	（任意）アドバタイズメントのIRDP最大間隔を設定します。デフォルトは 600 秒です。

	コマンドまたはアクション	目的
ステップ 8	**ip irdp minadvertinterval** *seconds* 例： `Device(config-if)#ip irdp minadvertinterval 500`	（任意）アドバタイズ間のIRDPの最小インターバルを設定します。デフォルト値は**maxadvertinterval**値の0.75倍です。**maxadvertinterval**を変更すると、この値も新しいデフォルト値（**maxadvertinterval**の0.75倍）に変更されます。
ステップ 9	**ip irdp preference** *number* 例： `Device(config-if)#ip irdp preference 2`	（任意）デバイスの IRDP プリファレンス レベルを設定します。指定できる範囲は-231～231です。デフォルトは0です。大きな値を設定すると、ルータのプリファレンス レベルも高くなります。
ステップ 10	**ip irdp address** *address* [*number*] 例： `Device(config-if)#ip irdp address 10.1.10.10`	（任意）プロキシアドバタイズを行うためのIRDPアドレスとプリファレンスを設定します。
ステップ 11	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 12	**show ip irdp** 例： `Device#show ip irdp`	IRDP 値を表示し、設定を確認します。
ステップ 13	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ブロードキャスト パケットの処理方法の設定

これらの方式をイネーブルにするには、次に示す作業を実行します。

- ダイレクト ブロードキャストから物理ブロードキャストへの変換のイネーブル化
- UDP ブロードキャスト パケットおよびプロトコルの転送
- IP ブロードキャスト アドレスの確立
- IP ブロードキャストのフラッディング

ダイレクト ブロードキャストから物理ブロードキャストへの変換のイネーブル化

デフォルトでは、IP ダイレクト ブロードキャストがドロップされるため、転送されることはありません。IP ダイレクト ブロードキャストがドロップされると、ルータが DoS 攻撃（サービス拒絶攻撃）にさらされる危険が少なくなります。

ブロードキャストが物理（MAC レイヤ）ブロードキャストになるインターフェイスでは、IP ダイレクト ブロードキャストの転送をイネーブルにできます。**ip forward-protocol** グローバル コンフィギュレーション コマンドを使用し、設定されたプロトコルだけを転送できます。

転送するブロードキャストを制御するアクセス リストを指定できます。アクセス リストを指定すると、アクセス リストで許可されている IP パケットだけが、ダイレクト ブロードキャストから物理ブロードキャストに変換できるようになります。アクセスリストの詳細については、『*Security Configuration Guide*』の「Configuring ACLs」の章を参照してください。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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	**interface** *interface-id* 例： `Device(config)#interface gigabitethernet 1/0/2`	インターフェイス コンフィギュレーション モードを開始し、設定するインターフェイスを指定します。
ステップ 4	**ip directed-broadcast** [*access-list-number*] 例： `Device(config-if)#ip directed-broadcast 103`	インターフェイス上で、ダイレクト ブロードキャストから物理ブロードキャストへの変換をイネーブルにします。転送するブロードキャストを制御するアクセス リストを指定できます。アクセス リストを指定すると、アクセス リストで許可されている IP パケットだけが変換可能になります。
ステップ 5	**exit** 例： `Device(config-if)#exit`	グローバル コンフィギュレーション モードに戻ります。

	コマンドまたはアクション	目的
ステップ 6	**ip forward-protocol** {**udp** [*port*] | **nd** | **sdns**} 例： `Device(config)#ip forward-protocol nd`	ブロードキャスト パケットを転送するとき、ルータによって転送されるプロトコルおよびポートを指定します。 • **udp**：UPD データグラムを転送します。 port：（任意）転送される UDP サービスを制御する宛先ポートです。 • **nd**：ND データグラムを転送します。 • **sdns**：SDNS データグラムを転送します。
ステップ 7	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 8	**show ip interface** [*interface-id*] 例： `Device#show ip interface`	指定されたインターフェイスまたはすべてのインターフェイスの設定を確認します。
ステップ 9	**show running-config** 例： `Device#show running-config`	入力を確認します。
ステップ 10	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

UDP ブロードキャスト パケットおよびプロトコルの転送

UDPブロードキャストの転送を設定するときにUDPポートを指定しないと、ルータはBOOTP フォワーディング エージェントとして動作するように設定されます。BOOTP パケットは Dynamic Host Configuration Protocol（DHCP）情報を伝達します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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	**interface** *interface-id* 例： `Device(config)#interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ3インターフェイスを指定します。
ステップ 4	**ip helper-address** *address* 例： `Device(config-if)#ip helper address 10.1.10.1`	転送をイネーブルにし、BOOTP などの UDP ブロードキャスト パケットを転送するための宛先アドレスを指定します。
ステップ 5	**exit** 例： `Device(config-if)#exit`	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 6	**ip forward-protocol** {**udp** [*port*] | **nd** | **sdns**} 例： `Device(config)#ip forward-protocol sdns`	ブロードキャストパケットを転送するときに、ルータによって転送されるプロトコルを指定します。
ステップ 7	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 8	**show ip interface** [*interface-id*] 例： `Device#show ip interface gigabitethernet 1/0/1`	指定されたインターフェイスまたはすべてのインターフェイスの設定を確認します。
ステップ 9	**show running-config** 例：	入力を確認します。

	コマンドまたはアクション	目的
	Device#**show running-config**	
ステップ 10	**copy running-config startup-config** 例： Device#**copy running-config startup-config**	（任意）コンフィギュレーション ファイルに設定を保存します。

IP ブロードキャスト アドレスの確立

最も一般的な（デフォルトの）IP ブロードキャスト アドレスは、すべて 1 で構成されているアドレス（255.255.255.255）です。ただし、任意の形式の IP ブロードキャスト アドレスを生成するようにスイッチを設定することもできます。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： Device>**enable**	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： Device#**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	**interface** *interface-id* 例： Device(config)#interface gigabitethernet 1/0/1	インターフェイス コンフィギュレーション モードを開始し、設定するインターフェイスを指定します。
ステップ 4	**ip broadcast-address** *ip-address* 例： Device(config-if)#ip broadcast-address 128.1.255.255	デフォルト値と異なるブロードキャスト アドレス（128.1.255.255 など）を入力します。
ステップ 5	**end** 例： Device(config)#**end**	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
ステップ 6	**show ip interface** [*interface-id*] 例： `Device#show ip interface`	指定されたインターフェイスまたはすべてのインターフェイスのブロードキャストアドレスを確認します。
ステップ 7	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

IP ブロードキャストのフラッディング

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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 forward-protocol spanning-tree** 例： `Device(config)#ip forward-protocol spanning-tree`	ブリッジング スパニングツリー データベースを使用し、UDPデータグラムをフラッディングします。
ステップ 4	**ip forward-protocol turbo-flood** 例： `Device(config)#ip forward-protocol turbo-flood`	スパニングツリーデータベースを使用し、UDPデータグラムのフラッディングを高速化します。
ステップ 5	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
ステップ 6	**show running-config** 例： `Device#show running-config`	入力を確認します。
ステップ 7	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

IP アドレスのモニタリングおよびメンテナンス

特定のキャッシュ、テーブル、またはデータベースの内容が無効になっている場合、または無効である可能性がある場合は、**clear** 特権 EXEC コマンドを使用し、すべての内容を削除できます。次の表に、内容をクリアするために使用するコマンドを示します。

表 *6:* キャッシュ、テーブル、データベースをクリアするコマンド

コマンド	目的
clear arp-cache	IP ARP キャッシュおよび高速スイッチング キャッシュを す。
clear host {*name* | *****}	ホスト名およびアドレス キャッシュから 1 つまたはすへ リを削除します。
clear ip route {**network** [*mask*] | *****}	IP ルーティング テーブルから 1 つまたは複数のルートを す。

IP ルーティング テーブル、キャッシュ、データベースの内容、ノードへの到達可能性、ネットワーク内のパケットのルーティング パスなど、特定の統計情報を表示できます。次の表に、IP 統計情報を表示するために使用する特権 EXEC コマンドを示します。

表 *7:* キャッシュ、テーブル、データベースを表示するコマンド

コマンド	目的
show arp	ARP テーブル内のエントリを表示します。
show hosts	デフォルトのドメイン名、検索サービスの方式、サーバー ホ よびキャッシュに格納されているホスト名とアドレスのリス ます。

コマンド	目的
show ip aliases	TCP ポートにマッピングされた IP アドレスを表示します（
show ip arp	IP ARP キャッシュを表示します。
show ip interface [*interface-id*]	インターフェイスの IP ステータスを表示します。
show ip irdp	IRDP 値を表示します。
show ip masks *address*	ネットワーク アドレスに対して使用されるマスクおよび するサブネット番号を表示します。
show ip redirects	デフォルト ゲートウェイのアドレスを表示します。
show ip route [*address* [*mask*]]	[*protocol*]	ルーティング テーブルの現在の状態を表示します。
show ip route summary	サマリー形式でルーティング テーブルの現在のステータス

IP ユニキャスト ルーティングの設定方法

IP ユニキャスト ルーティングのイネーブル化

デフォルトで、デバイスはレイヤ 2 スイッチングモード、IP ルーティングはディセーブルとなっています。デバイスのレイヤ 3 機能を使用するには、IP ルーティングをイネーブルにする必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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 routing** 例： `Device(config)# ip routing`	IP ルーティングを有効にします。

	コマンドまたはアクション	目的
ステップ 4	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `Device#show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

IP ルーティングの有効化の例

次に、ルーティングプロトコルとしてRIPを使用し、IPルーティングをイネーブルにする例を示します。

```
Device#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)#ip routing
Device(config-router)#end
```

次の作業

ここで、選択したルーティングプロトコルのパラメータを設定できます。具体的な手順は次のとおりです。

- RIP
- OSPF
- EIGRP
- BGP
- ユニキャスト Reverse Path Forwarding
- プロトコル独立機能（任意）

IP ネットワークのモニタリングおよびメンテナンス

特定のキャッシュ、テーブル、またはデータベースのすべての内容を削除できます。特定の統計情報を表示することもできます。

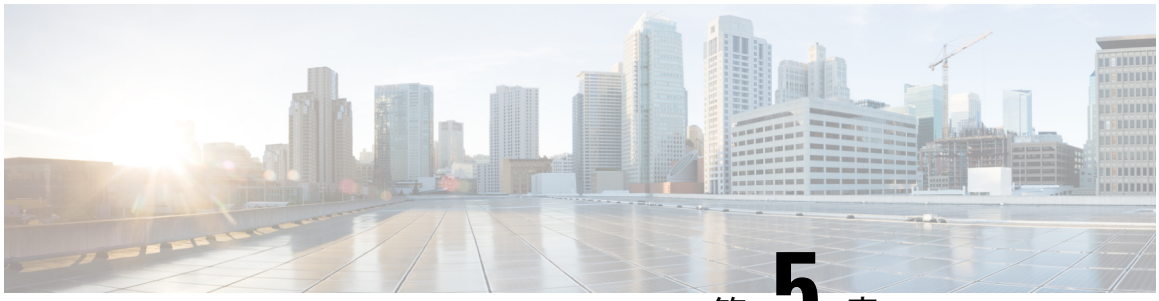
表 *8 : IP* ルートの削除またはルート ステータスの表示を行うコマンド

コマンド	目的
show ip route summary	サマリー形式でルーティング テーブルの現在の 示します。

IP ユニキャスト ルーティングの機能情報

表 *9 : IP* ユニキャスト ルーティングの機能情報

リリース	機能情報
Cisco IOS XE Everest 16.5.1a	この機能が導入されました



第 5 章

IPv6 ユニキャスト ルーティングの設定

IPv6 ユニキャスト ルーティングの設定について

この章では、スイッチに IPv6 ユニキャスト ルーティングを設定する方法について説明します。

（注） この章のすべての IPv6 機能を使用するには、スイッチまたはアクティブスイッチが Network Advantage ライセンスを実行している必要があります。Network Essentials ライセンスを実行しているスイッチは、IPv6 スタティック ルーティングと IPv6 用の RIP をサポートしています。Network Advantage ライセンスを実行しているスイッチは、IPv6 に対し OSPF、EIGRP および BGP をサポートしています。

IPv6 の概要

IPv4 ユーザーは IPv6 に移行することができ、エンドツーエンドのセキュリティ、Quality of Service（QoS）、およびグローバルに一意なアドレスのようなサービスを利用できます。IPv6 アドレス スペースによって、プライベート アドレスの必要性が低下し、ネットワーク エッジの境界ルータで Network Address Translation（NAT; ネットワーク アドレス変換）処理を行う必要性も低下します。

シスコの IPv6 の実装方法については、次の URL を参照してください。

http://www.cisco.com/en/US/products/ps6553/products_ios_technology_home.html

IPv6 およびこの章のその他の機能については、

- 『*Cisco IOS IPv6 Configuration Library*』を参照してください。

- Cisco.com の [Search] フィールドを使用して、Cisco IOS ソフトウェア マニュアルを特定します。たとえば、スタティック ルートについての情報が必要な場合は、[Search] フィールドで *Implementing Static Routes for IPv6* と入力すると、スタティック ルートについて調べられます。

IPv6 のスタティック ルート

スタティック ルートは手動で設定され、2 つのネットワーキング デバイス間のルートを明示的に定義します。スタティック ルートが有効なのは、外部ネットワークへのパスが 1 つしかない小規模ネットワークの場合、または大規模ネットワークで特定のトラフィック タイプにセキュリティを設定する場合です。

IPv6 のスタティック ルーティングの設定（CLI）

IPv6 用のスタティックルートの設定については、「*IPv6* 用のスタティックルーティングの設定」を参照してください。

スタティック ルートの詳細については、Cisco.com で『*Cisco IOS IPv6 Configuration Library*』の「Implementing Static Routes for IPv6」の章を参照してください。

IPv6 ユニキャストのパス MTU ディスカバリ

スイッチはシステム最大伝送単位（MTU）の IPv6 ノードへのアドバタイズおよびパス MTU ディスカバリをサポートします。パス MTU ディスカバリを使用すると、ホストは指定されたデータ パスを通るすべてのリンクの MTU サイズを動的に検出して、サイズに合せて調整できます。IPv6 では、パスを通るリンクの MTU サイズが小さくてパケット サイズに対応できない場合、パケットの送信元がフラグメンテーションを処理します。

ICMPv6

IPv6 のインターネット制御メッセージ プロトコル（ICMP）は、ICMP 宛先到達不能メッセージなどのエラー メッセージを生成して、処理中に発生したエラーや、その他の診断機能を報告します。IPv6 では、ネイバー探索プロトコルおよびパス MTU ディスカバリに ICMP パケットも使用されます。

ネイバー探索

スイッチは、IPv6 対応の NDP、ICMPv6 の最上部で稼働するプロトコル、および NDP をサポートしない IPv6 ステーション対応のスタティック ネイバー エントリをサポートします。IPv6 ネイバー探索プロセスは ICMP メッセージおよび送信請求ノード マルチキャスト アドレスを使用して、同じネットワーク（ローカル リンク）上のネイバーのリンク層アドレスを判別し、ネイバーに到達できるかどうかを確認し、近接ルータを追跡します。

スイッチは、マスク長が 64 未満のルートに対して ICMPv6 リダイレクトをサポートしています。マスク長が 64 ビットを超えるホスト ルートまたは集約ルートでは、ICMP リダイレクトがサポートされません。

ネイバー探索スロットリングにより、IPv6 パケットをルーティングするためにネクスト ホップ転送情報を取得するプロセス中に、スイッチ CPU に不必要な負荷がかかりません。IPv6 パ

ケットのネクストホップがスイッチによってアクティブに解決しようとしている同じネイバーである場合は、そのようなパケットが追加されると、スイッチはそのパケットをドロップします。このドロップにより、CPU に余分な負荷がかからないようになります。

DNS 設定の IPv6 ルータ アドバタイズメント オプション

大部分のインターネット サービスは、ドメイン ネーム サーバー（DNS）名によって識別されます。IPv6 ルータアドバタイズメント（RA）には、IPv6 ホストでの自動 DNS 設定の実行を可能にする次の 2 つのオプションがあります。

- 再帰 DNS サーバー（RDNSS）
- DNS 検索リスト（DNSSL）

RDNSS には、IPv6 ホストでの DNS 名前解決に役立つ再帰 DNS サーバーのアドレスが含まれています。DNS 検索リストは DNS サフィックス ドメイン名のリストであり、IPv6 ホストで DNS クエリ検索を実行する際に使用されます。

DNS 設定の RA オプションの詳細については、IETF RFC 6106 を参照してください。

DNSSL の設定については、『*IP Addressing Services Configuration Guide*』の「*Configuring DNS Search List Using IPv6 Router Advertisement Options*」を参照してください。

デフォルト ルータ プリファレンス

スイッチは、ルータのアドバタイズメント メッセージの拡張機能である、IPv6 Default Router Prefernce（DRP）をサポートします。DRP では、特にホストがマルチホーム構成されていて、ルータが異なるリンク上にある場合に、ホストが適切なルータを選択する機能が向上しました。スイッチは、Route Information Option（RFC 4191）をサポートしません。

IPv6 ホストは、オフリンク宛先へのトラフィック用にルータを選択する、デフォルト ルータ リストを維持します。次に、宛先用に選択されたルータは、宛先キャッシュに格納されます。IPv6 NDP では、到達可能であるルータまたは到達可能性の高いルータが、到達可能性が不明または低いルータよりも優先されます。NDP は、到達可能または到達できる可能性の高いルータとして、常に同じルータを選択するか、またはルータ リストを循環して選択できます。DRP を使用することにより、両方ともが到達可能または到達できる可能性の高い 2 台のルータの一方を他方に対して優先させるよう IPv6 ホストを設定することができます。

DRP for IPv6 の設定については、「*DRP* の設定」を参照してください。

DRP for IPv6 の詳細情報については、Cisco.com の『*Cisco IOS IPv6 Configuration Library*』を参照してください。

IPv6 のポリシーベース ルーティング

ポリシーベース ルーティング（PBR）は、トラフィック フローに定義ポリシーを設定し、ルートにおけるルーティング プロトコルへの依存度を軽くして、パケットのルーティングを柔軟に行えるようにします。したがって、PBR は、ルーティング プロトコルで提供される既存のメカニズムを拡張および補完することにより、ルーティングの制御を強化します。PBR を使用すると、IPv6 precedence を設定できます。単純なポリシーでは、これらのタスクのいずれかを使

用し、複雑なポリシーでは、これらすべてのタスクを使用できます。高コストリンク上のプライオリティ トラフィックなど、特定のトラフィックのパスを指定することもできます。

PBR for IPv6 は、転送される IPv6 パケットおよび送信される IPv6 パケットの両方に適用できます。転送されるパケットの場合、PBR for IPv6 は、次の転送パスでサポートされる IPv6 入力インターフェイス機能として実装されます。

- プロセス
- シスコ エクスプレス フォワーディング（旧称 CEF）
- 分散型シスコ エクスプレス フォワーディング

ポリシーは、IPv6アドレス、ポート番号、プロトコル、またはパケットのサイズに基づいて作成できます。

PBR を使用すると、次の処理を実行できます。

- 拡張アクセス リスト基準に基づいてトラフィックを分類する。リストにアクセスし、次に一致基準を設定します。
- 差別化されたサービス クラスを有効にする機能をネットワークに与える IPv6 precedence ビットを設定する。
- 特定のトラフィック エンジニアリング パスにパケットをルーティングする。ネットワークを介して特定の Quality of Service（QoS）を得るためにパケットをルーティングする必要がある場合があります。

PBR を使用すると、ネットワークのエッジでパケットを分類およびマーキングできます。PBR では、precedence 値を設定することにより、パケットをマーキングします。precedence 値は、ネットワーク コアにあるデバイスが適切な QoS をパケットに適用するために直接使用でき、これにより、パケットの分類がネットワーク エッジで維持されます。

PBR for IPv6 の有効化については、「ローカル *PBR for IPv6* の有効化」を参照してください。

インターフェイスの IPv6 PBR の有効化については、「インターフェイスでの *IPv6 PBR* の有効化」を参照してください。

サポートされていない IPv6 ユニキャスト ルーティング機能

スイッチは、次の IPv6 機能をサポートしません。

- サイトローカルアドレス宛ての IPv6 パケット
- IPv4/IPv6 や IPv6/IPv4 などのトンネリング プロトコル
- IPv4/IPv6 または IPv6/IPv4 トンネリング プロトコルをサポートするトンネル エンドポイントとしてのスイッチ
- IPv6 Web Cache Communication Protocol（WCCP）

IPv6 機能の制限

IPv6 はスイッチのハードウェアに実装されるため、ハードウェアメモリ内の IPv6 圧縮アドレスによる制限がいくつか発生します。ハードウェアの制限により、機能の一部が失われ、一部の機能が制限されます。たとえば、スイッチはハードウェアでソースルーテッド IPv6 パケットに QoS 分類を適用できません。

IPv6 とスイッチ スタック

スイッチにより、スタック全体で IPv6 転送がサポートされ、アクティブスイッチで IPv6 ホスト機能がサポートされます。アクティブスイッチは IPv6 ユニキャスト ルーティング プロトコルを実行してルーティングテーブルを計算します。スタック メンバー スイッチはテーブルを受信して、転送用にハードウェア IPv6 ルートを作成します。アクティブスイッチは、すべての IPv6 アプリケーションも実行します。

新しいスイッチがアクティブスイッチになる場合、新しいマスターは IPv6 ルーティングテーブルを再計算してこれをメンバースイッチに配布します。新しいアクティブスイッチが選択中およびリセット中の間には、スイッチスタックによる IPv6 パケットの転送は行われません。スタック MAC アドレスが変更され、これによって IPv6 アドレスが変更されます。**ipv6 address** *ipv6-prefix/prefix length* **eui-64** インターフェイス コンフィギュレーション コマンドを使用して、拡張固有識別子（EUI）でスタック IPv6 アドレスを指定する場合、アドレスは、インターフェイス MAC アドレスに基づきます。「*IPv6* アドレッシングの設定と *IPv6* ルーティングの有効化」を参照してください。

スタック上で永続的な MAC アドレスを設定し、アクティブスイッチが変更された場合、スタック MAC アドレスは、約 4 分間、変更されません。

IPv6 アクティブスイッチおよびメンバーの機能は次のとおりです。

- アクティブスイッチ：
  - IPv6 ルーティングプロトコルの実行
  - ルーティング テーブルの生成
  - IPv6 用の分散型シスコ エクスプレス フォワーディングを使用するメンバースイッチにルーティングテーブルを配布します。
  - IPv6 ホスト機能および IPv6 アプリケーションの実行
- メンバースイッチ：
  - アクティブスイッチから IPv6 用のシスコ エクスプレス フォワーディングのルーティングテーブルを受信します。
  - ハードウェアへのルートのプログラミング

（注） IPv6パケットに例外（IPv6オプション）がなく、スタック内のスイッチでハードウェア リソースが不足していない場合、IPv6 パケットがスタック全体にわたってハードウェアでルーティングされます。

- アクティブスイッチの再選択で IPv6 用のシスコ エクスプレス フォワーディングのテーブルをフラッシュします。

IPv6 のデフォルト設定

表 *10 : IPv6* のデフォルト設定

機能	デフォルト設定
SDM テンプレート	デフォルトは拡張テンプレート
IPv6 ルーティング	すべてのインターフェイスでグローバルに無効
IPv6 用 Cisco Express Forwarding または IPv6 用 distributed Cisco Express Forwarding（dCEF; 分散型シスコ エクスプレス フォワーディング）	無効（IPv4 Cisco Express Forwarding および distributed Cisco Express Forwarding（dCEF; 分散型シスコ エクスプレス フォワーディング）はデフォルトでは有効） （注） IPv6ルーティングを有効にすると、IPv6 用 Cisco Express Forwarding および IPv6 用 distributed Cisco Express Forwarding（dCEF; 分散型シスコエクスプレス フォワーディング）は自動的に有効になります。
IPv6 アドレス	未設定

IPv6 ユニキャストルーティングの設定方法

ここでは、IPv6ユニキャストルーティングに関して使用できるさまざまな設定オプションを示します。

IPv6 アドレッシングの設定と IPv6 ルーティングの有効化

ここでは、IPv6 アドレスを各レイヤ 3 インターフェイスに割り当てて、IPv6 トラフィックをスイッチ上でグローバル転送する方法を説明します。

（注） IPv6 ルーティングはデフォルトでは有効になっていないため、**ipv6 unicast-routing** コマンドを使用して有効にする必要があります。

スイッチ上の IPv6 を設定する前に、次の注意事項に従ってください。

- スイッチでは、この章で説明されたすべての機能がサポートされるわけではありません。「サポートされていない IPv6 ユニキャスト ルーティング機能」を参照してください。
- **ipv6 address** インターフェイス コンフィギュレーション コマンドでは、16 ビット値を使用したコロン区切りの 16 進形式で指定したアドレスで *ipv6-address* 変数および *ipv6-prefix* 変数を入力する必要があります。*prefix-length* 変数（スラッシュ（/）で始まる）は、プレフィックス（アドレスのネットワーク部分）を構成するアドレスの上位連続ビット数を示す 10 進値です。

インターフェイス上の IPv6 トラフィックを転送するには、そのインターフェイス上でグローバル IPv6 アドレスを設定する必要があります。インターフェイス上で IPv6 アドレスを設定すると、リンクローカルアドレスの設定、およびそのインターフェイスに対する IPv6 のアクティブ化が自動的に行われます。設定されたインターフェイスは、次に示す、該当リンクの必須マルチキャスト グループに自動的に参加します。

- インターフェイスに割り当てられた各ユニキャストアドレスの送信要求ノードマルチキャスト グループ FF02:0:0:0:0:1:ff00::/104（このアドレスはネイバー探索プロセスで使用される）
- 全ノード向けリンクローカルマルチキャストグループ FF02::1
- 全ルータ向けリンクローカルマルチキャストグループ FF02::2

IPv6 アドレスをインターフェイスから削除するには、 **no ipv6 address** *ipv6-prefix/prefix length* **eui-64** または **no ipv6 address** *ipv6-address* **link-local** インターフェイス コンフィギュレーション コマンドを使用します。インターフェイスから手動で設定したすべての IPv6 アドレスを削除するには、**no ipv6 address** インターフェイス コンフィギュレーション コマンドを引数なしで使用します。IPv6 アドレスが明確に設定されていないインターフェイスで IPv6 処理を無効にするには、**no ipv6 enable** インターフェイス コンフィギュレーション コマンドを使用します。IPv6 ルーティングをグローバルに無効にするには、**no ipv6 unicast-routing** グローバル コンフィギュレーション コマンドを使用します。

IPv6 ルーティングの設定の詳細については、Cisco.com で『*Cisco IOS IPv6 Configuration Library*』の「Implementing Addressing and Basic Connectivity for IPv6」の章を参照してください。

IPv6 アドレスをレイヤ 3 インターフェイスに割り当て、IPv6 ルーティングを有効にするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `# 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	**sdm prefer access** 例： `(config)# sdm prefer access`	スイッチをアクセステンプレートに設定します。
ステップ 4	**end** 例： `(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**reload** 例： `# reload`	オペレーティング システムをリロードします。
ステップ 6	**configure terminal** 例： `# configure terminal`	スイッチのリロード後、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	**interface** *interface-id* 例： `(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ3インターフェイスを指定します。インターフェイスは物理インターフェイス、スイッチ仮想インターフェイス（SVI）、またはレイヤ 3 EtherChannel に設定できます。
ステップ 8	**no switchport** 例： `(config-if)# no switchport`	レイヤ 2 コンフィギュレーション モードからインターフェイスを削除します（物理インターフェイスの場合）。

	コマンドまたはアクション	目的

ステップ 9	次のいずれかを使用します。
	• **ipv6 address** *ipv6-prefix/prefix length* **eui-64**
	• **ipv6 address** *ipv6-address/prefix length*
	• **ipv6 address** *ipv6-address* **link-local**
	• **ipv6 enable**
	• **ipv6 address** *WORD*
	• **ipv6 address** *autoconfig*
	• **ipv6 address** *[dhcp]*
	例：
	`(config-if)# ipv6 address 2001:0DB8:c18:1::/64 eui 64`
	`(config-if)# ipv6 address 2001:0DB8:c18:1::/64`
	`(config-if)# ipv6 address 2001:0DB8:c18:1:: link-local`
	`(config-if)# ipv6 enable`
		• IPv6 アドレスの下位 64 ビットの拡張固有識別子（EUI）を使用して、グローバル IPv6 アドレスを指定します。ネットワーク プレフィックスだけを指定します。最終の 64 ビットは、スイッチの MAC アドレスから自動的に計算されます。これにより、インターフェイス上で IPv6 処理が有効になります。
		• インターフェイスの IPv6 アドレスを手動で設定します。
		• インターフェイスで IPv6 が有効な場合に自動設定されるリンクローカルアドレスでなく、インターフェイス上の特定のリンクローカルなアドレスを使用するように指定します。このコマンドにより、インターフェイス上で IPv6 処理が有効になります。
		• インターフェイスに IPv6 リンクローカルアドレスを自動設定し、インターフェイスでの IPv6 処理を有効にします。リンクローカルアドレスを使用できるのは、同じリンク上のノードと通信する場合だけです。

ステップ 10	**exit** 例： `(config-if)# exit`	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 routing** 例： `(config)# ip routing`	スイッチ上で IP ルーティングをイネーブルにします。
ステップ 12	**ipv6 unicast-routing** 例： `(config)# ipv6 unicast-routing`	IPv6 ユニキャスト データ パケットの転送を有効にします。
ステップ 13	**end** 例： `(config)# end`	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
ステップ 14	**show ipv6 interface** *interface-id* 例： `# show ipv6 interface gigabitethernet 1/0/1`	入力を確認します。
ステップ 15	**copy running-config startup-config** 例： `# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

IPv4 および IPv6 プロトコル スタックの設定

IPv4 および IPv6 を両方サポートし、IPv6 ルーティングがイネーブルになるようにレイヤ 3 インターフェイスを設定するには、特権 EXEC モードで次の手順を実行します。

（注） IPv6 アドレスが設定されていないインターフェイスで IPv6 処理をディセーブルにするには、インターフェイス コンフィギュレーション モードで **no ipv6 enable** コマンドを使用します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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 routing** 例： `Device(config)# ip routing`	スイッチ上でルーティングを有効にします。
ステップ 4	**ipv6 unicast-routing** 例： `Device(config)# ipv6 unicast-routing`	スイッチ上で IPv6 データ パケットの転送を有効にします。

	コマンドまたはアクション	目的
ステップ 5	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ3インターフェイスを指定します。
ステップ 6	**no switchport** 例： `Device(config-if)# no switchport`	レイヤ 2 コンフィギュレーション モードからインターフェイスを削除します（物理インターフェイスの場合）。
ステップ 7	**ip address** *ip-address mask* [**secondary**] 例： `Device(config-if)# ip address 10.1.2.3 255.255.255`	インターフェイスのプライマリまたはセカンダリ IPv4 アドレスを指定します。
ステップ 8	次のいずれかを使用します。 • **ipv6 address** *ipv6-prefix/prefix length* **eui-64** • **ipv6 address** *ipv6-address/prefix length* • **ipv6 address** *ipv6-address* **link-local** • **ipv6 enable** • **ipv6 address***WORD* • **ipv6 address***autoconfig* • **ipv6 address***dhcp*	• グローバル IPv6 アドレスを指定します。ネットワーク プレフィックスだけを指定します。最終の 64 ビットは、スイッチの MAC アドレスから自動的に計算されます。 • インターフェイスで IPv6 が有効な場合に自動設定されるリンクローカルアドレスでなく、インターフェイス上のリンクローカルアドレスを使用するように指定します。 • インターフェイスに IPv6 リンクローカルアドレスを自動設定し、インターフェイスでの IPv6 処理を有効にします。リンクローカルアドレスを使用できるのは、同じリンク上のノードと通信する場合だけです。 （注） インターフェイスから手動で設定したすべての IPv6 アドレスを削除するには、**no ipv6 address** インターフェイス コンフィギュレーション コマンドを引数なしで使用します。
ステップ 9	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 10	次のいずれかを使用します。 • **show interface** *interface-id* • **show ip interface** *interface-id* • **show ipv6 interface** *interface-id*	入力を確認します。

	コマンドまたはアクション	目的
ステップ 11	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

再帰 DNS サーバー（RDNSS）の設定

最大 8 つの DNS サーバーを設定し、ルータ アドバタイズメントを使用してアドバタイズできます。また、このコマンドの **no** 形式を使用して、アドバタイジングリストから 1 つ以上の DNS サーバーを削除できます。

始める前に

正しい VDC 内にいることを確認します（あるいは、**switchto vdc** コマンドを使用します）。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> 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： `Device# 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	**interface ethernet** *number* 例： `Device(config)# interface ethernet 3/3`	インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**ipv6 nd ra dns server** *ipv6-addr* [*rdnss-life* | **infinite**] **sequence** *sequence-num* 例： `Device(config-if)# ipv6 nd ra dns server 1::1 1000 sequence 0`	再帰 DNS サーバーを設定します。サーバーの有効期間と順序を指定できます。
ステップ 5	**show ipv6 nd ra dns server** [**interface** *interface*] 例： `Device(config-if)# show ipv6 nd ra dns server`	（任意）設定した RDNSS リストを表示します。
ステップ 6	**ipv6 nd ra dns server suppress** 例： `Device(config-if)# ipv6 nd ra dns server suppress`	（任意）設定したサーバー リストをディセーブルにします。

デフォルト ルータ プリファレンス（DRP）の設定

ルータアドバタイズメント（RA）メッセージは、**ipv6 nd router-preference** インターフェイス コンフィギュレーション コマンドによって設定されるデフォルト ルータ プリファレンス（DRP）とともに送信されます。DRP が設定されていない場合は、RA はプリファレンス「中」とともに送信されます。

リンク上の 2 つのルータが等価ではあっても、等コストではないルーティングを提供する可能性がある場合、およびポリシーでホストがいずれかのルータを選択するよう指示された場合は、DRP が有効です。

IPv6 の DRP の設定の詳細については、Cisco.com で『*Cisco IOS IPv6 Configuration Library*』の「Implementing IPv6 Addresses and Basic Connectivity」の章を参照してください。

インターフェイス上のルータに DRP を設定するには、特権 EXEC モードで次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始して、DRP を指定するレイヤ 3 インターフェイスを特定します。
ステップ 4	**ipv6 nd router-preference** {**high** | **medium** | **low**} 例： `Device(config-if)# ipv6 nd router-preference medium`	スイッチ インターフェイス上のルータに DRP を指定します。
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show ipv6 interface** 例： `Device# show ipv6 interface`	設定を確認します。

	コマンドまたはアクション	目的
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

IPv6 ICMP レート制限の設定

ICMP レート制限はデフォルトで有効です。エラー メッセージのデフォルト間隔は 100 ミリ秒、デフォルト バケット サイズ（バケットに格納される最大トークン数）は 10 です。

ICMP のレート制限パラメータを変更するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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	**ipv6 icmp error-interval** *interval* [*bucketsize*] 例： `Device(config)# ipv6 icmp error-interval 50 20`	IPv6 ICMP エラー メッセージの間隔とバケット サイズを設定します。 • *interval*：バケットに追加されるトークンの間隔（ミリ秒）。指定できる範囲は 0 ～ 2147483647 ミリ秒です。 • *bucketsize*：（任意）バケットに格納される最大トークン数。指定できる範囲は 1 ～ 200 です。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show ipv6 interface** [*interface-id*] 例： `Device# show ipv6 interface gigabitethernet0/1`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

IPv6 用のシスコ エクスプレス フォワーディングおよび分散型シスコ エクスプレス フォワーディングの設定

シスコ エクスプレス フォワーディングは、ネットワークパフォーマンスを最適化するためのレイヤ 3 IP スイッチングテクノロジーです。シスコ エクスプレス フォワーディングには高度な IP 検索および転送アルゴリズムが実装されているため、レイヤ 3 スイッチングのパフォーマンスを最大化できます。高速スイッチング ルート キャッシュよりも CPU にかかる負担が少ないため、CEF はより多くの CPU 処理能力をパケット転送に振り分けることができます。スイッチスタックでは、ハードウェアによって分散型シスコ エクスプレス フォワーディングが使用されます。IPv4 用のシスコ エクスプレス フォワーディングおよび分散型シスコ エクスプレス フォワーディングはデフォルトで有効になっています。IPv6 用のシスコ エクスプレス フォワーディングおよび分散型シスコ エクスプレス フォワーディングはデフォルトでは無効になっていますが、IPv6 ルーティングを設定すると自動的に有効になります。

IPv6 ルーティングの設定を解除すると IPv6 用のシスコ エクスプレス フォワーディングおよび分散型シスコ エクスプレス フォワーディングは自動的に無効になります。IPv6 用のシスコ エクスプレス フォワーディングおよび分散型シスコ エクスプレス フォワーディングを設定で無効にすることはできません。IPv6 の状態を確認するには、特権 EXEC モードで **show ipv6 cef** コマンドを入力します。

IPv6 ユニキャストパケットをルーティングするには、最初に **ipv6 unicast-routing** グローバル コンフィギュレーション コマンドを使用して、IPv6 ユニキャストパケットの転送をグローバルに設定してから、インターフェイス コンフィギュレーション モードで **ipv6 address** コマンドを使用して、特定のインターフェイスに IPv6 アドレスおよび IPv6 処理を設定する必要があります。

シスコ エクスプレス フォワーディングおよび分散型シスコ エクスプレス フォワーディングの設定の詳細については、Cisco.com の『*Cisco IOS IPv6 Configuration Library*』を参照してください。

IPv6 のスタティック ルーティングの設定

スタティック IPv6 ルーティングの設定の詳細については、Cisco.com で『*Cisco IOS IPv6 Configuration Library*』の「Implementing Static Routes for IPv6」の章を参照してください。

スタティック IPv6 ルーティングを設定するには、次の手順を実行します。

始める前に

ip routing グローバル コンフィギュレーション コマンドを使用してルーティングをイネーブルにし、グローバル コンフィギュレーション モードで **ipv6 unicast-routing** コマンドを使用して IPv6 パケットの転送をイネーブルにします。また、インターフェイスに IPv6 アドレスを設定して少なくとも 1 つのレイヤ 3 インターフェイス上で IPv6 をイネーブルにする必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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	**ipv6 route** *ipv6-prefix/prefix length* {*ipv6-address* | *interface-id* [*ipv6-address*]} [*administrative distance*] 例： `Device(config)# ipv6 route 2001:0DB8::/32 gigabitethernet2/0/1 130`	スタティック IPv6 ルートを設定します。 • *ipv6-prefix*：スタティック ルートの宛先となる IPv6 ネットワーク。スタティック ホスト ルートを設定する場合は、ホスト名も設定できます。 • */prefix length*：IPv6 プレフィックスの長さ。プレフィックス（アドレスのネットワーク部分）を構成するアドレスの上位連続ビット数を示す 10 進値です。10 進数値の前にスラッシュ記号が必要です。 • *ipv6-address*：指定したネットワークに到達するために使用可能なネクスト ホップの IPv6 アドレス。ネクスト ホップの IPv6 アドレスを直接接続する必要はありません。再帰処理が実行されて、直接接続されたネクスト ホップの IPv6 アドレスが検出されます。このアドレスはRFC 2373 に記載された形式（16 ビット値を使用したコロン区切りの 16 進表記で指定）で設定する必要があります。 • *interface-id*：Point-To-Point（ポイントツーポイント）インターフェイスおよびブロードキャスト インターフェイスからのダイレクト スタティック ルートを指定します。ポイントツーポイントインターフェイスの場合、ネクストホップの IPv6 アドレスを指定する必要はありません。ブロードキャストインターフェイスの場合は、常にネクストホップの IPv6 アドレスを指定するか、または指定したプレフィックスをリンクに割り当てて、リンクローカルアドレスをネクストホップとして指定する必要があります。パケットの送信先となるネクストホップの IPv6 アドレスを指定することもできます。

	コマンドまたはアクション	目的
		（注） リンクローカルアドレスをネクストホップとして使用する場合は、*interface-id* を指定する必要があります（リンクローカルのネクストホップを隣接ルータに設定する必要もあります）。 • *administrative distance*：（任意）アドミニストレーティブ ディスタンス。指定できる範囲は 1 ～ 254 です。デフォルト値は 1 で、この場合、接続されたルートを除くその他のどのルート タイプよりも、スタティック ルートが優先します。フローティング スタティック ルートを設定する場合は、ダイナミック ルーティング プロトコルよりも大きなアドミニストレーティブ ディスタンスを使用します。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

ステップ 5	次のいずれかを使用します。
	• **show ipv6 static** [*ipv6-address* | *ipv6-prefix/prefix length*] [**interface** *interface-id*] [**detail**]][**recursive**] [**detail**]
	• **show ipv6 route static** [*updated*]
	例：
	`Device# show ipv6 static 2001:0DB8::/32 interface gigabitethernet2/0/1`
	または
	`Device# show ipv6 route static`
		IPv6 ルーティング テーブルの内容を表示して、設定を確認します。
		• **interface** *interface-id*：（任意）出力インターフェイスとして指定されたインターフェイスを含むスタティック ルートのみを表示します。
		• **recursive**：（任意）再帰スタティック ルートのみを表示します。**recursive** キーワードは **interface** キーワードと相互に排他的です。ただし、コマンド構文に IPv6 プレフィックスが指定されているかどうかに関係なく、使用できます。
		• **detail**：（任意）次に示す追加情報を表示します。
		• 有効な再帰ルートの場合、出力パス セットおよび最大分解深度
		• 無効なルートの場合、ルートが無効な理由

ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

インターフェイスでの IPv6 PBR の有効化

IPv6 のポリシーベース ルーティング（PBR）を有効にするには、パケットの一致基準と目的のポリシールーティング アクションを指定する、ルート マップを作成する必要があります。次に、そのルートマップを必要なインターフェイスに関連付けます。指定されたインターフェイスに到着し、match 句に一致するすべてのパケットに対して、PBR が実行されます。

PBR では、**set vrf** コマンドにより Virtual Routing and Forwarding（VRF）インスタンスとインターフェイスアソシエーションを切り離し、既存の PBR またはルートマップ設定を使用して、アクセスコントロールリスト（ACL）ベースの分類に基づいて VRF を選択できるようになります。このコマンドは、1 つのルータに複数ルーティング テーブルを提供し、ACL 分類に基づいてルートを選択できるようにします。ルータは、ACL に基づいてパケットを分類し、ルーティング テーブルを選択し、宛先アドレスを検索し、パケットをルーティングします。

PBR for IPv6 を有効にするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] 例： `Device(config)# route-map rip-to-ospf permit`	ルーティング プロトコル間でルートを再配布する条件を定義するか、ポリシー ルーティングを有効にしてルート 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 4	次のいずれかを実行します。
	• **match length** *minimum-length maximum-length*
	• **match ipv6 address** {**prefix-list** *prefix-list-name* | *access-list-name*}
	例：
	`Device(config-route-map)# match length 3 200`
	例：
	`Device(config-route-map)# match ipv6 address marketing`
		一致基準を指定します。
		• 次のうちの任意の項目またはすべてを指定できます。
		• レベル 3 のパケット長とのマッチング。
		• 指定された IPv6 アクセス リストとのマッチング。
		• **match** コマンドを指定しない場合、ルートマップはすべてのパケットに適用されます。

ステップ 5	次のいずれかを実行します。 • **set ipv6 next-hop** *global-ipv6-address* [*global-ipv6-address...*]	基準に一致したパケットに適用するアクション（1 つまたは複数）を指定します。

	コマンドまたはアクション	目的

	• **set interface** *type number* [*...type number*]
	• **set ipv6 default next-hop** *global-ipv6-address* [*global-ipv6-address...*]
	• **set vrf** *vrf-name*
	例：
	`Device(config-route-map)# set ipv6 next-hop 2001:DB8:2003:1::95`
	例：
	`Device(config-route-map)# set ipv6 default next-hop 2001:DB8:2003:1::95`
	例：
	`Device(config-route-map)# set vrf vrfname`
		• 次のうちの任意の項目またはすべてを指定できます。
		• パケットのルーティング先となるネクストホップを設定します（ネクストホップは隣接している必要があります）。
		• 宛先への明示的なルートがない場合に、パケットのルーティング先となるネクストホップを設定します。

ステップ 6	**exit** 例： `Device(config-route-map)# exit`	ルート マップ インターフェイス 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 モードに戻ります。
ステップ 7	**interface** *type number* 例： `Device(config)# interface FastEthernet 1/0`	インターフェイスのタイプと番号を指定し、ルータをインターフェイス コンフィギュレーション モードにします。
ステップ 8	**ipv6 policy route-map** *route-map-name* 例： `Device(config-if)# ipv6 policy-route-map interactive`	インターフェイスで IPv6 PBR に使用するルート マップを特定します。
ステップ 9	**end** 例： `Device(config-if)# end`	インターフェイス 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。

ローカル PBR for IPv6 のイネーブル化

デバイスが生成したパケットに対して、通常はポリシーによるルーティングは行われません。これらのパケットのためのローカル IPv6 ポリシーベース ルーティング（PBR）をイネーブルにするには、この作業を実行して、どのルート マップをデバイスで使用するべきかを示します。

ローカル PBR for IPv6 を有効にするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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	**ipv6 local policy route-map** *route-map-name* 例： `Device(config)# ipv6 local policy route-map pbr-src-90`	デバイスによって生成されるパケットに対する IPv6 PBR を設定します。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

IPv6 の表示

次のコマンドの構文および使用方法の詳細については、Cisco IOS のコマンド リファレンスを参照してください。

表 *11 : IPv6* をモニタリングするコマンド

コマンド	目的
show ipv6 access-list	アクセス リストのサマリーを表示します。
show ipv6 cef	IPv6 の Cisco エクスプレス フォワーディングを表示します。
show ipv6 interface *interface-id*	IPv6 インターフェイスのステータスと設定を表示します。
show ipv6 mtu	宛先キャッシュごとに IPv6 MTU を表示します。
show ipv6 neighbors	IPv6 ネイバー キャッシュ エントリを表示します。
show ipv6 prefix-list	IPv6 プレフィックス リストを表示します。
show ipv6 protocols	スイッチの IPv6 ルーティング プロトコルのリストを表示します。

コマンド	目的
show ipv6 rip	IPv6 RIP ルーティング プロトコル ステータスを表示します。
show ipv6 route	IPv6 ルート テーブル エントリを表示します。
show ipv6 static	IPv6 スタティック ルートを表示します。
show ipv6 traffic	IPv6 トラフィックの統計情報を表示します。

IPv6 ユニキャスト ルーティングの設定例

ここでは、IPv6ユニキャストルーティングに関して使用できるさまざまな設定例を示します。

例：IPv4 および IPv6 プロトコルスタックの設定

次に、インターフェイス上で IPv4 および IPv6 ルーティングを有効にする例を示します。

```
Device> enable
Device# configure terminal
Device(config)# ip routing
Device(config)# ipv6 unicast-routing
Devoce(config)# interface fastethernet1/0/11
Device(config-if)# no switchport
Device(config-if)# ip address 192.168.99.1 255.255.255.0
Device(config-if)# ipv6 address 2001:0DB8:c18:1::/64 eui 64
Device(config-if)# end
```

例：RDNSS の設定

次の例は、Ethernet 3/3 に再帰 DNS サーバー リストを設定し、同じであることを確認する方法を示しています。

```
Device> enable
Device# configure terminal
Device(config)# interface ethernet 3/3
Device(config-if)# ipv6 nd ra dns server 1::1 1000 sequence 0
Device(config-if)# ipv6 nd ra dns server 2::1 infinite sequence 1
Device(config-if)# exit

Device(config)# show ipv6 nd ra dns server

Recursive DNS Server List on: mgmt0
Suppress DNS Server List: No
Recursive DNS Server List on: Ethernet3/3
  Suppress DNS Server List: No
  DNS Server 1: 1::1 Lifetime:1000 seconds Sequence:0
  DNS Server 2: 2::1 Infinite Sequence:1
```

例：DNSSL の設定

次の例は、Ethernet 3/3 に DNS 検索リストを設定し、同じであることを確認する方法を示しています。

```
Device> enable
Device# configure terminal
Device(config)# interface ethernet 3/3
Device(config-if)# ipv6 nd ra dns search-list cisco.com 100 sequence 1
Device(config-if)# ipv6 nd ra dns search-list ind.cisco.com 100 sequence 2
Device(config-if)# exit

Device(config)# show ipv6 nd ra dns search-list

DNS Search List on: mgmt0
Suppress DNS Search List: No
 DNS Search List on: Ethernet3/3
  Suppress DNS Search List: No
  DNS Server 1:cisco.com 100 Sequence:1
  DNS Server 2:ind.cisco.com 100 Sequence:2
```

例：デフォルト ルータ プリファレンスの設定

次に、インターフェイス上のルータに高い DRP を設定する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# ipv6 nd router-preference high
Device(config-if)# end
```

例：IPv6 ICMP レート制限の設定

次に、IPv6 ICMP エラー メッセージ間隔を 50 ミリ秒に、バケット サイズを 20 トークンに設定する例を示します。

```
Device> enable
Device# configure terminal
Device(config)#ipv6 icmp error-interval 50 20
```

例：IPv6 のスタティックルーティングの設定

次に、アドミニストレーティブ ディスタンスが 130 のフローティング スタティック ルートをインターフェイスに設定する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# ipv6 route 2001:0DB8::/32 gigabitethernet 0/1 130
```

例：インターフェイスでの PBR のイネーブル化

次の例では、pbr-dest-1 という名前のルート マップを作成および設定し、パケット一致基準および目的のポリシー ルーティング アクションを指定します。次に、PBR が GigabitEthernet インターフェイス 0/0/1 で有効にされます。

```
Device> enable
Device# configure terminal
Device(config)# ipv6 access-list match-dest-1
Device(config)# permit ipv6 any 2001:DB8:2001:1760::/32
Device(config)# route-map pbr-dest-1 permit 10
Device(config)# match ipv6 address match-dest-1
Device(config)# set interface GigabitEthernet 0/0/0
Device(config)# interface GigabitEthernet0/0/1
Device(config-if)# ipv6 policy-route-map interactive
```

例：ローカル PBR for IPv6 のイネーブル化

次の例では、宛先 IPv6 アドレスがアクセス リスト pbr-src-90 で許可されている IPv6 アドレス範囲に一致するパケットが、IPv6 アドレス 2001:DB8:2003:1::95 のデバイスに送信されています。

```
Device> enable
Device# configure terminal
Device(config)# ipv6 access-list src-90
Device(config)# permit ipv6 host 2001:DB8:2003::90 2001:DB8:2001:1000::/64
Device(config)# route-map pbr-src-90 permit 10
Device(config)# match ipv6 address src-90
Device(config)# set ipv6 next-hop 2001:DB8:2003:1::95
Device(config)# ipv6 local policy route-map pbr-src-90
```

例：IPv6 の表示

次に、**show ipv6 interface** コマンドの出力の例を示します。

```
Device> enable
Device# show ipv6 interface
Vlan1 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::20B:46FF:FE2F:D940
  Global unicast address(es):
    3FFE:C000:0:1:20B:46FF:FE2F:D940, subnet is 3FFE:C000:0:1::/64 [EUI]
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF2F:D940
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND advertised reachable time is 0 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
<output truncated>
```

その他の参考資料

標準および RFC

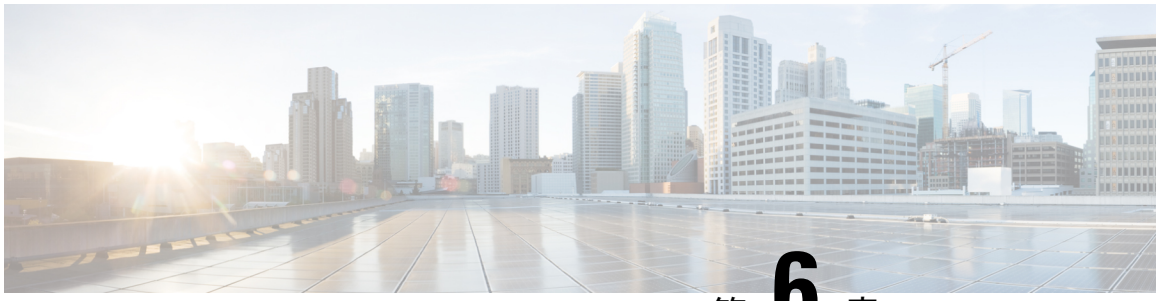
標準/RFC	タイトル
RFC 5453	予約済み *IPv6* インターフェイス識別子

機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。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 *12 : IPv6* ユニキャストおよびルーティングの機能情報

機能名	リリース	機能情報
IPv6 ユニキャストおよびルーティング	Cisco IOS XE Everest 16.5.1a	ユニキャストおよびルーティング機能が IPv6 に対してサポートされました。
RFC 5453	Cisco IOS XE Gibraltar 16.11.1	RFC 5453 がサポートされています。
DNS 設定の IPv6 ルータ アドバタイズメント オプション	Cisco IOS XE Gibraltar 16.11.1	この機能が導入されました。



第 6 章

RIP の設定

RIP 情報

RIP は、小規模な同種ネットワーク間で使用するために作成された Interior Gateway Protocol（IGP）です。RIP は、ブロードキャスト ユーザー データグラム プロトコル（UDP）データパケットを使用してルーティング情報を交換するディスタンスベクトル ルーティング プロトコルです。このプロトコルは RFC 1058 に文書化されています。RIP の詳細については、『*IP Routing Fundamentals*』（Cisco Press 刊）を参照してください。

（注） RIP は Network Essentials 機能セットでサポートされています。

スイッチは RIP を使用し、30 秒ごとにルーティング情報アップデート（アドバタイズメント）を送信します。180 秒以上を経過しても別のルータからアップデートがルータに届かない場合、該当するルータから送られたルートは使用不能としてマークされます。240 秒後もまだ更新がない場合、ルータは更新のないルータのルーティング テーブル エントリをすべて削除します。

RIP では、各ルートの値を評価するためにホップ カウントが使用されます。ホップ カウントは、ルート内で経由されるルータ数です。直接接続されているネットワークのホップ カウントは 0 です。ホップ カウントが 16 のネットワークに到達できません。このように範囲（0 ～ 15）が狭いため、RIP は大規模ネットワークには適していません。

ルータにデフォルトのネットワーク パスが設定されている場合、RIP はルータを疑似ネットワーク 0.0.0.0 にリンクするルートをアドバタイズします。0.0.0.0 ネットワークは存在しません。RIP はデフォルトのルーティング機能を実行するためのネットワークとして、このネットワークを処理します。デフォルト ネットワークが RIP によって取得された場合、またはルータが最終ゲートウェイで、RIP がデフォルト メトリックによって設定されている場合、スイッ

チはデフォルトネットワークをアドバタイズします。RIPは指定されたネットワーク内のインターフェイスにアップデートを送信します。インターフェイスのネットワークを指定しなければ、RIP のアップデート中にアドバタイズされません。

RIP for IPv6

IPv6 の Routing Information Protocol（RIP）は、ルーティング メトリックとしてホップ カウントを使用するディスタンスベクトル プロトコルです。IPv6 アドレスおよびプレフィックスのサポート、すべての RIP ルータを含むマルチキャスト グループ アドレス FF02::9 を RIP アップデート メッセージの宛先アドレスとして使用する機能などがあります。

IPv6 の RIP の設定については、「*IPv6 の RIP の設定*」を参照してください。

IPv6 の RIP の詳細については、Cisco.com で『*Cisco IOS IPv6 Configuration Library*』の「Implementing RIP for IPv6」の章を参照してください。

サマリー アドレスおよびスプリット ホライズン

ブロードキャストタイプの IP ネットワークに接続され、ディスタンスベクトル ルーティング プロトコルを使用するルータでは、通常ルーティング ループの発生を抑えるために、スプリット ホライズン メカニズムが使用されます。スプリット ホライズンは、ルートに関する情報の発信元であるインターフェイス上の、ルータによって、その情報がアドバタイズされないようにします。この機能を使用すると、通常の場合は複数のルータ間通信が最適化されます（特にリンクが壊れている場合）。

RIP の設定方法

RIP のデフォルト設定

表 *13 : RIP* のデフォルト設定

機能	デフォルト設定
自動サマリー	イネーブル。
デフォルト情報送信元	ディセーブル。
デフォルト メトリック	自動メトリック変換（組み込み）
IP RIP 認証キーチェーン	認証なし 認証モード：クリア テキスト
IP RIP の起動	無効
IP スプリット ホライズン	メディアにより異なる

機能	デフォルト設定
Neighbor	未定義
ネットワーク	指定なし
オフセット リスト	ディセーブル。
出力遅延	0 ミリ秒
タイマー基準	• 更新：30 秒 • 無効：180 秒 • ホールドダウン：180 秒 • フラッシュ：240 秒
アップデート送信元の検証	イネーブル。
バージョン	RIP バージョン 1 およびバージョン 2 パケットを受信し、バージョン信します。

基本的な RIP パラメータの設定

RIP を設定するには、ネットワークに対して RIP ルーティングを有効にします。他のパラメータを設定することもできます。スイッチでは、ネットワーク番号を設定するまでRIP コンフィギュレーション コマンドは無視されます。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • 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： `Device# 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 routing** 例： `Device(config)# ip routing`	IP ルーティングを有効にします。（IP ルーティングが無効になっている場合だけ、必須です）。

	コマンドまたはアクション	目的
ステップ 4	**router rip** 例： `Device(config)# router rip`	RIP ルーティング プロセスを有効にし、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 5	**network** *network number* 例： `Device(config-router)# network 12.0.0.0`	ネットワークを RIP ルーティング プロセスと関連付けます。複数の **network** コマンドを指定できます。RIP ルーティング アップデートの送受信は、これらのネットワークのインターフェイスを経由する場合だけ可能です。 （注） RIP コマンドを有効にするには、ネットワーク番号を設定する必要があります。
ステップ 6	**neighbor** *ip-address* 例： `Device(config-router)# neighbor 10.2.5.1`	（任意）ルーティング情報を交換する隣接ルータを定義します。このステップを使用すると、RIP（通常はブロードキャストプロトコル）からのルーティングアップデートが非ブロードキャストネットワークに到達するようになります。
ステップ 7	**offset-list** [*access-list number* | *name*] {**in** | **out**} *offset* [*type number*] 例： `Device(config-router)# offset-list 103 in 10`	（任意）オフセットリストをルーティングメトリックに適用し、RIPによって取得したルートへの着信および発信メトリックを増加します。アクセス リストまたはインターフェイスを使用し、オフセットリストを制限できます。
ステップ 8	**timers basic** *update invalid holddown flush* 例： `Device(config-router)# timers basic 45 360 400 300`	（任意）ルーティング プロトコル タイマーを調整します。すべてのタイマーの有効範囲は 0 ～ 4294967295 秒です。 • *update*：ルーティング アップデートの送信間隔。デフォルトは 30 秒です。 • *invalid*：ルートが無効と宣言されるまでの時間。デフォルト値は 180 秒です。 • *holddown*：ルートがルーティング テーブルから削除されるまでの時間。デフォルト値は180秒です。 • *flush*：ルーティング アップデートが延期される時間。デフォルトは 240 秒です。
ステップ 9	**version** {**1** | **2**} 例：	（任意）RIP バージョン 1 または RIP バージョン 2 のパケットだけを送受信するようにスイッチを設定

	コマンドまたはアクション	目的
	`Device(config-router)# version 2`	します。デフォルトの場合、スイッチではバージョン 1 および 2 を受信しますが、バージョン 1 だけを送信します。インターフェイスコマンド **ip rip** {**send** | **receive**} **version 1** | **2** | **1 2**} を使用して、インターフェイスでの送受信に使用するバージョンを制御することもできます。
ステップ 10	**no auto summary** 例： `Device(config-router)# no auto summary`	（任意）自動要約を無効にします。デフォルトでは、クラスフル ネットワーク境界を通過するときにサブプレフィックスがサマライズされます。サマライズを無効にし（RIP バージョン 2 だけ）、クラスフル ネットワーク境界にサブネットおよびホスト ルーティング情報をアドバタイズします。
ステップ 11	**output-delay** *delay* 例： `Device(config-router)# output-delay 8`	（任意）送信する RIP アップデートにパケット間遅延を追加します。デフォルトでは、複数のパケットからなる RIP アップデートのパケットに、パケット間遅延が追加されません。パケットを低速なデバイスに送信する場合は、8 ～ 50 ミリ秒のパケット間遅延を追加できます。
ステップ 12	**end** 例： `Device(config-router)# end`	特権 EXEC モードに戻ります。
ステップ 13	**show ip protocols** 例： `Device# show ip protocols`	入力を確認します。
ステップ 14	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

RIP 認証の設定

RIP バージョン 1 は認証をサポートしていません。RIP バージョン 2 のパケットを送受信する場合は、インターフェイスで RIP 認証を有効にできます。インターフェイスで使用できる一連のキーは、キーチェーンによって指定されます。キーチェーンが設定されていないと、デフォルトの場合でも認証は実行されません。

RIP認証が有効であるインターフェイスでは、プレーンテキストとMD5という2つの認証モードがサポートされています。デフォルトはプレーン テキストです。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するインターフェイスを指定します。
ステップ 4	**ip rip authentication key-chain** *name-of-chain* 例： `Device(config-if)# ip rip authentication key-chain trees`	RIP 認証を有効にします。
ステップ 5	**ip rip authentication mode** {**text** | **md5**} 例： `Device(config-if)# ip rip authentication mode md5`	プレーン テキスト認証（デフォルト）または MD5 ダイジェスト認証を使用するように、インターフェイスを設定します。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show running-config** 例： `Device# show running-config`	入力を確認します。

	コマンドまたはアクション	目的
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

IPv6 RIP の設定

IPv6 の RIP ルーティングの設定の詳細については、Cisco.com で『*Cisco IOS IPv6 Configuration Library*』の「Implementing RIP for IPv6」の章を参照してください。

IPv6 の RIP ルーティングを設定するには、次の手順を実行します。

始める前に

IPv6 RIP を実行するようにスイッチを設定する前に、グローバル コンフィギュレーション モードで **ip routing** コマンドを使用してルーティングを有効にし、グローバル コンフィギュレーション モードで **ipv6 unicast-routing** コマンドを使用して IPv6 パケットの転送を有効にして、IPv6 RIP を有効にするレイヤ 3 インターフェイス上で IPv6 を有効にする必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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	**ipv6 router rip** *name* 例： `Device(config)# ipv6 router rip cisco`	IPv6 RIP ルーティング プロセスを設定し、このプロセスに対してルータ コンフィギュレーションモードを開始します。
ステップ 4	**maximum-paths** *number-paths* 例： `Device(config-router)# maximum-paths 6`	（任意）IPv6 RIP がサポートできる等コスト ルートの最大数を定義します。指定できる範囲は 1 ～ 32 で、デフォルトは 16 ルートです。
ステップ 5	**exit** 例： `Device(config-router)# exit`	グローバル コンフィギュレーション モードに戻ります。

	コマンドまたはアクション	目的
ステップ 6	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ3インターフェイスを指定します。
ステップ 7	**ipv6 rip** *name* **enable** 例： `Device(config-if)# ipv6 rip cisco enable`	指定された IPv6 RIP ルーティング プロセスをインターフェイス上で有効にします。
ステップ 8	**ipv6 rip** *name* **default-information** {**only** | **originate**} 例： `Device(config-if)# ipv6 rip cisco default-information only`	（任意）IPv6 デフォルト ルート（::/0）を RIP ルーティング プロセス アップデートに格納して、指定インターフェイスから送信します。 （注） 任意のインターフェイスから IPv6 デフォルト ルート（::/0）を送信したあとに、ルーティング ループが発生しないようにするために、ルーティング プロセスは任意のインターフェイスで受信したすべてのデフォルト ルートを無視します。 • **only**：このインターフェイスから送信するアップデートに、デフォルト ルートを格納し、その他のすべてのルートを含めない場合に選択します。 • **originate**：このインターフェイスから送信するアップデートに、デフォルト ルートおよびその他のすべてのルートを格納する場合に選択します。
ステップ 9	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

ステップ 10	次のいずれかを使用します。
	• **show ipv6 rip** [*name*] [**interface** *interface-id*] [**database**] [**next-hops**]
	• **show ipv6 rip**
	例：
	`Device# show ipv6 rip cisco interface gigabitethernet 2/0/1`
	または
	`Device# show ipv6 rip`
		• 現在の IPv6 RIP プロセスに関する情報を表示します。
		• IPv6 ルーティング テーブルの現在の内容を表示します。

	コマンドまたはアクション	目的
ステップ 11	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

サマリー アドレスおよびスプリット ホライズンの設定

（注） ルートを適切にアドバタイズするため、アプリケーションがスプリット ホライズンを無効にする必要がある場合を除き、通常はこの機能を無効にしないでください。

ダイヤルアップクライアント用のネットワークアクセスサーバーで、サマライズされたローカルIPアドレスプールをアドバタイズするように、RIPが動作しているインターフェイスを設定する場合は、**ip summary-address rip** インターフェイス コンフィギュレーション コマンドを使用します。

（注） スプリット ホライズンが有効の場合、自動サマリーとインターフェイス IP サマリー アドレスはともにアドバタイズされません。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 4	**ip address** *ip-address subnet-mask* 例：	IP アドレスおよび IP サブネットを設定します。

	コマンドまたはアクション	目的
	`Device(config-if)# ip address 10.1.1.10 255.255.255.0`	
ステップ 5	**ip summary-address rip ip address** *ip-network mask* 例： `Device(config-if)# ip summary-address rip ip address 10.1.1.30 255.255.255.0`	サマライズする IP アドレスおよび IP ネットワーク マスクを設定します。
ステップ 6	**no ip split horizon** 例： `Device(config-if)# no ip split horizon`	インターフェイスでスプリット ホライズンを無効にします。
ステップ 7	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 8	**show ip interface** *interface-id* 例： `Device# show ip interface gigabitethernet 1/0/1`	入力を確認します。
ステップ 9	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

スプリット ホライズンの設定

ブロードキャストタイプの IP ネットワークに接続され、ディスタンスベクトル ルーティング プロトコルを使用するルータでは、通常ルーティング ループの発生を抑えるために、スプリット ホライズン メカニズムが使用されます。スプリット ホライズンは、ルートに関する情報の発信元であるインターフェイス上の、ルータによって、その情報がアドバタイズされないようにします。この機能を使用すると、複数のルータ間通信が最適化されます（特にリンクが壊れている場合）。

（注） ルートを適切にアドバタイズするために、アプリケーションがスプリット ホライズンを無効にする必要がある場合を除き、通常この機能を無効にしないでください。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するインターフェイスを指定します。
ステップ 4	**ip address** *ip-address subnet-mask* 例： `Device(config-if)# ip address 10.1.1.10 255.255.255.0`	IP アドレスおよび IP サブネットを設定します。
ステップ 5	**no ip split-horizon** 例： `Device(config-if)# no ip split-horizon`	インターフェイスでスプリット ホライズンを無効にします。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show ip interface** *interface-id* 例： `Device# show ip interface gigabitethernet 1/0/1`	入力を確認します。
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

例：IPv6 用の RIP の設定

次に、最大 8 の等コスト ルートにより RIP ルーティング プロセス *cisco* を有効にし、インターフェイス上でこれを有効にする例を示します。

```
Device> enable
Devce# configure terminal
Device(config)# ipv6 router rip cisco
Device(config-router)# maximum-paths 8
Device(config)# exit
Device(config)# interface gigabitethernet2/0/11
Device(config-if)# ipv6 rip cisco enable
```

サマリー アドレスおよびスプリット ホライズンの設定例

次の例では、主要ネットは 10.0.0.0 です。自動サマリー アドレス 10.0.0.0 はサマリー アドレス 10.2.0.0 によって上書きされるため、10.2.0.0 はインターフェイス ギガビット イーサネット ポート 2 からアドバタイズされますが、10.0.0.0 はアドバタイズされません。この例では、インターフェイスがレイヤ 2 モード（デフォルト）の場合は、**no switchport** インターフェイス コンフィギュレーション コマンドを入力してから、**ip address** インターフェイス コンフィギュレーション コマンドを入力する必要があります。

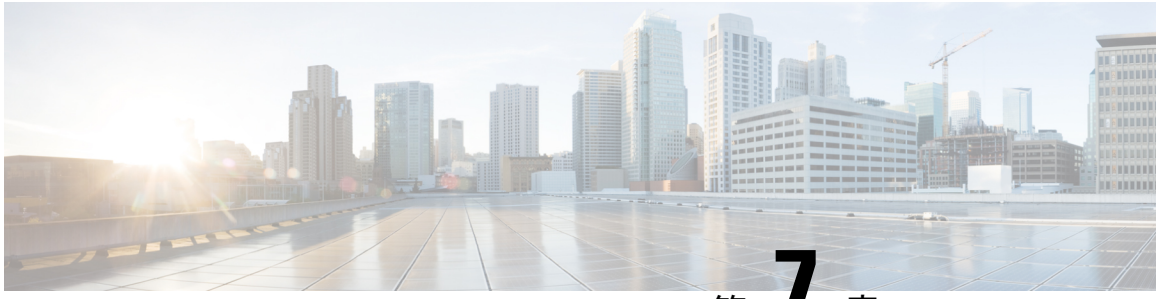
（注） スプリット ホライズンが有効である場合、（**ip summary-address rip** ルータ コンフィギュレーション コマンドによって設定される）自動サマリーとインターフェイス サマリー アドレスはともにアドバタイズされません。

```
Device(config)# router rip
Device(config-router)# interface gigabitethernet1/0/2
Device(config-if)# ip address 10.1.5.1 255.255.255.0
Device(config-if)# ip summary-address rip 10.2.0.0 255.255.0.0
Device(config-if)# no ip split-horizon
Device(config-if)# exit
Device(config)# router rip
Device(config-router)# network 10.0.0.0
Device(config-router)# neighbor 2.2.2.2 peer-group mygroup
Device(config-router)# end
```

Routing Information Protocol に関する機能情報

表 *14 : Routing Information Protocol* に関する機能情報

リリース	機能情報
Cisco IOS XE Everest 16.5.1a	この機能が導入されました。



第 7 章

OSPF の設定

OSPF に関する情報

OSPF は IP ネットワーク専用の IGP で、IP サブネット化、および外部から取得したルーティング情報のタグ付けをサポートしています。OSPF を使用するとパケット認証も可能になり、パケットを送受信するときに IP マルチキャストが使用されます。シスコの実装では、RFC1253 の OSPF 管理情報ベース（MIB）がサポートされています。

シスコの実装は、次の主要機能を含む OSPF バージョン 2 仕様に準拠します。

- スタブ エリアの定義がサポートされています。
- 任意の IP ルーティング プロトコルによって取得されたルートは、別の IP ルーティング プロトコルに再配信されます。つまり、ドメイン内レベルで、OSPF は EIGRP および RIP によって取得したルートを取り込むことができます。OSPF ルートを RIP に伝達することもできます。
- エリア内の隣接ルータ間でのプレーン テキスト認証および MD5 認証がサポートされています。
- 設定可能なルーティング インターフェイス パラメータには、インターフェイス出力コスト、再送信インターバル、インターフェイス送信遅延、ルータ プライオリティ、ルータのデッド インターバルと hello インターバル、認証キーなどがあります。
- 仮想リンクがサポートされています。
- RFC 1587 に基づく Not-So-Stubby-Area（NSSA）がサポートされています。

通常、OSPF を使用するには、多くの内部ルータ、複数のエリアに接続された Area Border Router（ABR;エリア境界ルータ）、および自律システム境界ルータ（ASBR）間で調整する必要があります。最小設定では、すべてのデフォルトパラメータ値、エリアに割り当てられたインターフェイスが使用され、認証は行われません。環境をカスタマイズする場合は、すべてのルータの設定を調整する必要があります。

OSPF for IPv6

スイッチは、IP のリンクステート プロトコルの 1 つである、IPv6 の Open Shortest Path First（OSPF）をサポートしています。

IPv6 用の OSPF の設定については、「*IPv6* 用の *OSPF* の設定」を参照してください。

詳細については、Cisco.com の『*Cisco IOS IPv6 Configuration Library*』を参照してください。

OSPF NSF

スイッチまたはスイッチ スタックは、次の 2 つのレベルの NSF をサポートします。

• OSPF NSF 認識 （134 ページ）
• OSPF NSF 対応 （134 ページ）

OSPF NSF 認識

Network Advantageライセンスは IPv4 の OSPF NSF 認識をサポートしています。隣接ルータが NSF 対応である場合、レイヤ 3 デバイスでは、ルータに障害（クラッシュ）が発生してプライマリルートプロセッサ（RP）がバックアップ RP によって引き継がれる間、または処理を中断させずにソフトウェアアップグレードを行うためにプライマリ RP を手動でリロードしている間、隣接ルータからパケットを転送し続けます。

この機能をディセーブルにできません。

OSPF NSF 対応

Network Advantage ライセンスでは、前のリリースでサポートされていた OSPFv2 NSF Cisco フォーマットに加えて、OSPFv2 NSF IETF フォーマットもサポートされます。この機能の詳細については、『*NSF—OSPF (RFC 3623 OSPF Graceful Restart)*』を参照してください。

Network Advantage ライセンスは、OSPF NSF 対応ルーティングも IPv4 に対してサポートし、スタックのアクティブスイッチ変更後のコンバージェンス向上と、トラフィック損失低減を実現します。

OSPF NSF 対応スタックでアクティブスイッチの変更が生じた場合、新しいアクティブスイッチは自身のリンクステートデータベースを OSPF ネイバーと再同期化するために、次の 2 つの処理をする必要があります。

- ネイバー関係をリセットせずにネットワーク上の使用可能な OSPF ネイバーを解放します。

- ネットワークのリンクステート データベースの内容を再取得します。

アクティブスイッチの変更後、新しいアクティブスイッチはネイバー NSF 認識デバイスに OSPF NSF 信号を送信します。デバイスはこの信号を、スタックとのネイバー関係をリセットしない指示として認識します。NSF 対応アクティブスイッチは、ネットワーク上の他のルータから信号を受け取ると、自身のネイバーリストの再構築を開始します。

NSF 対応アクティブスイッチはネイバー関係を再確立すると、自身のデータベースを NSF 認識ネイバーと再同期化し、OSPF ネイバー間でルーティング情報を交換します。新しいアクティブスイッチはこのルーティング情報を使用して、新しい情報を基に古いルートの削除、ルーティング情報ベース（RIB）の更新、転送情報ベース（FIB）の更新を行います。これで OSPF プロトコルは完全に収束します。

（注） OSPF NSF では、すべてのネイバー ネットワーク デバイスが NSF 認識である必要があります。ネットワーク セグメント上に非 NSF 認識ネイバーが検出された場合、NSF 対応ルータはそのセグメントに対する NSF 機能をディセーブルにします。すべてのデバイスが NSF 認識または NSF 対応デバイスとなっているその他のネットワーク セグメントでは、NSF 対応機能が継続して提供されます。

OSPF NSF ルーティングを有効にするには、**nsf** OSPF ルーティング コンフィギュレーション コマンドを使用します。OSPF NSF ルーティングが有効になっていることを確認するには、**show ip ospf** 特権 EXEC コマンドを使用します。

OSPF エリア パラメータ

複数の OSPF エリア パラメータを設定することもできます。設定できるパラメータには、エリア、スタブ エリア、および NSSA への無許可アクセスをパスワードによって阻止する認証用パラメータがあります。スタブエリアは、外部ルートの情報が送信されないエリアです。が、代わりに、自律システム（AS）外の宛先に対するデフォルトの外部ルートが、ABR によって生成されます。NSSA ではコアからそのエリアへ向かう LSA の一部がフラッディングされませんが、再配信することによって、エリア内の AS 外部ルートをインポートできます。

経路集約は、アドバタイズされたアドレスを、他のエリアでアドバタイズされる単一のサマリー ルートに統合することです。ネットワーク番号が連続する場合は、**area range** ルータ コンフィギュレーション コマンドを使用し、範囲内のすべてのネットワークを対象とするサマリールートをアドバタイズするように ABR を設定できます。

その他の OSPF パラメータ

ルータ コンフィギュレーション モードで、その他の OSPF パラメータを設定することもできます。

- ルート集約：他のプロトコルからルートを再配信すると、各ルートは外部 LSA 内で個別にアドバタイズされます。OSPF リンクステートデータベースのサイズを小さくするには、**summary-address** ルータ コンフィギュレーション コマンドを使用し、指定されたネット

ワークアドレスおよびマスクに含まれる、再配信されたすべてのルートを単一のルータにアドバタイズします。

- 仮想リンク：OSPF では、すべてのエリアがバックボーン エリアに接続されている必要があります。バックボーンが不連続である場合に仮想リンクを確立するには、2 つの ABR を仮想リンクのエンドポイントとして設定します。設定情報には、他の仮想エンドポイント（他の ABR）の ID、および 2 つのルータに共通する非バックボーン リンク（通過エリア）などがあります。仮想リンクをスタブ エリアから設定できません。
- デフォルト ルート：OSPF ルーティング ドメイン内へのルート再配信を設定すると、ルータは自動的に自律システム境界ルータ（ASBR）になります。ASBR を設定し、強制的に OSPF ルーティング ドメインにデフォルト ルートを生成できます。
- すべての OSPF **show** 特権 EXEC コマンドでの表示にドメインネームサーバー（DNS）名を使用すると、ルータ ID やネイバー ID を指定して表示する場合に比べ、ルータを簡単に特定できます。
- デフォルト メトリック：OSPF は、インターフェイスの帯域幅に従ってインターフェイスの OSPF メトリックを計算します。メトリックは、帯域幅で分割された *ref-bw* として計算されます。ここでの *ref* のデフォルト値は 10 で、帯域幅（*bw*）は **bandwidth** インターフェイス コンフィギュレーション コマンドによって指定されます。大きな帯域幅を持つ複数のリンクの場合は、大きな数値を指定し、これらのリンクのコストを区別できます。
- アドミニストレーティブ ディスタンスは、ルーティング情報送信元の信頼性を表す数値です。0 ～ 255 の整数を指定でき、値が大きいほど信頼性は低下します。アドミニストレーティブ ディスタンスが 255 の場合はルーティング情報の送信元をまったく信頼できないため、無視する必要があります。OSPF では、エリア内のルート（エリア内）、別のエリアへのルート（エリア間）、および再配信によって学習した別のルーティング ドメインからのルート（外部）の 3 つの異なるアドミニストレーティブ ディスタンスが使用されます。どのアドミニストレーティブ ディスタンスの値でも変更できます。
- 受動インターフェイス：イーサネット上の 2 つのデバイス間のインターフェイスは 1 つのネットワーク セグメントしか表しません。このため、OSPF が送信側インターフェイスに hello パケットを送信しないようにするには、送信側デバイスを受動インターフェイスに設定する必要があります。両方のデバイスは受信側インターフェイス宛ての hello パケットを使用することで、相互の識別を可能にします。
- ルート計算タイマー：OSPF がトポロジ変更を受信してから SPF 計算を開始するまでの遅延時間、および 2 つの SPF 計算の間のホールド タイムを設定できます。
- ネイバー変更ログ：OSPF ネイバー ステートが変更されたときに Syslog メッセージを送信するようにルータを設定し、ルータの変更を詳細に表示できます。

LSA グループ ペーシング

OSPF LSA グループ ペーシング機能を使用すると、OSPF LSA をグループ化し、リフレッシュ、チェックサム、エージング機能の同期を取って、ルータをより効率的に使用できるようになります。デフォルトでこの機能はイネーブルとなっています。デフォルトのペーシング インター

バルは 4 分間です。通常は、このパラメータを変更する必要はありません。最適なグループペーシング インターバルは、ルータがリフレッシュ、チェックサム、エージングを行う LSA 数に反比例します。たとえば、データベース内に約 10000 個の LSA が格納されている場合は、ペーシング インターバルを短くすると便利です。小さなデータベース（40 ～ 100 LSA）を使用する場合は、ペーシング インターバルを長くし、10 ～ 20 分に設定してください。

ループバック インターフェイス

OSPF は、インターフェイスに設定されている最大の IP アドレスをルータ ID として使用します。このインターフェイスがダウンした場合、または削除された場合、OSPF プロセスは新しいルータ ID を再計算し、すべてのルーティング情報をそのルータのインターフェイスから再送信します。ループバック インターフェイスが IP アドレスによって設定されている場合、他のインターフェイスにより大きな IP アドレスがある場合でも、OSPF はこの IP アドレスをルータ ID として使用します。ループバック インターフェイスに障害は発生しないため、安定性は増大します。OSPF は他のインターフェイスよりもループバック インターフェイスを自動的に優先し、すべてのループバック インターフェイスの中で最大の IP アドレスを選択します。

OSPF の設定方法

OSPF のデフォルト設定

表 *15 : OSPF* のデフォルト設定

機能	デフォルト設定
インターフェイス パラメータ	コスト： 再送信インターバル：5 秒 送信遅延：1 秒 プライオリティ：1 hello インターバル：10 秒 デッド インターバル：hello インターバルの 4 倍 認証なし パスワードの指定なし MD5 認証はディセーブル

機能	デフォルト設定
エリア	認証タイプ：0（認証なし） デフォルト コスト：1 範囲：ディセーブル スタブ：スタブ エリアは未定義 NSSA：NSSA エリアは未定義
自動コスト	100 Mb/s
デフォルト情報送信元	ディセーブル。イネーブルの場合、デフォルトのメトリック設定は 1 ルート タイプのデフォルトはタイプ 2 です。
デフォルト メトリック	各ルーティング プロトコルに適切な、組み込みの自動メトリック変換
距離 OSPF	dist1（エリア内のすべてのルート）：110。dist2（エリア間のすべての 110。 および dist3（他のルーティング ドメインからのルート）：110
OSPF データベース フィルタ	ディセーブル。すべての発信 LSA がインターフェイスにフラッディン す。
IP OSPF 名検索	ディセーブル。
隣接関係変更ログ	イネーブル。
ネイバー	指定なし
ネイバー データベース フィルタ	ディセーブル。すべての発信 LSA はネイバーにフラッディングされ
ネットワーク エリア	ディセーブル。
ノンストップ フォワーディング（NSF）認識	イネーブル。レイヤ 3 スイッチでは、ハードウェアやソフトウェアの 隣接する NSF 対応ルータからのパケットを転送し続けることができ
ルータ ID	OSPF ルーティング プロセスは未定義
サマリー アドレス	ディセーブル。
タイマー LSA グループのペーシング	240 秒
タイマー Shortest Path First（SPF）	spf 遅延：50 ミリ秒、spf ホールド時間：200 ミリ秒

機能	デフォルト設定
仮想リンク	エリア ID またはルータ ID は未定義 hello インターバル：10 秒 再送信インターバル：5 秒 送信遅延：1 秒 デッド インターバル：40 秒 認証キー：キーは未定義 メッセージ ダイジェスト キー（MD5）：キーは未定義

基本的な OSPF パラメータの設定

OSPF をイネーブルにするには、OSPF ルーティング プロセスを作成し、そのルーティング プロセスに関連付けられる IP アドレスの範囲を指定し、その範囲に関連付けられるエリア ID を割り当てます。Network Essentials イメージを実行するスイッチの場合は、Cisco OSPFv2 NSF 形式または IETF OSPFv2 NSF 形式のいずれかを設定できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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 ospf** process-id 例： `Device(config)#router ospf 15`	OSPF ルーティングをイネーブルにし、ルータ コンフィギュレーション モードを開始します。プロセス ID はローカルに割り当てられ、内部で使用される識別パラメータで、任意の正の整数を指定できます。各 OSPF ルーティング プロセスには一意の値があります。 （注） OSPF for Routed Access は、OSPFv2 インスタンスと OSPFv3 インスタンスをそれぞれ 1 つずつと、最大 1000 のダイナミックに学習されるルートをサポートします。

	コマンドまたはアクション	目的
ステップ 4	**nsf cisco** [**enforce global**] 例： `Device(config)#nsf cisco enforce global`	（任意）OSPF での Cisco NSF 動作をイネーブルにします。**enforce global** キーワードを指定すると、非 NSF 認識のネイバー ネットワーキング デバイスが検出されたときに NSF 再起動がキャンセルされます。 （注） ステップ 3 またはステップ 4 でコマンドを入力し、ステップ 5 に進みます。
ステップ 5	**nsf ietf** [**restart-interval** *seconds*] 例： `Device(config)#nsf ietf restart-interval 60`	（任意）OSPF での IETF NSF 動作をイネーブルにします。**restart-interval** キーワードでは、グレースフルリスタート間隔の長さを秒単位で指定します。有効な範囲は 1 ～ 1800 です。デフォルトは 120 です。 （注） ステップ 3 またはステップ 4 でコマンドを入力し、ステップ 5 に進みます。
ステップ 6	**network** address wildcard-mask **area** area-id 例： `Device(config)#network 10.1.1.1 255.240.0.0 area 20`	OSPF が動作するインターフェイス、およびそのインターフェイスのエリア ID を定義します。単一のコマンドにワイルドカードマスクを指定し、特定の OSPF エリアに関連付けるインターフェイスを 1 つまたは複数定義できます。エリア ID には 10 進数または IP アドレスを指定できます。
ステップ 7	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 8	**show ip protocols** 例： `Device#show ip protocols`	入力を確認します。
ステップ 9	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

IPv6 OSPF の設定

IPv6 の OSPF ルーティングの設定の詳細については、Cisco.com で『*Cisco IOS IPv6 Configuration Library*』の「Implementing OSPF for IPv6」の章を参照してください。

IPv6 の OSPF ルーティングを設定するには、次の手順を実行します。

始める前に

ネットワークでは、IPv6 の OSPF をカスタマイズできます。ただし、IPv6 の OSPF のデフォルト設定は、ほとんどのお客様および機能の要件を満たします。

次の注意事項に従ってください。

- IPv6 コマンドのデフォルト設定を変更する場合は注意してください。デフォルト設定を変更すると、IPv6 ネットワークの OSPF に悪影響が及ぶことがあります。
- インターフェイスで IPv6 OSPF を有効にする前に、グローバル コンフィギュレーション モードで **ip routing** コマンドを使用してルーティングを有効にし、グローバル コンフィギュレーション モードで **ipv6 unicast-routing** コマンドを使用して IPv6 パケットの転送を有効にし、IPv6 OSPF を有効にするレイヤ 3 インターフェイスで IPv6 を有効にする必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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	**ipv6 router ospf** *process-id* 例： `Device(config)# ipv6 router ospf 21`	プロセスに対して OSPF ルータ コンフィギュレーション モードを有効にします。プロセス ID は、IPv6 OSPF ルーティング プロセスを有効にする場合に管理上割り当てられる番号です。この ID はローカルに割り当てられ、1 ～ 65535 の正の整数を指定できます。
ステップ 4	**area** *area-id* **range** {*ipv6-prefix/prefix length*} [**advertise** | **not-advertise**] [**cost** *cost*] 例： `Device(config)# area .3 range 2001:0DB8::/32 not-advertise`	（任意）エリア境界でルートを統合および集約します。 • *area-id*：ルートをサマライズするエリアの ID。10 進数または IPv6 プレフィックスのどちらかを指定できます。 • *ipv6-prefix/prefix length*：宛先 IPv6 ネットワーク、およびプレフィックス（アドレスのネットワーク部分）を構成するアドレスの上位連続ビット数を示す 10 進数。10 進値の前にスラッシュ（/）を付加する必要があります。

	コマンドまたはアクション	目的
		• **advertise**：（任意）アドバタイズするアドレス範囲ステータスを設定し、タイプ3のサマリー リンクステート アドバタイズメント（LSA）を生成します。 • **not-advertise**：（任意）アドレス範囲ステータスを DoNotAdvertise に設定します。Type3 サマリー LSA は抑制され、コンポーネント ネットワークは他のネットワークから隠された状態のままです。 • **cost** *cost*：（任意）現在のサマリー ルートのメトリックまたはコストを設定します。宛先への最短パスを判別する場合に、OSPF SPF 計算で使用します。指定できる値は 0 ～ 16777215 です。
ステップ 5	**maximum paths** *number-paths* 例： `Device(config)# maximum paths 16`	（任意）IPv6 OSPF がルーティング テーブルに入力する必要がある、同じ宛先への等コスト ルートの最大数を定義します。指定できる範囲は 1 ～ 32 で、デフォルトは 16 です。
ステップ 6	**exit** 例： `Device(config-if)# exit`	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 7	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ3インターフェイスを指定します。
ステップ 8	**ipv6 ospf** *process-id* **area** *area-id* [**instance** *instance-id*] 例： `Device(config-if)# ipv6 ospf 21 area .3`	インターフェイスで IPv6 の OSPF を有効にします。 • **instance** *instance-id*：（任意）インスタンス ID
ステップ 9	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。
ステップ 10	次のいずれかを使用します。 • **show ipv6 ospf** [*process-id*] [*area-id*] **interface** [*interface-id*] • **show ipv6 ospf** [*process-id*] [*area-id*] 例：	• OSPF インターフェイスに関する情報を表示します。 • OSPF ルーティング プロセスに関する一般情報を表示します。

	コマンドまたはアクション	目的

	`Device# show ipv6 ospf 21 interface gigabitethernet2/0/1`
	または
	`Device# show ipv6 ospf 21`	

ステップ 11	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

OSPF インターフェイスの設定

ip ospf インターフェイス コンフィギュレーション コマンドを使用すると、インターフェイス固有の OSPF パラメータを変更できます。これらのパラメータを変更する必要はありませんが、一部のインターフェイス パラメータ（hello インターバル、デッド インターバル、認証キーなど）については、接続されたネットワーク内のすべてのルータで統一性を維持する必要があります。これらのパラメータを変更した場合は、ネットワーク内のすべてのルータの値も同様に変更してください。

（注） **ip ospf** インターフェイス コンフィギュレーション コマンドはすべてオプションです。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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	**interface** interface-id 例： `Device(config)#interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 4	**ip ospf** cost 例：	（任意）インターフェイスでパケットを送信するコストを明示的に指定します。

	コマンドまたはアクション	目的
	`Device(config-if)#ip ospf 8`	
ステップ 5	**ip ospf retransmit-interval** seconds 例： `Device(config-if)#ip ospf transmit-interval 10`	（任意）LSA 送信間隔を秒数で指定します。指定できる範囲は 1 ～ 65535 秒です。デフォルト値は 5 秒です。
ステップ 6	**ip ospf transmit-delay** seconds 例： `Device(config-if)#ip ospf transmit-delay 2`	（任意）リンク ステート アップデート パケットを送信するまでの予測待機時間を秒数で設定します。指定できる範囲は 1 ～ 65535 秒です。デフォルト値は 1 秒です。
ステップ 7	**ip ospf priority** number 例： `Device(config-if)#ip ospf priority 5`	（任意）ネットワークに対して、OSPF で指定されたルータを検索するときに役立つプライオリティを設定します。有効な範囲は 0 ～ 255 です。デフォルトは 1 です。
ステップ 8	**ip ospf hello-interval** seconds 例： `Device(config-if)#ip ospf hello-interval 12`	（任意）OSPF インターフェイスで hello パケットの送信間隔を秒数で設定します。ネットワークのすべてのノードで同じ値を指定する必要があります。指定できる範囲は 1 ～ 65535 秒です。デフォルトは 10 秒です。
ステップ 9	**ip ospf dead-interval** seconds 例： `Device(config-if)#ip ospf dead-interval 8`	（任意）最後のデバイスで hello パケットが確認されてから、OSPF ルータがダウンしていることがネイバーによって宣言されるまでの時間を秒数で設定します。ネットワークのすべてのノードで同じ値を指定する必要があります。指定できる範囲は 1 ～ 65535 秒です。デフォルト値は hello インターバルの 4 倍です。
ステップ 10	**ip ospf authentication-key** key 例： `Device(config-if)#ip ospf authentication-key password`	（任意）隣接 OSPF ルータで使用されるパスワードを割り当てます。パスワードには、キーボードから入力した任意の文字列（最大 8 バイト長）を指定できます。同じネットワーク上のすべての隣接ルータには、OSPF 情報を交換するため、同じパスワードを設定する必要があります。
ステップ 11	**ip ospf message digest-key** keyid **md5** key 例： `Device(config-if)#ip ospf message digest-key 16 md5 your1pass`	（任意）MDS 認証をイネーブルにします。 • *keyid*：1 ～ 255 の ID。 • *key*：最大 16 バイトの英数字パスワード
ステップ 12	**ip ospf database-filter all out** 例：	（任意）インターフェイスへの OSPF LSA パケットのフラッディングを阻止します。デフォルトで

	コマンドまたはアクション	目的
	`Device(config-if)#ip ospf database-filter all out`	は、OSPFは、LSAが到着したインターフェイスを除き、同じエリア内のすべてのインターフェイスで新しい LSA をフラッドします。
ステップ 13	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 14	**show ip ospf interface** [*interface-name*] 例： `Device#show ip ospf interface`	OSPFに関連するインターフェイス情報を表示します。
ステップ 15	**show ip ospf neighbor detail** 例： `Device#show ip ospf neighbor detail`	ネイバー スイッチの NSF 認証ステータスを表示します。出力には、次のいずれかが表示されます。 • *Options is 0x52* *LLS Options is 0x1 (LR)* これらの行の両方が表示される場合、ネイバー スイッチが NSF 認識です。 • *Options is 0x42*：ネイバー スイッチが NSF 認識でないことを示します。
ステップ 16	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

OSPF エリア パラメータの設定

始める前に

（注） OSPF **area** ルータ コンフィギュレーション コマンドはすべて任意です。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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 ospf** process-id 例： `Device(config)#router ospf 109`	OSPF ルーティングをイネーブルにし、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	**area** *area-id* **authentication** 例： `Device(config-router)#area 1 authentication`	（任意）特定のエリアへの無許可アクセスに対して、パスワードベースの保護を可能にします。ID には 10 進数または IP アドレスのいずれかを指定できます。
ステップ 5	**area** *area-id* **authentication message-digest** 例： `Device(config-router)#area 1 authentication message-digest`	（任意）エリアに関して MD5 認証を有効にします。
ステップ 6	**area** *area-id* **stub** [**no-summary**] 例： `Device(config-router)#area 1 stub`	（任意）エリアをスタブエリアとして定義します。**no-summary** キーワードを指定すると、ABR はサマリー リンク アドバタイズメントをスタブエリアに送信できなくなります。
ステップ 7	**area** *area-id* **nssa** [**no-redistribution**] [**default-information-originate**] [**no-summary**] 例： `Device(config-router)#area 1 nssa default-information-originate`	（任意）エリアを NSSA として定義します。同じエリア内のすべてのルータは、エリアが NSSA であることを認識する必要があります。次のキーワードのいずれかを選択します。 • **no-redistribution**：ルータが NSSA ABR の場合、**redistribute** コマンドを使用して、ルートを NSSA エリアでなく通常のエリアに取り込む場合に使用します。 • **default-information-originate**：LSA タイプ 7 を NSSA に取り込めるようにする場合に、ABR で選択します。

	コマンドまたはアクション	目的
		• **no-redistribution**：サマリー LSA を NSSA に送信しない場合に選択します。
ステップ 8	**area** *area-id* **range** *address mask* 例： `Device(config-router)#area 1 range 255.240.0.0`	（任意）単一のルートをアドバタイズするアドレス範囲を指定します。このコマンドは、ABR に対してだけ使用します。
ステップ 9	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 10	**show ip ospf** [*process-id*] 例： `Device#show ip ospf`	設定を確認するため、一般的な OSPF ルーティング プロセスまたは特定のプロセス ID に関する情報を表示します。
ステップ 11	**show ip ospf** [*process-id* [*area-id*]] **database** 例： `Device#show ip osfp database`	特定のルータの OSPF データベースに関連する情報のリストを表示します。
ステップ 12	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

その他の OSPF パラメータの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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。

	コマンドまたはアクション	目的
ステップ 3	**router ospf** *process-id* 例： `Device(config)#router ospf 10`	OSPF ルーティングをイネーブルにし、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	**summary-address** *address mask* 例： `Device(config)#summary-address 10.1.1.1 255.255.255.0`	（任意）1 つのサマリー ルートだけがアドバタイズされるように、再配信されたルートのアドレスおよび IP サブネット マスクを指定します。
ステップ 5	**area** *area-id* **virtual-link** *router-id* [**hello-interval** *seconds*] [**retransmit-interval** *seconds*] [**trans**] [[**authentication-key** *key*] | **message-digest-key** *keyid* **md5** *key*]] 例： `Device(config)#area 2 virtual-link 192.168.255.1 hello-interval 5`	（任意）仮想リンクを確立し、パラメータを設定します。
ステップ 6	**default-information originate** [**always**] [**metric** *metric-value*] [**metric-type** *type-value*] [**route-map** *map-name*] 例： `Device(config)#default-information originate metric 100 metric-type 1`	（任意）強制的に OSPF ルーティング ドメインにデフォルト ルートを生成するように ASBR を設定します。パラメータはすべて任意です。
ステップ 7	**ip ospf name-lookup** 例： `Device(config)#ip ospf name-lookup`	（任意）DNS 名検索を設定します。デフォルトでは無効になっています。
ステップ 8	**ip auto-cost reference-bandwidth** *ref-bw* 例： `Device(config)#ip auto-cost reference-bandwidth 5`	（任意）単一のルートをアドバタイズするアドレス範囲を指定します。このコマンドは、ABR に対してだけ使用します。
ステップ 9	**distance ospf** {[**inter-area** *dist1*] [**inter-area** *dist2*] [**external** *dist3*]} 例： `Device(config)#distance ospf inter-area 150`	（任意）OSPF の距離の値を変更します。各タイプのルートのデフォルト距離は 110 です。有効値は 1 ～ 255 です。
ステップ 10	**passive-interface** *type number* 例：	（任意）指定されたインターフェイス経由の hello パケットの送信を抑制します。

	コマンドまたはアクション	目的
	`Device(config)#passive-interface gigabitethernet 1/0/6`	
ステップ 11	**timers throttle spf** *spf-delay spf-holdtime spf-wait* 例： `Device(config)#timers throttle spf 200 100 100`	（任意）ルート計算タイマーを設定します。 • *spf-delay*：SPF 計算の変更を受信する間の遅延。指定できる範囲は1～600000ミリ秒です。 • *spf-holdtime*：最初と 2 番目の SPF 計算の間の遅延。指定できる範囲は 1 ～ 600000 ミリ秒です。 • *spf-wait*：SPF計算の最大待機時間（ミリ秒）。指定できる範囲は 1 ～ 600000 ミリ秒です。
ステップ 12	**ospf log-adj-changes** 例： `Device(config)#ospf log-adj-changes`	（任意）ネイバー ステートが変更されたとき、syslog メッセージを送信します。
ステップ 13	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 14	**show ip ospf** [*process-id* [*area-id*]] **database** 例： `Device#show ip ospf database`	特定のルータのOSPFデータベースに関連する情報のリストを表示します。
ステップ 15	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

LSA グループ ペーシングの変更

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例：	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
	`Device>enable`	
ステップ 2	**configure terminal** 例： `Device#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 ospf** *process-id* 例： `Device(config)#router ospf 25`	OSPF ルーティングをイネーブルにし、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	**timers lsa-group-pacing** *seconds* 例： `Device(config-router)#timers lsa-group-pacing 15`	LSA の グループ ペーシングを変更します。
ステップ 5	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 6	**show running-config** 例： `Device#show running-config`	入力を確認します。
ステップ 7	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ループバック インターフェイスの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例：	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
	`Device>enable`	
ステップ 2	**configure terminal** 例： `Device#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	**interface loopback 0** 例： `Device(config)#interface loopback 0`	ループバック インターフェイスを作成し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**ip address** address mask 例： `Device(config-if)#ip address 10.1.1.5 255.255.240.0`	このインターフェイスに IP アドレスを割り当てます。
ステップ 5	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 6	**show ip interface** 例： `Device#show ip interface`	入力を確認します。
ステップ 7	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

OSPF のモニタリング

IP ルーティング テーブルの内容、キャッシュの内容、およびデータベースの内容など、特定の統計情報を表示できます。

表 *16 : IP OSPF* 統計情報の表示コマンド

コマンド	目的
show ip ospf [*process-id*]	OSPF ルーティング 示します。
show ip ospf [*process-id*] **database** [**router**] [*link-state-id*] **show ip ospf** [*process-id*] **database** [**router**] [**self-originate**] **show ip ospf** [*process-id*] **database** [**router**] [**adv-router** [*ip-address*]] **show ip ospf** [*process-id*] **database** [**network**] [*link-state-id*] **show ip ospf** [*process-id*] **database** [**summary**] [*link-state-id*] **show ip ospf** [*process-id*] **database** [**asbr-summary**] [*link-state-id*] **show ip ospf** [*process-id*] **database** [**external**] [*link-state-id*] **show ip ospf** [*process-id area-id*] **database** [**database-summary**]	OSPF データベース します。
show ip ospf border-routes	内部の OSPF ルー ブル エントリを表
show ip ospf interface [*interface-name*]	OSPF に関連するイ す。
show ip ospf neighbor [*interface-name*] [*neighbor-id*] **detail**	OSPF インターフェ す。
show ip ospf virtual-links	OSPF に関連する仮

OSPF の設定例

例：基本的な OSPF パラメータの設定

次に、OSPF ルーティング プロセスを設定し、プロセス番号 109 を割り当てる例を示します。

```
Device(config)#router ospf 109
Device(config-router)#network 131.108.0.0 255.255.255.0 area 24
```

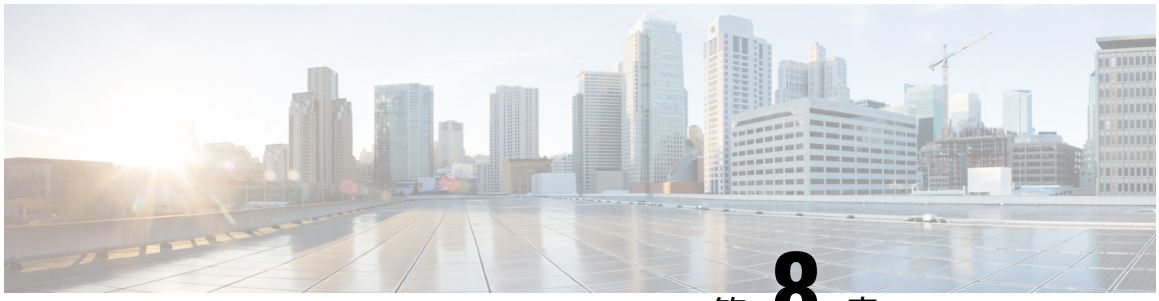
OSPF の機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだ

けを示しています。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 *17 : OSPF* の機能情報

リリース	機能情報
Cisco IOS XE Everest 16.5.1a	この機能が導入されました



第 8 章

OSPFv3 高速コンバージェンス：LSA および SPF スロットリングの設定

OSPFv3 高速コンバージェンス：LSA および SPF スロットリング

Open Shortest Path First バージョン 3（OSPFv3）のリンクステート アドバタイズメント（LSA）および最短パス優先（SPF）スロットリング機能では、ネットワークが不安定な間、OSPFv3 でのリンクステート アドバタイズメント アップデートを低速化するためのダイナミック メカニズムを提供します。さらに LSA のレート制限をミリ秒単位で指定することにより、OSPFv3 コンバージェンス時間の短縮が可能になります。

OSPFv3 高速コンバージェンスについて：LSA および SPF スロットリング

高速コンバージェンス：LSA および SPF スロットリング

OSPFv3 の LSA および SPF スロットリング機能は、ネットワークが不安定な間、OSPFv3 でのリンクステート アドバタイズメント アップデートを低速化するためのダイナミック メカニズムを提供します。さらに LSA のレート制限をミリ秒単位で指定することにより、OSPFv3 コンバージェンス時間の短縮が可能になります。

OSPFv3 ではレート制限 SPF 計算および LSA 生成にスタティック タイマーを使用できます。これらのタイマーを設定することもできますが、使用する値は秒単位で指定するため、OSPFv3 コンバージェンスに制限が課せられます。LSA および SPF スロットリングは、すばやく応答できる高度な SPF および LSA レート制限メカニズムを提供することにより、1 秒未満単位でのコンバージェンスを実現し、長引く不安定期間中にも安定性および保護を提供します。

OSPFv3 高速コンバージェンスの設定方法：LSA および SPF スロットリング

OSPFv3 高速コンバージェンスに対する LSA および SPF タイマーの調整

OSPFv3 高速コンバージェンスに対する LSA および SPF タイマーを調整するには、次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **router ospfv3** [*process-id*]
4. **timers lsa arrival** *milliseconds*
5. **timers pacing flood** *milliseconds*
6. **timers pacing lsa-group** *seconds*
7. **timers pacing retransmission** *milliseconds*

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： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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 ospfv3** [*process-id*] 例： `Device(config)# router ospfv3 1`	IPv4 または IPv6 アドレス ファミリの OSPFv3 ルータ コンフィギュレーション モードをイネーブルにします。
ステップ 4	**timers lsa arrival** *milliseconds* 例： `Device(config-rtr)# timers lsa arrival 300`	ソフトウェアが OSPFv3 ネイバーから同じ LSA を受け入れる最小間隔を設定します。
ステップ 5	**timers pacing flood** *milliseconds* 例： `Device(config-rtr)# timers pacing flood 30`	LSA フラッド パケット ペーシングを設定します。

	コマンドまたはアクション	目的
ステップ 6	**timers pacing lsa-group** *seconds* 例： `Device(config-router)# timers pacing lsa-group 300`	OSPFv3 LSA を収集してグループ化し、リフレッシュ、チェックサム、またはエージングを行う間隔を変更します。
ステップ 7	**timers pacing retransmission** *milliseconds* 例： `Device(config-router)# timers pacing retransmission 100`	IPv4 OSPFv3 での LSA 再送信パケット ペーシングを設定します。

OSPFv3 高速コンバージェンスに対する LSA および SPF スロットリングの設定

OSPFv3 高速コンバージェンスに対する LSA および SPF スロットリングを設定するには、次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **ipv6 router ospf** *process-id*
4. **timers throttle spf** *spf-start spf-hold spf-max-wait*
5. **timers throttle lsa** *start-interval hold-interval max-interval*
6. **timers lsa arrival** *milliseconds*
7. **timers pacing flood** *milliseconds*

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： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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	**ipv6 router ospf** *process-id* 例： `Device(config)# ipv6 router ospf 1`	OSPFv3 ルータ コンフィギュレーション モードをイネーブルにします。
ステップ 4	**timers throttle spf** *spf-start spf-hold spf-max-wait* 例： `Device(config-rtr)# timers throttle spf 200 200 200`	SPF スロットリングをオンにします。

	コマンドまたはアクション	目的
ステップ 5	**timers throttle lsa** *start-interval hold-interval max-interval* 例： `Device(config-rtr)# timers throttle lsa 300 300 300`	OSPFv3 LSA 生成に対するレート制限値を設定します。
ステップ 6	**timers lsa arrival** *milliseconds* 例： `Device(config-rtr)# timers lsa arrival 300`	ソフトウェアが OSPFv3 ネイバーから同じ LSA を受け入れる最小間隔を設定します。
ステップ 7	**timers pacing flood** *milliseconds* 例： `Device(config-rtr)# timers pacing flood 30`	LSA フラッド パケット ペーシングを設定します。

OSPFv3 高速コンバージェンスの設定例：LSA および SPF スロットリング

OSPFv3 高速コンバージェンスに対する LSA および SPF スロットリングの設定例

次に、SPF および LSA スロットリング タイマーの設定値を表示する例を示します。

```
Device# show ipv6 ospf

Routing Process "ospfv3 1" with ID 10.9.4.1
Event-log enabled, Maximum number of events: 1000, Mode: cyclic
 It is an autonomous system boundary router
 Redistributing External Routes from,
    ospf 2
 Initial SPF schedule delay 5000 msecs
 Minimum hold time between two consecutive SPFs 10000 msecs
 Maximum wait time between two consecutive SPFs 10000 msecs
 Minimum LSA interval 5 secs
 Minimum LSA arrival 1000 msecs
```

その他の参考資料

関連資料

関連項目	マニュアル タイトル
IPv6 アドレッシングと接続	『*IPv6 Configuration Guide*』
OSPFv3 高速コンバージェンス：LSA および SPF スロットリング	*OSPF Shortest Path First* スロットリングモジュール

標準および RFC

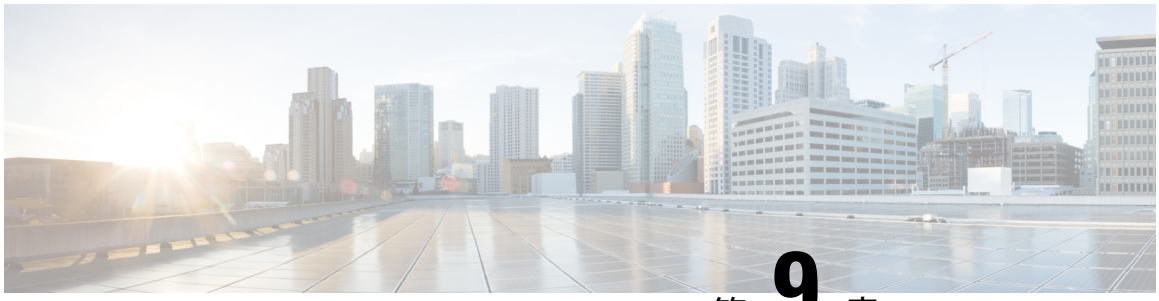
標準/RFC	タイトル
IPv6 に関する RFC	IPv6 RFCs

OSPFv3 高速コンバージェンス：LSA および SPF スロットリングの機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。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 *18 : OSPFv3* 高速コンバージェンス：*LSA* および *SPF* スロットリングの機能情報

リリース	機能情報
Cisco IOS XE Gibraltar 16.11.1	この機能が導入されました。



第 9 章

IPsec を使用した OSPFv3 認証サポートの設定

IPsec を使用した OSPFv3 認証サポートに関する情報

ここでは、IPsec および OSPFv3 仮想リンクを使用した OSPFv3 認証サポートについて説明します。

IPsec を使用した OSPFv3 認証サポートの概要

OSPFv3 パケットが変更されてデバイスに再送信されることにより、デバイスがシステム管理者にとって望ましくない動作をすることにならないように、OSPFv3 パケットを認証する必要があります。OSPFv3 は、IPsec セキュアソケットを使用して OSPFv3 パケットに認証を追加します。

OSPFv3 では、認証をイネーブルにするために IPsec を使用する必要があります。OSPFv3 で使用するために必要な IPsec は暗号イメージのみに含まれるため、認証を使用するには暗号イメージが必要です。

OSPFv3 では、認証フィールドが OSPFv3 パケット ヘッダーから削除されています。IPv6 で OSPFv3 を実行する場合、ルーティング変更の整合性、認証、および機密性を確保するために、OSPFv3 には IPv6 認証ヘッダーまたは IPv6 カプセル化セキュリティペイロード（ESP）ヘッダーが必要です。IPv6 認証ヘッダーおよび ESP 拡張ヘッダーを使用すると、OSPFv3 に認証および機密性を提供できます。

IPsec 認証ヘッダーを使用するには、**ipv6 ospf authentication** コマンドをイネーブルにする必要があります。IPsec ESP ヘッダーを使用するには、**ipv6 ospf encryption** コマンドをイネーブルにする必要があります。ESP ヘッダーは、単独で適用することも、認証ヘッダーとともに適用することもできます。ESP を使用した場合、暗号化と認証の両方が提供されます。セキュリティ サービスは、通信する 1 組のホスト、通信する 1 組のセキュリティ ゲートウェイ、またはセキュリティ ゲートウェイとホストの間に提供できます。

IPsec を設定するには、セキュリティポリシーを設定する必要があります。これは、Security Policy Index（SPI）とキーの組み合わせです（このキーはハッシュ値の作成および検証に使用されます）。OSPFv3 の IPsec は、インターフェイスまたは OSPFv3 エリアに対して設定できます。セキュリティを強化するには、IPsec を設定する各インターフェイスで異なるポリシーを設定する必要があります。OSPFv3 エリアに対して IPsec を設定した場合、ポリシーはそのエリア内のすべてのインターフェイス（IPsec が直接設定されているインターフェイスを除く）に適用されます。OSPFv3 に対して IPsec を設定すると、IPsec は見えなくなります。

アプリケーションは、IPsecure ソケットを使用することで、セキュアソケットのオープン、リッスン、およびクローズが可能になり、トラフィックが保護されます。また、アプリケーションと Secure Socket Layer の間のバインディングにより、Secure Socket Layer は、接続のオープンやイベントのクローズなど、ソケットへの変更をアプリケーションに通知できます。IPsecure ソケットは、ソケットを識別できます。つまり、セキュリティを必要とするトラフィックを伝送するローカルおよびリモートのアドレス、マスク、ポート、およびプロトコルを識別できます。

各インターフェイスのセキュア ソケット ステートは、次のいずれかになります。

- NULL：エリアに対して認証が設定されていれば、インターフェイスに対してセキュア ソケットを作成しません。
- DOWN：インターフェイス（またはインターフェイスが含まれるエリア）に対して IPsec は設定されていますが、OSPFv3 がこのインターフェイスに対するセキュアソケットの作成を IPsec に要求していないか、またはエラー条件が存在します。

（注） DOWN 状態の間は、OSPFv3 はパケットを受け入れたり、送信したりすることはありません。

- GOING UP：OSPFv3 はセキュア ソケットを IPsec に要求し、IPsec からの CRYPTO_SS_SOCKET_UP メッセージを待っています。
- UP：OSPFv3 は IPsec から CRYPTO_SS_SOCKET_UP メッセージを受信しました。
- CLOSING：インターフェイスのセキュア ソケットはクローズされています。インターフェイスに対して新しいソケットがオープンされることがあります。この場合、現在のセキュア ソケットは DOWN ステートに移行します。オープンされない場合、インターフェイスは UNCONFIGURED となります。
- UNCONFIGURED：インターフェイス上に認証は設定されていません。

OSPFv3 仮想リンク

仮想リンクごとに、プライマリセキュリティ情報データブロックが作成されます。各インターフェイスでセキュア ソケットをオープンする必要があるため、トランジット エリア内のインターフェイスごとに、対応するセキュリティ情報データブロックが存在することになります。セキュアソケットステートは、インターフェイスのセキュリティ情報データブロック内に保持されます。プライマリセキュリティ情報データブロック内の**ステート**フィールドは、対応する仮想リンクに対してオープンされたすべてのセキュアソケットのステータスを示します。すべてのセキュアソケットが UP の場合、仮想リンクのセキュリティステートは UP に設定されます。

IPsec が設定された仮想リンク上を送信されるパケットは、事前に決定された送信元アドレスと宛先アドレスを使用する必要があります。エリアのデバイスのエリア内プレフィックス リンクステート アドバタイズメント（LSA）で見つかった最初のローカルエリアアドレスが、送信元アドレスとして使用されます。この送信元アドレスはエリアのデータ構造に保存されます。セキュアソケットがオープンされ、パケットが対応する仮想リンク経由で送信されるときにこの送信元アドレスが使用されます。送信元アドレスが選択されるまで、仮想リンクはポイントツーポイント ステートに移行しません。また、送信元アドレスまたは宛先アドレスが変更された場合は、以前のセキュア ソケットをクローズして、新しいセキュア ソケットをオープンする必要があります。

（注） 仮想リンクは、IPv4 アドレスファミリについてはサポートされません。

IPsec を使用した OSPFv3 認証サポートの設定方法

ここでは、インターフェイスで認証を定義する方法と、OSPFv3 エリアで認証を定義する方法について説明します。

インターフェイスでの認証の定義

インターフェイスで認証を定義するには、次の手順を実行します。

始める前に

インターフェイスで IPsec を設定する前に、そのインターフェイスで OSPFv3 を設定する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例：	特権 EXEC モードを有効にします。

	コマンドまたはアクション	目的
	`Device> enable`	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： `Device# 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	**interface** *type number* 例： `Device(config)# interface ethernet 1/0/1`	インターフェイスを設定します。

ステップ 4	次のいずれかを選択します。
	• **ospfv3 authentication** {{ **ipsec spi** *spi* {**md5** | **sha1**}} | { *key-encryption-type key* } | **null**}
	• **ipv6 ospf authentication** {**null** | **ipsec spi** *spi authentication-algorithm* [*key-encryption-type*] [*key*]}
	例：
	`Device(config-if)# ospfv3 authentication md5 0 27576134094768132473302031209727`
	または
	`Device(config-if)# ipv6 ospf authentication ipsec spi 500 md5 1234567890abcdef1234567890abcdef`
		インターフェイスの認証タイプを指定します。

OSPFv3 エリア内の認証の定義

OSPFv3 エリア内で認証を定義するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 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： `Device# 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	**ipv6 router ospf** *process-id* 例： `Device(config)# ipv6 router ospf 1`	OSPFv3 ルータ コンフィギュレーション モードをイネーブルにします。

	コマンドまたはアクション	目的
ステップ 4	**area** *area-id* **authentication ipsec spi** *spi authentication-algorithm* [*key-encryption-type*] *key* 例： `Device(config-router)# area 1 authentication ipsec spi 678 md5 1234567890ABCDEF1234567890ABCDEF`	OSPFv3 エリア内の認証をイネーブルにします。

OSPFv3 IPSec ESP 暗号化および認証の設定方法

ここでは、インターフェイスで暗号化を定義する方法、OSPFv3 エリアで暗号化を定義する方法、および OSPFv3 エリアで仮想リンクの認証と暗号化を定義する方法について説明します。

インターフェイスでの暗号化の定義

インターフェイスで暗号化を定義するには、次の手順を実行します。

始める前に

インターフェイスで IPsec を設定する前に、そのインターフェイスで OSPFv3 を設定する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 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： `Device# 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	**interface** *type number* 例： `Device(config)# interface ethernet 1/0/1`	インターフェイスを設定します。
ステップ 4	次のいずれかを選択します。 • **ospfv3 authentication** { **ipsec spi** *spi* **esp** *encryption-algorithm key-encryption-type key authentication-algorithm key-encryption-type key* | **null**} • **ipv6 ospf authentication** { **ipsec spi** *spi* **esp** {*encryption-algorithm* [*key-encryption-type*] *key* |	インターフェイスに暗号化タイプを指定します。

	コマンドまたはアクション	目的

	null} *authentication-algorithm* [*key-encryption-type*] *key*] | **null**}
	例：
	`Device(config-if)# ospfv3 encryption ipsec spi 1001 esp null md5 0 275761340947681324733020312097 27`
	または
	`Device(config-if)# ipv6 ospf encryption ipsec spi 1001 esp null sha1 123456789A123456789B123456789C123456789D`	

OSPFv3 エリア内の暗号化の定義

OSPFv3 エリアで暗号化を定義するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 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： `Device# 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	**ipv6 router ospf** *process-id* 例： `Device(config)# ipv6 router ospf 1`	OSPFv3 ルータ コンフィギュレーション モードをイネーブルにします。
ステップ 4	**area** *area-id* **encryption ipsec spi** *spi* **esp** {*encryption-algorithm* [*key-encryption-type*] *key* | **null**} *authentication-algorithm* [*key-encryption-type*] *key* 例： `Device(config-router)# area 1 encryption ipsec spi 500 esp null md5 1aaa2bbb3ccc4ddd5eee6fff7aaa8bbb`	OSPFv3 エリア内の暗号化をイネーブルにします。

OSPFv3 エリア内の仮想リンクに対する認証および暗号化の定義

OSPFv3 エリア内の仮想リンクに対する認証および暗号化を定義するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 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： `Device# 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	**ipv6 router ospf** *process-id* 例： `Device(config)# ipv6 router ospf 1`	OSPFv3 ルータ コンフィギュレーション モードをイネーブルにします。
ステップ 4	**area** *area-id* **virtual-link** *router-id* **authentication ipsec spi** *spi authentication-algorithm* [*key-encryption-type*] *key* 例： `Device(config-router)# area 1 virtual-link 10.0.0.1 authentication ipsec spi 940 md5 1234567890ABCDEF1234567890ABCDEF`	OSPFv3エリア内の仮想リンクに対して認証をイネーブルにします。
ステップ 5	**area** *area-id* **virtual-link** *router-id* **authentication ipsec spi** *spi* **esp** {*encryption-algorithm* [*key-encryption-type*] *key* | **null**} *authentication-algorithm* [*key-encryption-type*] *key* 例： `Device(config-router)# area 1 virtual-link 10.1.0.1 hello-interval 2 dead-interval 10 encryption ipsec spi 3944 esp null sha1 123456789A123456789B123456789C123456789D`	OSPFv3 エリア内の仮想リンクに対して暗号化をイネーブルにします。

IPsec を使用した OSPFv3 認証サポートの設定例

ここでは、IPsec を使用した OSPFv3 認証サポートのさまざまな設定例を示します。

例：インターフェイスでの認証の定義

次に、イーサネット インターフェイス 1/0/1 で認証を定義する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# interface Ethernet1/0/1
Device(config-if)# ipv6 enable
Device(config-if)# ipv6 ospf 1 area 0
```

```
Device(config-if)# ipv6 ospf authentication ipsec spi 500 md5
1234567890ABCDEF1234567890ABCDEF
Device(config-if)# exit
Device(config)# interface Ethernet1/0/1
Device(config-if)# ipv6 enable
Device(config-if)# ipv6 ospf authentication null
Device(config-if)# ipv6 ospf 1 area 0
```

例：OSPFv3 エリア内の認証の定義

次に、OSPFv3 エリア 0 で認証を定義する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# ipv6 router ospf 1
Device(config-router)# router-id 10.11.11.1
Device(config-router)# area 0 authentication ipsec spi 1000 md5
1234567890ABCDEF1234567890ABCDEF
```

OSPFv3 IPSec ESP 暗号化および認証の設定例

ここでは、OSPFv3 IPsec ESP 暗号化および認証を確認する例を示します。

例：OSPFv3 エリアでの暗号化の確認

次に、**show ipv6 ospf interface** コマンドの出力例を示します。

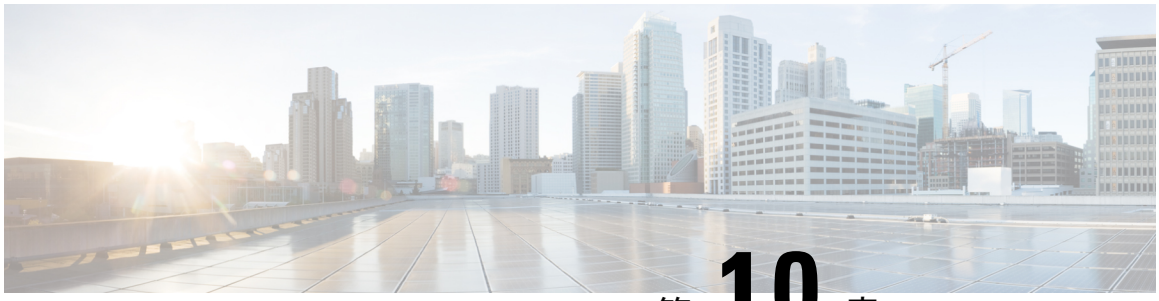
```
Device> enable
Device# show ipv6 ospf interface

Ethernet1/0/1 is up, line protocol is up
  Link Local Address 2001:0DB1:A8BB:CCFF:FE00:6E00, Interface ID 2
  Area 0, Process ID 1, Instance ID 0, Router ID 10.10.10.1
  Network Type BROADCAST, Cost:10
  MD5 Authentication (Area) SPI 1000, secure socket state UP (errors:0)
  Transmit Delay is 1 sec, State BDR, Priority 1
  Designated Router (ID) 10.11.11.1, local address 2001:0DB1:A8BB:CCFF:FE00:6F00
  Backup Designated router (ID) 10.10.10.1, local address
FE80::A8BB:CCFF:FE00:6E00
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:03
  Index 1/1/1, flood queue length 0
  Next 0x0(0)/0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 10.11.11.1  (Designated Router)
  Suppress hello for 0 neighbor(s)
```

IPsec を使用した OSPFv3 認証サポートの機能履歴と機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。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 *19 : IPsec* を使用した *OSPFv3* 認証サポートの機能履歴

機能名	リリース	機能情報
IPsec を使用した OSPFv3 認証サポート	Cisco IOS XE Fuji 16.8.1a	OSPFv3 は、IPsec セキュアソケットを使用して OSPFv3 パケットに認証を追加します。



第 10 章

OSPFv3 認証トレーラの設定

OSPFv3 認証トレーラに関する情報

OSPFv3 認証トレーラ機能（RFC 7166 で定義されている）は、Open Shortest Path First バージョン 3（OSPFv3）プロトコルパケットを認証する代替メカニズムを提供します。OSPFv3 認証トレーラの前は、OSPFv3 IPsec（RFC 4552 で定義されている）がプロトコルパケットの認証を行う唯一のメカニズムでした。OSPFv3 認証トレーラー機能は、シーケンス番号を介したパケットリプレイ保護も提供し、プラットフォームに依存しません。

非 IPsec 暗号化認証を実行するため、デバイスは OSPFv3 パケットの末尾に特別なデータブロック（認証トレーラ）を追加します。認証トレーラの長さは OSPFv3 パケットの長さに含まれず、IPv6 ペイロード長に含まれます。リンクローカルシグナリング（LLS）ブロックは OSPFv3 hello パケットおよびデータベース記述パケットの **OSPFv3 Options** フィールドの L-bit 設定で確立されます。存在する場合、LLS データブロックは OSPFv3 パケットとともに暗号化認証計算に含まれます。

新しい認証トレーラビットは **OSPFv3 Options** フィールドに導入されています。OSPFv3 デバイスは、このリンク上のすべてのパケットに認証トレーラが含まれていることを示すため、OSPFv3 hello パケットおよびデータベース記述パケットで認証トレーラビットを設定する必要があります。OSPFv3 hello パケットおよびデータベース記述パケットの場合、認証トレーラビットは認証トレーラが存在することを示します。他の OSPFv3 パケットタイプでは、OSPFv3 hello およびデータベース記述設定の OSPFv3 認証トレーラビット設定は OSPFv3 ネイバーデータ構造に保持されます。**OSPFv3 Options** フィールドを含まない OSPFv3 パケットタイプでは、ネイバーデータ構造の設定を使用して認証トレーラが必要かどうかを決定します。認証トレーラビットは、認証トレーラを含むすべての OSPFv3 hello パケットおよびデータベース記述パケットで設定する必要があります。

認証トレーラを設定するには、OSPFv3 では既存の Cisco IOS **key chain** コマンドを使用します。発信 OSPFv3 パケットでは、次のルールを使用してキー チェーンからキーを選択します。

- 最後に期限切れになるキーを選択します。
- 2 つのキーの終了時間が同じ場合、最も大きいキー ID のキーを選択します。

セキュリティ アソシエーション ID は認証アルゴリズムと秘密鍵にマッピングされ、メッセージダイジェストの生成および検証に使用されます。認証が設定されていても、最後の有効なキーが期限切れになると、パケットはそのキーを使用して送信されます。syslog メッセージも生成されます。有効なキーが使用できない場合は、トレーラ認証なしでパケットが送信されます。パケットが受信されると、そのキーのデータを検索するためにキー ID が使用されます。キーチェーンにキー ID が見つからない、またはセキュリティ アソシエーションが有効でない場合、パケットはドロップされます。そうでない場合、パケットはキー ID で設定されたアルゴリズムとキーを使用して検証されます。キーチェーンはキーのライフタイムを使用するロールオーバーをサポートします。新しいキーは、将来設定する開始時間の送信でキーチェーンに追加できます。この設定により、キーが実際に使用される前に新しいキーをすべてのデバイスで設定できます。

hello パケットの優先順位はその他の OSPFv3 パケットより高いため、発信インターフェイスで順序変更することができます。この再順序付けにより、隣接デバイスでシーケンス番号の検証に関する問題が発生することがあります。シーケンスの不一致を防ぐには、OSPFv3 でパケット タイプごとに個別にシーケンス番号を検証します。認証手順の詳細については、RFC 7166 を参照してください。

ネットワークでの認証トレーラ機能の初期ロールオーバー時に、認証ルートで設定されているデバイスと展開モードを使用してまだ設定されていないデバイスの隣接関係を維持できます。**authentication mode deployment** コマンドを使用して展開モードが設定されている場合、パケットの処理が異なります。発信パケットの場合は、認証トレーラが設定されていても、OSPF チェックサムが計算されます。着信パケットの場合は、認証トレーラのないパケットまたは認証ハッシュが正しくないパケットはドロップされます。展開モードでは、**show ospfv3 neighbor** *detail* コマンドによって最後のパケット認証ステータスが表示されます。**authentication mode normal** コマンドを使用して通常モードに設定する前に、この情報を使用して、認証トレーラ機能が動作しているかどうかを確認できます。

OSPFv3 認証トレーラの設定方法

OSPFv3 認証トレーラを設定するには、次の手順を実行します。

始める前に

OSPFv3 認証トレーラを設定するには、認証キーが必要です。認証キーの設定の詳細については、「プロトコル独立機能」の「認証キーの設定方法」を参照してください。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 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： `Device# 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	**interface** *type number* 例： `Device(config)# interface GigabitEthernet 2/0/1`	インターフェイス タイプおよび番号を指定します。
ステップ 4	**ospfv3** [*pid*] [**ipv4** | **ipv6**] **authentication** { **key-chain** *chain-name* | **null**} 例： `Device(config-if)# ospfv3 1 ipv6 authentication key-chain ospf-1`	OSPFv3 インターフェイスの認証タイプを指定します。
ステップ 5	**router ospfv3** [*process-id*] 例： `Device(config-if)# router ospfv3 1`	OSPFv3 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 6	**address-family ipv6 unicast** 例： `Device(config-router)# address-family ipv6 unicast`	OSPFv3 プロセスに IPv6 アドレス ファミリを設定し、IPv6 アドレス ファミリ コンフィギュレーション モードを開始します。
ステップ 7	**area** *area-id* **authentication** {**key-chain** *chain-name* | **null**} 例： `Device(config-router-af)# area 1 authentication key-chain ospf-chain-1`	OSPFv3 エリア内のすべてのインターフェイスの認証トレーラを設定します。
ステップ 8	**area** *area-id* **virtual-link** *router-id* **authentication key-chain** *chain-name* 例： `Device(config-router-af)# area 1 virtual-link 1.1.1.1 authentication key-chain ospf-chain-1`	仮想リンクの認証を設定します。
ステップ 9	**area** *area-id* **sham-link** *source-address destination-address* **authentication key-chain** *chain-name* 例：	模造リンクの認証を設定します。

	コマンドまたはアクション	目的
	`Device(config-router-af)# area 1 sham-link 1.1.1.1 1.1.1.0 authentication key-chain ospf-chain-1`	
ステップ 10	**authentication mode** {**deployment** |**normal**} 例： `Device(config-router-af)# authentication mode deployment`	（任意）OSPFv3インスタンスに使用する認証のタイプを指定します。 **deployment** キーワードは、認証を設定済みのデバイスと未設定のデバイス間の隣接関係を表示します。
ステップ 11	**end** 例： `Device(config-router-af)# end`	IPv6 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、特権 EXEC モードに戻ります。
ステップ 12	**show ospfv3 interface** 例： `Device# show ospfv3`	（任意）OSPFv3関連のインターフェイス情報を表示します。
ステップ 13	**show ospfv3 neighbor** [*detail*] 例： `Device# show ospfv3 neighbor detail`	（任意）OSPFv3ネイバー情報をインターフェイスごとに表示します。
ステップ 14	**debug ospfv3** 例： `Device# debug ospfv3`	（任意）OSPFv3 のデバッグ情報を表示します。

OSPFv3 認証トレーラの設定例

ここでは、OSPFv3 認証トレーラを設定する方法と OSPFv3 認証トレーラの設定を確認する方法の例を示します。

例：OSPFv3 認証トレーラの設定

次に、ギガビット イーサネット インターフェイス 1/0/1 で認証トレーラを定義する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# interface GigabitEthernet 1/0/1
Device(config-if)# ospfv3 1 ipv6 authentication key-chain ospf-1
Device(config-if)# router ospfv3 1
Device(config-router)# address-family ipv6 unicast
Device(config-router-af)# area 1 authentication key-chain ospf-1
Device(config-router-af)# area 1 virtual-link 1.1.1.1 authentication key-chain ospf-1
```

```
Device(config-router-af)# area 1 sham-link 1.1.1.1 authentication key-chain ospf-1
Device(config-router-af)# authentication mode deployment
Device(config-router-af)# end
Device(config)# key chain ospf-1
Device(config-keychain)# key 1
Device(config-keychain-key)# key-string ospf
Device(config-keychain-key)# cryptographic-algorithm hmac-sha-256
!
```

例：OSPFv3 認証トレーラの確認

次に、**show ospfv3** コマンドの出力例を示します

```
Device# show ospfv3
 OSPFv3 1 address-family ipv6
 Router ID 1.1.1.1
…
RFC1583 compatibility enabled
 Authentication configured with deployment key lifetime
 Active Key-chains:
  Key chain ospf-1: Send key 1, Algorithm HMAC-SHA-256, Number of interfaces 1
    Area BACKBONE(0)
```

次に、**show ospfv3 neighbor detail** コマンドの出力例を示します

```
Device# show ospfv3 neighbor detail
OSPFv3 1 address-family ipv6 (router-id 2.2.2.2)
 Neighbor 1.1.1.1
    In the area 0 via interface GigabitEthernet0/0
    Neighbor: interface-id 2, link-local address FE80::A8BB:CCFF:FE01:2D00
    Neighbor priority is 1, State is FULL, 6 state changes
    DR is 2.2.2.2 BDR is 1.1.1.1
    Options is 0x000413 in Hello (V6-Bit, E-Bit, R-Bit, AT-Bit)
    Options is 0x000413 in DBD (V6-Bit, E-Bit, R-Bit, AT-Bit)
    Dead timer due in 00:00:33
    Neighbor is up for 00:05:07
    Last packet authentication succeed
    Index 1/1/1, retransmission queue length 0, number of retransmission 0
    First 0x0(0)/0x0(0)/0x0(0) Next 0x0(0)/0x0(0)/0x0(0)
    Last retransmission scan length is 0, maximum is 0
    Last retransmission scan time is 0 msec, maximum is 0 msec
```

次に、**show ospfv3 interface** コマンドの出力例を示します

```
Device# show ospfv3 interface
GigabitEthernet1/0/1 is up, line protocol is up
  Cryptographic authentication enabled
    Sending SA: Key 25, Algorithm HMAC-SHA-256 - key chain ospf-1
    Last retransmission scan time is 0 msec, maximum is 0 msec
```

OSPFv3 認証トレーラに関する追加情報

関連資料

関連項目	マニュアル タイトル
OSPF機能の設定	*IP* ルーティング：*OSPF* 設定ガイド

標準および RFC

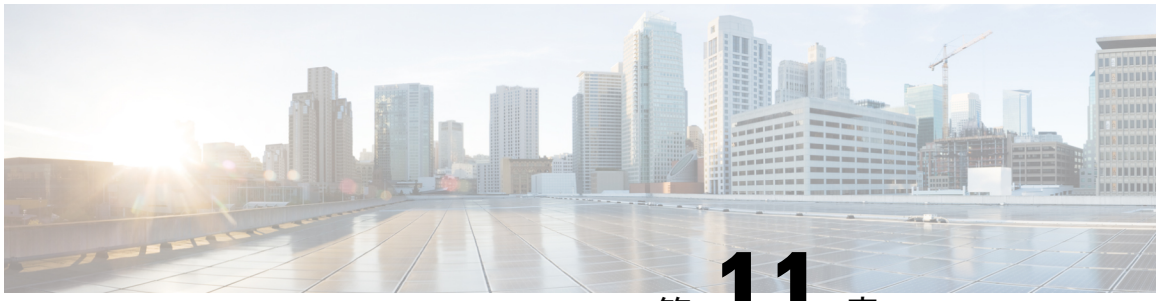
標準/RFC	マニュアル タイトル
RFC 7166	OSPFv3認証トレーラのサポートに関する RFC
RFC 6506	OSPFv3認証トレーラのサポートに関する RFC
RFC 4552	OSPFv3 の認証/機密性に関する RFC

OSPFv3 認証トレーラの機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。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 *20 : OSPFv3* 認証トレーラの機能情報

機能名	リリース	機能情報
OSPFv3 認証トレーラ	Cisco IOS XE Fuji 16.8.1a	OSPFv3認証トレーラ機能は、既存の OSPFv3 IPsec 認証の代替として OSPFv3 プロトコル パケットを認証するメカニズムを提供します。



第 11 章

OSPFv3 のルート再配布数制限の設定

OSPFv3 のルート再配布数の制限に関する制約事項

この機能は、IPv6 アドレスファミリについてのみサポートされています。

OSPFv3 のルート再配布数制限の前提条件

再配布するには、ネットワークで Open Shortest Path First バージョン 3（OSPFv3）を、別のプロトコルまたは別の OSPFv3 プロセスとともに設定する必要があります。

OSPFv3 のルート再配布数制限について

OSPFv3 は、別のプロトコルまたは別の OSPFv3 プロセスから OSPFv3 内に再配布できるプレフィックスの最大数をユーザーが定義する機能をサポートします。こうした制限により、デバイスが大量のルートの再配布でフラッディングを起こすことを回避できます。

たとえば、ボーダー ゲートウェイ プロトコル（BGP）の OSPFv3 への再配布が可能なネットワークで OSPFv3 に多数の IP ルートが送信されると、ネットワークで深刻なフラッディング状態になるおそれがあります。ルートの再配布数を制限すると、この潜在的な問題を回避できます。

OSPFv3 のルート再配布数制限を設定する方法

ここでは、OSPFv3 のルート再配布数制限の設定について説明します。

（注） 以下の手順は相互に排他的です。つまり、再配布されるルートの数を制限するか、OSPFv3 に再配布されるルートの数に関する警告を要求するかのいずれかを実行できます。

OSPFv3 のルート再配布数の制限

このタスクでは、OSPFv3 のルート再配布数を制限する方法について説明します。ルート再配布数が設定された最大数に到達すると、これ以上のルートは再配信されません。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 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： `Device# 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 ospfv3** *process-id* 例： `Device(config)# router ospfv3 1`	OSPFv3 ルーティング プロセスを設定します。
ステップ 4	**address-family ipv6** [**unicast**] 例： `Device(config-router)# address-family ipv6 unicast`	IPv6アドレスファミリ コンフィギュレーションモードを開始します。
ステップ 5	**redistribute** *protocol* [*process-id*] [*as-number*] [**include-connected** {**level-1** | **level-1-2** | **level-2**} [**metric** *metric-value*] [**metric-type** *type-value*] [**nssa-only**] [**tag** *tag-value*] [**route-map** *map-tag*] 例： `Device(config-router-af)# redistribute eigrp 10`	ルートを 1 つのルーティング ドメインから他のルーティング ドメインに再配布します。
ステップ 6	**redistribute maximum-prefix** *maximum* [*threshold*] 例：	OSPFv3 への再配布が許可される IPv6 プレフィックスの最大数を設定します。 • 引数 *maximum* のデフォルト値はありません。

	コマンドまたはアクション	目的
	`Device(config-router-af)# redistribute maximum-prefix 100 80`	• *threshold* 値はデフォルトで 75% に設定されています。 （注） **warning-only** キーワードをこのコマンドで設定すると、再配布数の制限は設定されず、警告メッセージがログに記録されるようになります。
ステップ 7	**exit-address-family** 例： `Device(config-router-af)# exit-address-family`	IPv6 アドレス ファミリ コンフィギュレーション モードを終了します。
ステップ 8	**end** 例： `Device(config-router)# end`	ルータ コンフィギュレーション モードを終了します。

OSPFv3 へのルートの再配布数に関する警告メッセージの要求

OSPFv3 に再配布されるルートの数が設定制限を超えたときの警告メッセージを要求するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 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： `Device# 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 ospfv3** *process-id* 例： `Device(config)# router ospfv3 1`	OSPFv3 ルーティング プロセスを設定します。
ステップ 4	**address-family ipv6** [**unicast**] 例： `Device(config-router)# address-family ipv6 unicast`	IPv6 アドレス ファミリ コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 5	**redistribute** *protocol* [*process-id*] [*as-number*] [**include-connected** {**level-1** | **level-1-2** | **level-2**} [**metric** *metric-value*] [**metric-type** *type-value*] [**nssa-only**] [**tag** *tag-value*] [**route-map** *map-tag*] 例： `Device(config-router-af)# redistribute eigrp 10`	ルートを 1 つのルーティング ドメインから他のルーティング ドメインに再配布します。
ステップ 6	**redistribute maximum-prefix** *maximum* [*threshold*] [**warning-only**] 例： `Device(config-router-af)# redistribute maximum-prefix 100 80 warning-only`	IP プレフィックスの最大数が OSPFv3 内に再配布されたときに警告メッセージのログが記録されます。 • **warning-only** キーワードが含まれているため、OSPFv3 へのプレフィックスの再配布数に制限は設定されません。 • 引数 *maximum* のデフォルト値はありません。 • *threshold* 値はデフォルトで 75% に設定されています。 • ここでは、1000 の 80%（800 個のルート再配布）で警告する場合と、1000 個のルート再配布で警告する場合の、2 つの例について説明します。
ステップ 7	**end** 例： `Device(config-router)# end`	ルータ コンフィギュレーション モードを終了します。

OSPFv3 のルート再配布数制限の設定例

ここでは、OSPFv3 のルート再配布数制限の設定例を示します。

例：OSPFv3 のルート再配布数の制限

次に、OSPFv3 プロセス 1 に再配布できるプレフィックスの最大数に 1200 を設定する例を示します。制限に達する前に、再配布されたプレフィックス数が 1200 の 80%（960 個のプレフィックス）に達すると、警告メッセージのログが記録されます。制限に達すると、もう 1 種類の警告メッセージがログに記録され、これ以降、プレフィックスは再配布されなくなります。

```
Device> enable
Device# configure terminal
Device(config)# router ospfv3 1
Device(config-router)# address-family ipv6
Device(config-router-af)# redistribute static subnets
Device(config-router-af)# redistribute maximum-prefix 1200 80
```

例：ルートの再配布数に関する警告メッセージの要求

次に、プレフィックスの再配布数が 600 の 85%（510 個のプレフィックス）に達した場合とルートの再配布数が 600 に達した場合にそれぞれ警告メッセージを記録するように設定する例を示します。ただし、再配布されるルート数は制限されません。

```
Device> enable
Device# configure terminal
Device(config)# router ospfv3 11
Device(config-router)# address-family ipv6
Device(config-router-af)# redistribute eigrp 10 subnets
Device(config-router-af)# redistribute maximum-prefix 600 85 warning-only
```

OSPFv3 のルート再配布数制限のモニタリング

ルート再配布数制限をモニターするには、次の表の特権 EXEC コマンドを使用します。

表 *21 : OSPFv3* のルート再配布数制限をモニターするためのコマンド

コマンド	目的
show ipv6 ospf [*process-id*] または **show ospfv3 ipv6** [*process-id*]	OSPFv3 ルーティング プロセスに関する一般情報を表示します。出力には、プレフィックスの再配布数の最大制限値と、警告メッセージが生成されるしきい値が含まれます。

その他の参考資料

関連資料

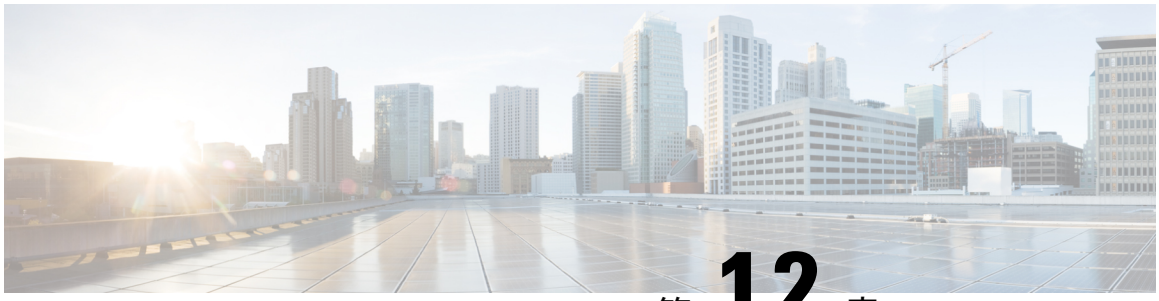
関連項目	マニュアル タイトル
この章で使用するコマンドの完全な構文および使用方法の詳細。	次のドキュメントのルーティングに関する項を参照してください：*Command Reference (Catalyst 9300 Series Switches)*

OSPFv3 のルート再配布数制限の機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。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 *22 : OSPFv3* のルート再配布数制限の機能情報

機能名	リリース	機能情報
OSPFv3 のルート再配布数の制限	Cisco IOS XE Gibraltar 16.11.1	OSPFv3 は、別のプロトコルまたは別の OSPFv3 プロセスから OSPFv3 内に再配布できるプレフィックスの最大数をユーザーが定義する機能をサポートします。こうした制限により、デバイスが大量のルートの再配布でフラッディングを起こすことを回避できます。



第 12 章

EIGRP の設定

EIGRP に関する情報

EIGRP は IGRP のシスコ独自の拡張バージョンです。EIGRP は IGRP と同じディスタンス ベクトル アルゴリズムおよび距離情報を使用しますが、EIGRP では収束性および動作効率が大幅に改善されています。

コンバージェンステクノロジーには、拡散更新アルゴリズム（DUAL）と呼ばれるアルゴリズムが採用されています。DUAL を使用すると、ルート計算の各段階でループが発生しなくなり、トポロジの変更に関連するすべてのデバイスを同時に同期できます。トポロジ変更の影響を受けないルータは、再計算に含まれません。

IP EIGRP を導入すると、ネットワークの幅が広がります。RIP の場合、ネットワークの最大幅は 15 ホップです。EIGRP メトリックは数千ホップをサポートするほど大きいため、ネットワークを拡張するときに問題となるのは、トランスポート レイヤのホップ カウンタだけです。IP パケットが 15 台のルータを経由し、宛先方向のネクスト ホップが EIGRP によって取得されている場合だけ、EIGRP は転送制御フィールドの値を増やします。RIP ルートを宛先へのネクスト ホップとして使用する場合、転送制御フィールドでは、通常どおり値が増加します。

EIGRP IPv6

スイッチは、IPv6 の Enhanced Interior Gateway Routing Protocol（EIGRP）をサポートしています。IPv6 の EIGRP は稼働するインターフェイス上で設定されるため、グローバルな IPv6 アドレスは不要です。Network Essentials を実行しているスイッチは EIGRPv6 スタブルーティングのみをサポートします。

EIGRP IPv6 インスタンスでは、実行する前に暗示的または明示的なルータ ID が必要です。暗示的なルータ ID はローカルの IPv6 アドレスを基にして作成されるため、すべての IPv6 ノー

ドには常に使用可能なルータ ID があります。ただし、EIGRP IPv6 は IPv6 ノードのみが含まれるネットワークで稼働するため、使用可能な IPv6 ルータ ID がない場合があります。

IPv6 用の EIGRP の設定については、「*IPv6* 用の *EIGRP* の設定」を参照してください。

IPv6 用の EIGRP の詳細については、Cisco.com の『*Cisco IOS IPv6 Configuration Library*』を参照してください。

EIGRP の機能

EIGRP には次の機能があります。

- 高速コンバージェンス
- 差分更新：宛先のステートが変更された場合、ルーティング テーブルの内容全体を送信する代わりに差分更新を行い、EIGRP パケットに必要な帯域幅を最小化します。
- 低い CPU 使用率：完全更新パケットを受信ごとに処理する必要がないため、CPU 使用率が低下します。
- プロトコルに依存しないネイバー探索メカニズム：このメカニズムを使用し隣接ルータに関する情報を取得します。
- 可変長サブネット マスク（VLSM）
- 任意のルート集約
- 大規模ネットワークへの対応

EIGRP コンポーネント

EIGRP には次に示す 4 つの基本コンポーネントがあります。

- ネイバー探索および回復：直接接続されたネットワーク上の他のルータに関する情報を動的に取得するために、ルータで使用されるプロセスです。また、ネイバーが到達不能または動作不能になっていることを検出するためにも使用されます。ネイバー探索および回復は、サイズの小さな hello パケットを定期的に送信することにより、わずかなオーバーヘッドで実現されます。hello パケットが受信されているかぎり、Cisco ISO ソフトウェアは、ネイバーが有効に機能していると学習します。このように判別された場合、隣接ルータはルーティング情報を交換できます。
- Reliable Transport Protocol：EIGRP パケットをすべてのネイバーに確実に、順序どおりに配信します。マルチキャスト パケットとユニキャスト パケットが混在した伝送もサポートされます。EIGRP パケットには確実に送信する必要があるものと、そうでないものがあります。効率化のため、信頼性は必要時にのみ提供されます。たとえば、マルチキャスト機能があるマルチアクセス ネットワーク（イーサネットなど）では、すべてのネイバーにそれぞれ hello パケットを確実に送信する必要はありません。そのため、EIGRP は、1 つのマルチキャスト hello を送信し、パケットに確認応答が必要ないという通知をそのパケットに含めます。他のタイプのパケット（アップデートなど）の場合は、確認応答（ACK

パケット）を要求します。信頼性の高い伝送であれば、ペンディング中の未確認応答パケットがある場合、マルチキャストパケットを迅速に送信できます。このため、リンク速度が変化する場合でも、コンバージェンス時間を短く保つことができます。

- DUAL有限状態マシンには、すべてのルート計算の決定プロセスが組み込まれており、すべてのネイバーによってアドバタイズされたすべてのルートが追跡されます。DUALは距離情報（メトリックともいう）を使用して、効率的な、ループのないパスを選択し、さらにDUALは適切な後継ルータに基づいて、ルーティング テーブルに挿入するルートを選択します。後継ルータは、宛先への最小コスト パス（ルーティング ループに関連しないことが保証されている）を持つ、パケット転送に使用される隣接ルータです。適切な後継ルータが存在しなくても、宛先にアドバタイズするネイバーが存在する場合は再計算が行われ、この結果、新しい後継ルータが決定されます。ルートの再計算に要する時間によって、コンバージェンス時間が変わります。再計算はプロセッサに負荷がかかるため、必要でない場合は、再計算しないようにしてください。トポロジが変更されると、DUALはフィジブル サクセサの有無を調べます。適切なフィジブル サクセサが存在する場合は、それらを探して使用し、不要な再計算を回避します。
- プロトコル依存モジュールは、ネットワーク層プロトコル固有のタスクを実行します。たとえば、IP EIGRP モジュールは、IP でカプセル化された EIGRP パケットを送受信します。また、EIGRPパケットを解析したり、DUALに受信した新しい情報を通知したりします。EIGRP は DUAL にルーティング決定を行うように要求しますが、結果は IP ルーティング テーブルに格納されます。EIGRP は、他の IP ルーティング プロトコルによって取得したルートの再配信も行います。

（注） EIGRP を有効にするには、スタンドアロンスイッチまたはアクティブスイッチで Network Advantage ライセンスを実行している必要があります。

EIGRP NSF

デバイススタックは、次の 2 つのレベルの EIGRP ノンストップ フォワーディングをサポートします。

- EIGRP NSF 認識
- EIGRP NSF 対応

EIGRP NSF 認識

Network Advantage ライセンスは IPv4 の EIGRP NSF 認識をサポートしています。隣接ルータが NSF 対応である場合、レイヤ 3 デバイスでは、ルータに障害が発生してプライマリ RP がバックアップ RP によって引き継がれる間、または処理を中断させずにソフトウェアアップグレードを行うためにプライマリ RP を手動でリロードしている間、隣接ルータからパケットを転送し続けます。この機能をディセーブルにできません。

EIGRP NSF 対応

Network Advantage ライセンスでは、EIGRP Cisco NSF ルーティングがサポートされています。それにより、コンバージェンスの時間が短くなり、アクティブスイッチ変更後のトラフィック損失がなくなります。

Network Advantage ライセンスは、EIGRP NSF 対応ルーティングも IPv4 に対してサポートし、アクティブスイッチ変更後のコンバージェンス向上と、トラフィック損失低減を実現します。EIGRP NSF 対応のアクティブスイッチが再起動したとき、または新しいアクティブスイッチが起動して NSF が再起動したとき、このデバイスにはネイバーが存在せず、トポロジテーブルは空の状態です。デバイスは、デバイススタックに対するトラフィックを中断することなく、インターフェイスの起動、ネイバーの再取得、およびトポロジテーブルとルーティングテーブルの再構築を行う必要があります。EIGRP ピアルータは新しいアクティブスイッチから学習したルートを維持し、NSF の再起動処理の間トラフィックの転送を継続します。

ネイバーによる隣接リセットを防ぐために、新しいアクティブスイッチは EIGRP パケットヘッダーの新しい Restart（RS）ビットを使用して再起動を示します。これを受信したネイバーは、ピア リスト内のスタックと同期を取り、スタックとの隣接関係を維持します。続いてネイバーは、RS ビットがセットされているアクティブスイッチにトポロジテーブルを送信して、自身が NSF 認識デバイスであることおよび新しいアクティブスイッチを補助していることを示します。

スタックのピアネイバーの少なくとも 1 つが NSF 認識デバイスであれば、アクティブスイッチはアップデート情報を受信してデータベースを再構築します。各 NSF 認識ネイバーは、最後のアップデート パケットに End of Table（EOT）マーカーを付けて送信して、テーブル情報の最後であることを示します。アクティブスイッチは、EOT マーカーを受信したときにコンバージェンスを認識し、続いてアップデートの送信を始めます。アクティブスイッチがネイバーからすべての EOT マーカーを受信した場合、または NSF コンバージタイマーが期限切れになった場合、EIGRP は RIB にコンバージェンスを通知し、すべての NSF 認識ピアにトポロジテーブルをフラッディングします。

EIGRP スタブ ルーティング

EIGRP スタブルーティング機能は、ネットワークの安定性を高め、リソース利用率を抑え、スタブデバイス構成を簡素化します。

スタブルーティングは一般にハブアンドスポーク型のネットワークトポロジで使用されます。ハブアンドスポーク型ネットワークでは、1 つ以上のエンド（スタブ）ネットワークが 1 台のリモートデバイス（スポーク）に接続され、そのリモートデバイスは 1 つ以上のディストリビューションデバイス（ハブ）に接続されています。リモートデバイスは、1つ以上のディストリビューション デバイスに隣接しています。IP トラフィックがリモートデバイスに到達するための唯一のルートは、ディストリビューションデバイスを経由するものです。このタイプの設定は、一般的に、ディストリビューション デバイスが WAN に直接接続されている WAN トポロジで使用されます。ディストリビューションデバイスは、多くの場合、多数のリモートデバイスに接続できます。ハブアンドスポーク型トポロジでは、リモートデバイスがすべての非ローカルトラフィックをディストリビューションデバイスに転送する必要があります。これにより、リモートデバイスが完全なルーティングテーブルを保有する必要はなくなります。一

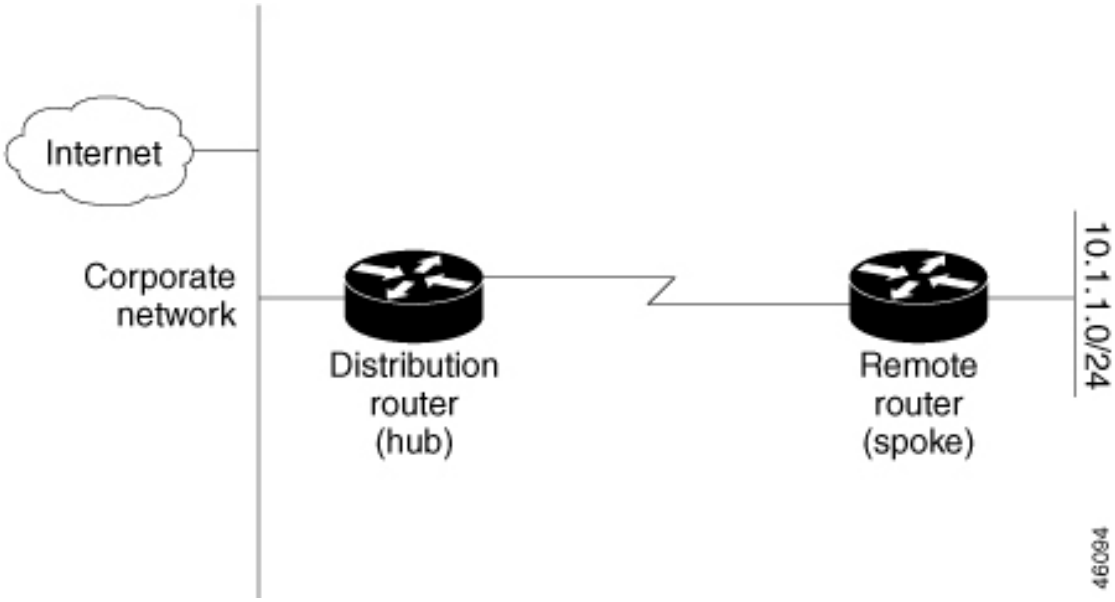
般に、ディストリビューションデバイスはデフォルトルート以外の情報をリモートデバイスに送信する必要はありません。

EIGRP スタブルーティング機能を使用する場合、EIGRP を使用するように、ディストリビューションデバイスおよびリモートデバイスを設定し、さらにリモートデバイスだけをスタブとして設定する必要があります。指定されたルートのみが、リモート（スタブ）デバイスから伝播されます。スタブデバイスは、サマリー、接続されているルート、再配布されたスタティックルート、外部ルート、および内部ルートに対するクエリーすべてに、応答として「inaccessible」というメッセージを返します。スタブとして設定されているデバイスは、特殊なピア情報パケットをすべての隣接デバイスに送信して、そのステータスをスタブデバイスとして報告します。

スタブステータスの情報を伝えるパケットを受信したネイバーはすべて、スタブデバイスにルートのクエリーを送信しなくなり、スタブピアを持つデバイスはそのピアのクエリーを送信しなくなります。スタブデバイスは、ディストリビューションデバイスを使用して適切なアップデートをすべてのピアに送信します。

次の図は、単純なハブアンドスポーク型ネットワークを示しています。

図 *5 :* 単純なハブアンドスポーク型ネットワーク



ルートがリモートデバイスにアドバタイズされることを、スタブルーティング機能自体が回避することはありません。上の例では、リモートデバイスはディストリビューションデバイスを経由してのみ企業ネットワークおよびインターネットにアクセスできます。リモートデバイスが完全なルートテーブルを保有しても機能面での意味はありません。これは、企業ネットワークとインターネットへのパスは常にディストリビューション デバイスを経由するためです。ルートテーブルが大きくなると、リモートデバイスに必要なメモリ量が減るだけです。帯域幅とメモリは、ディストリビューションデバイスのルートを集約およびフィルタリングすることによって節約できます。リモートデバイスは、宛先に関係なく、ディストリビューションデバイスにすべての非ローカルトラフィックを送信する必要があるため、他のネットワークから学習されたルートを受け取る必要がありません。真のスタブネットワークが望ましい場合は、

ディストリビューションデバイスがリモートデバイスにデフォルトルートだけを送信するように設定する必要があります。EIGRP スタブルーティング機能では、ディストリビューションデバイスでの集約を自動的に有効にしません。ほとんどの場合、ネットワーク管理者が、ディストリビューション デバイスにサマライズを設定する必要があります。

（注） ディストリビューションデバイスがリモートデバイスにデフォルトルートだけを送信するように設定する場合、リモートデバイスで **ip classless** コマンドを使用する必要があります。デフォルトでは、EIGRP スタブルーティング機能をサポートするシスコのすべてのイメージで **ip classless** コマンドが有効になっています。

EIGRP スタブルーティング機能がない場合、ディストリビューション デバイスからリモートデバイスに送信されたルートがフィルタリングまたは集約された後でも、問題が発生することがあります。企業ネットワーク内でルートが失われると、EIGRPはクエリーをディストリビューションデバイスに送信できます。ルートがサマライズされている場合でも、ディストリビューションデバイスが代わりにリモートデバイスにクエリーを送信します。ディストリビューションデバイスとリモートデバイスの間の通信（WANリンクを介した）に問題がある場合、EIGRP Stuck In Active（SIA）状態が発生し、ネットワークのどこかで不安定になる可能性があります。EIGRP スタブルーティング機能を使用することにより、ネットワーク管理者はリモートデバイスへクエリーが送信されないようにできます。

EIGRPv6 スタブ ルーティング

EIGRPv6 スタブ ルーティング機能は、エンド ユーザーの近くにルーテッド トラフィックを移動することでリソースの利用率を低減させます。

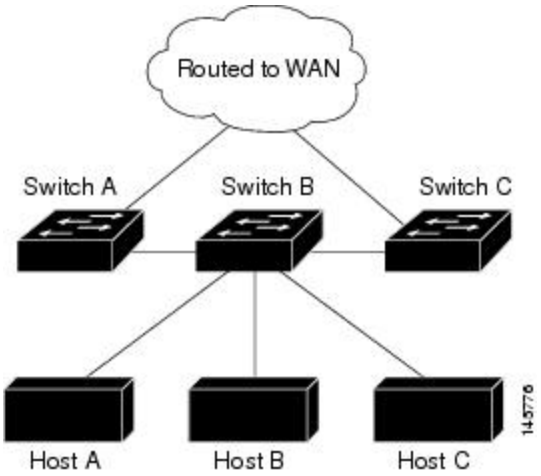
EIGRPv6 スタブ ルーティングを使用するネットワークでは、ユーザーに対する IPv6 トラフィックの唯一の許容ルートは、EIGRPv6 スタブ ルーティングを設定しているスイッチ経由のみです。スイッチは、ユーザー インターフェイスとして設定されているインターフェイスまたは他のデバイスに接続されているインターフェイスにルーテッド トラフィックを送信します。

EIGRPv6 スタブ ルーティングを使用しているときは、EIGRPv6 を使用してスイッチだけをスタブとして設定するように、ディストリビューションルータおよびリモート ルータを設定する必要があります。指定したルートだけがスイッチから伝播されます。スイッチは、サマリー、接続ルート、およびルーティング アップデートに対するすべてのクエリーに応答します。

スタブ ルータの状態を通知するパケットを受信した隣接ルータは、ルートについてはスタブ ルータに照会しません。また、スタブ ピアを持つルータは、そのピアについては照会しません。スタブ ルータは、ディストリビューション ルータを使用して適切なアップデートをすべてのピアに送信します。

次の図では、スイッチ B は EIGRPv6 スタブ ルータとして設定されています。スイッチ A および C は残りの WAN に接続されています。スイッチ B は、接続ルート、スタティック ルート、再配布ルート、およびサマリー ルートをスイッチ A と C にアドバタイズします。 スイッチ B は、スイッチ A から学習したルートをアドバタイズしません（逆の場合も同様です）。

図 *6 : EIGRP* スタブ ルータ設定



EIGRPv6 スタブ ルーティングの詳細については、『*Cisco IOS IP Configuration Guide, Volume 2 of 3: Routing Protocols, Release 12.4*』の「Implementing EIGRP for IPv6」を参照してください。

EIGRP の設定方法

EIGRP ルーティング プロセスを作成するには、EIGRP をイネーブルにし、ネットワークを関連付ける必要があります。EIGRP は指定されたネットワーク内のインターフェイスにアップデートを送信します。インターフェイス ネットワークを指定しないと、どの EIGRP アップデートでもアドバタイズされません。

（注） ネットワーク上に IGRP 用に設定されているルータがあり、この設定を EIGRP に変更する場合は、IGRP と EIGRP の両方が設定された移行ルータを指定する必要があります。この場合は、この次の項に記載されているステップ 1 ～ 3 を実行し、さらに「スプリット ホライゾンの設定」も参照してください。ルートを自動的に再配信するには、同じ AS 番号を使用する必要があります。

EIGRP のデフォルト設定

表 *23 : EIGRP* のデフォルト設定

機能	デフォルト設定
自動サマリー	ディセーブル。
デフォルト情報	再配信中は外部ルートが許可され、EIGRP プロセス間でが渡されます。

機能	デフォルト設定
デフォルト メトリック	デフォルトメトリックなしで再配信できるのは、接続された びインターフェイスのスタティック ルートだけです。デフ リックは次のとおりです。 • 帯域幅：0 以上の kb/s • 遅延（10 マイクロ秒）：0 または 39.1 ナノ秒の倍数であ の数値 • 信頼性：0 ～ 255 の任意の数値（255 の場合は信頼性が • 負荷：0 ～ 255 の数値で表される有効帯域幅（255 の場 負荷） • MTU：バイトで表されたルートの MTU サイズ（0 また の整数）
ディスタンス	内部距離：90 外部距離：170
EIGRP の隣接関係変更ログ	ディセーブル。隣接関係の変更はロギングされません。
IP 認証キーチェーン	認証なし
IP 認証モード	認証なし
IP 帯域幅比率	50%
IP hello 間隔	低速非ブロードキャスト マルチアクセス（NBMA）ネット 合：60 秒、それ以外のネットワークの場合：5 秒
IP ホールドタイム	低速NBMA ネットワークの場合：180 秒、それ以外のネット 合：15 秒
IP スプリットホライズン	イネーブル。
IP サマリー アドレス	サマリー集約アドレスは未定義
メトリック重み	tos：0、k1 および k3：1、k2、k4、および k5：0
ネットワーク	指定なし
ノンストップ フォワーディング（NSF）認識	Network Advantage ライセンスを実行するスイッチ上で IPv4 ネーブルになっています。レイヤ 3 スイッチでは、ハードウ トウェアの変更中に、隣接する NSF 対応ルータからのパケッ 続けることができます。

機能	デフォルト設定
NSF 対応	ディセーブル。 （注） デバイスは EIGRP NSF 対応ルーティングを I ポートします。
オフセットリスト	ディセーブル。
ルータ EIGRP	ディセーブル。
メトリック設定	ルート マップにはメトリック設定なし
トラフィック共有	メトリックの比率に応じて配分
バリアンス	1（等コスト ロード バランシング）

基本的な EIGRP パラメータの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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 eigrp** autonomous-system 例： `Device(config)#router eigrp 10`	EIGRP ルーティング プロセスをイネーブルにし、ルータ コンフィギュレーション モードを開始します。AS 番号によって他の EIGRP ルータへのルートを特定し、ルーティング情報をタグ付けします。
ステップ 4	**nsf** 例： `Device(config-router)#nsf`	（任意）EIGRP NSF をイネーブルにします。アクティブスイッチとそのすべてのピアでこのコマンドを入力します。
ステップ 5	**network** network-number 例： `Device(config-router)#network 192.168.0.0`	ネットワークを EIGRP ルーティング プロセスに関連付けます。EIGRP は指定されたネットワーク内のインターフェイスにアップデートを送信します。

	コマンドまたはアクション	目的
ステップ 6	**eigrp log-neighbor-changes** 例： `Device(config-router)#eigrp log-neighbor-changes`	（任意）EIGRP 隣接関係変更のロギングをイネーブルにし、ルーティング システムの安定性をモニターします。
ステップ 7	**metric weights** *tos k1 k2 k3 k4 k5* 例： `Device(config-router)#metric weights 0 2 0 2 0 0`	（任意）EIGRP メトリックを調整します。デフォルト値はほとんどのネットワークで適切に動作するよう入念に設定されていますが、調整することも可能です。 **注意** メトリックを設定する作業は複雑です。熟練したネットワーク設計者の指導がない場合は、行わないでください。
ステップ 8	**offset-list** [*access-list number* | *name*] {**in** | **out**} *offset* [*type number*] 例： `Device(config-router)#offset-list 21 out 10`	（任意）オフセットリストをルーティングメトリックに適用し、EIGRP によって取得したルートへの着信および発信メトリックを増加します。アクセスリストまたはインターフェイスを使用し、オフセット リストを制限できます。
ステップ 9	**auto-summary** 例： `Device(config-router)#auto-summary`	（任意）ネットワークレベル ルートへのサブネットルートの自動サマライズをイネーブルにします。
ステップ 10	**interface***interface-id* 例： `Device(config-router)#interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ3インターフェイスを指定します。
ステップ 11	**ip summary-address eigrp** *autonomous-system-number address mask* 例： `Device(config-if)#ip summary-address eigrp 1 192.168.0.0 255.255.0.0`	（任意）サマリー集約を設定します。
ステップ 12	**end** 例： `Device(config-if)#end`	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
ステップ 13	**show ip protocols** 例： `Device#show ip protocols`	入力を確認します。 NSF認識の場合、出力に次のように表示されます。 *** IP Routing is NSF aware *** EIGRP NSF enabled
ステップ 14	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

EIGRP インターフェイスの設定

インターフェイスごとに、他の EIGRP パラメータを任意で設定できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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	**interface***interface-id* 例： `Device(config)#interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ3インターフェイスを指定します。
ステップ 4	**ip bandwidth-percent eigrp** *percent* 例： `Device(config-if)#ip bandwidth-percent eigrp 60`	（任意）インターフェイスで EIGRP が使用できる帯域幅の割合を設定します。デフォルト値は 50% です。
ステップ 5	**ip summary-address eigrp** *autonomous-system-number address mask* 例： `Device(config-if)#ip summary-address eigrp 109 192.161.0.0 255.255.0.0`	（任意）指定されたインターフェイスのサマリー集約アドレスを設定します（auto-summary がイネーブルの場合は、通常設定する必要はありません）。

	コマンドまたはアクション	目的
ステップ 6	**ip hello-interval eigrp** *autonomous-system-number seconds* 例： `Device(config-if)#ip hello-interval eigrp 109 10`	（任意）EIGRP ルーティング プロセスの hello 時間間隔を変更します。指定できる範囲は 1 ～ 65535 秒です。低速 NBMA ネットワークの場合のデフォルト値は 60 秒、その他のすべてのネットワークでは 5 秒です。
ステップ 7	**ip hold-time eigrp** *autonomous-system-number seconds* 例： `Device(config-if)#ip hold-time eigrp 109 40`	（任意）EIGRP ルーティング プロセスのホールド時間間隔を変更します。指定できる範囲は 1 ～ 65535 秒です。低速 NBMA ネットワークの場合のデフォルト値は 180 秒、その他のすべてのネットワークでは 15 秒です。 **注意** ホールド タイムを調整する前に、シスコのテクニカルサポートにお問い合わせください。
ステップ 8	**no ip split-horizon eigrp** *autonomous-system-number* 例： `Device(config-if)#no ip split-horizon eigrp 109`	（任意）スプリット ホライズンをディセーブルにし、ルート情報が情報元インターフェイスからルータによってアドバタイズされるようにします。
ステップ 9	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 10	**show ip eigrp interface** 例： `Device#show ip eigrp interface`	EIGRP がアクティブであるインターフェイス、およびそれらのインターフェイスに関連する EIGRP の情報を表示します。
ステップ 11	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

IPv6 の EIGRP の設定

IPv6 EIGRP を実行するようにスイッチを設定する前に、 **ip routing global configuration** グローバル コンフィギュレーション コマンドを入力してルーティングを有効にし、**ipv6 unicast-routing global** グローバル コンフィギュレーション コマンドを入力して IPv6 パケットの転送を有効にし、IPv6 EIGRP を有効にするレイヤ 3 インターフェイス上で IPv6 を有効にします。

明示的なルータ ID を設定するには、**show ipv6 eigrp** コマンドを使用して設定済みのルータ ID を確認してから、**router-id** コマンドを使用します。

EIGRP IPv4 の場合と同様に、EIGRPv6 を使用して EIGRP IPv6 インターフェイスを指定し、これらのサブセットを受動インターフェイスとして選択できます。**passive-interface** コマンドを使用してインターフェイスをパッシブに設定してから、選択したインターフェイスで **no passive-interface** コマンドを使用してこれらのインターフェイスをアクティブにします。受動インターフェイスでは、EIGRP IPv6 を設定する必要がありません。

設定手順の詳細については、Cisco.com で『*Cisco IOS IPv6 Configuration Library*』の「Implementing EIGRP for IPv6」の章を参照してください。

EIGRP ルート認証の設定

EIGRP ルート認証を行うと、EIGRP ルーティング プロトコルからのルーティング アップデートに関する MD5 認証が可能になり、承認されていない送信元から無許可または問題のあるルーティング メッセージを受け取ることがなくなります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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	**interface** interface-id 例： `Device(config)#interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 4	**ip authentication mode eigrp** *autonomous-system* **md5** 例： `Device(config-if)#ip authentication mode eigrp 104 md5`	IP EIGRP パケットの MD5 認証をイネーブルにします。
ステップ 5	**ip authentication key-chain eigrp** *autonomous-system key-chain* 例：	IP EIGRP パケットの認証をイネーブルにします。

	コマンドまたはアクション	目的
	`Device(config-if)#ip authentication key-chain eigrp 105 chain1`	
ステップ 6	**exit** 例： `Device(config-if)#exit`	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 7	**key chain** *name-of-chain* 例： `Device(config)#key chain chain1`	キーチェーンを識別し、キーチェーン 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 で設定した名前を指定します。
ステップ 8	**key** *number* 例： `Device(config-keychain)#key 1`	キーチェーン コンフィギュレーション モードで、キー番号を識別します。
ステップ 9	**key-string** *text* 例： `Device(config-keychain-key)#key-string key1`	キーチェーン コンフィギュレーション モードで、キー ストリングを識別します。
ステップ 10	**accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} 例： `Device(config-keychain-key)#accept-lifetime 13:30:00 Jan 25 2011 duration 7200`	（任意）キーを受信できる期間を指定します。 *start-time* および *end-time* 構文には、*hh:mm:ss Month date year* または *hh:mm:ss date Month year* のいずれかを使用できます。デフォルトは、デフォルトの *start-time* 以降、無制限です。指定できる最初の日付は 1993 年 1 月 1 日です。デフォルトの *end-time* および **duration** は **infinite** です。
ステップ 11	**send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} 例： `Device(config-keychain-key)#send-lifetime 14:00:00 Jan 25 2011 duration 3600`	（任意）キーを送信できる期間を指定します。 *start-time* および *end-time* 構文には、*hh:mm:ss Month date year* または *hh:mm:ss date Month year* のいずれかを使用できます。デフォルトは、デフォルトの *start-time* 以降、無制限です。指定できる最初の日付は 1993 年 1 月 1 日です。デフォルトの *end-time* および **duration** は **infinite** です。
ステップ 12	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
ステップ 13	**show key chain** 例： `Device#show key chain`	認証キーの情報を表示します。
ステップ 14	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

EIGRP のモニタリングおよびメンテナンス

ネイバー テーブルからネイバーを削除できます。さらに、各種 EIGRP ルーティング統計情報を表示することもできます。下の図に、ネイバーを削除し、統計情報を表示する特権 EXEC コマンドを示します。

表 *24 : IP EIGRP* の *clear* および *show* コマンド

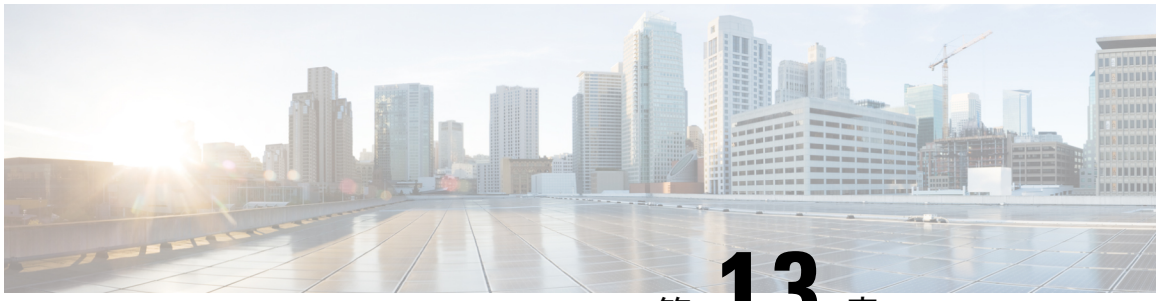
コマンド	目的
clear ip eigrp neighbors [*if-address* | *interface*]	ネイバー テーブルからネイ
show ip eigrp interface [*interface*] [*as number*]	EIGRP に設定されているイ
show ip eigrp neighbors [*type-number*]	EIGRP によって検出された
show ip eigrp topology [*autonomous-system-number*] | [[*ip-address*] *mask*]]	指定されたプロセスの EIG
show ip eigrp traffic [*autonomous-system-number*]	すべてまたは指定された E ます。

EIGRP の機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。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 *25 : EIGRP* 機能の機能情報

リリース	機能情報
Cisco IOS XE Everest 16.5.1a	この機能が導入されました。



第 13 章

BGP の設定

BGP の制約事項

グレースフルリスタートが無効になっている場合でも、BGP ホールド時間は常にデバイスのグレースフルリスタートのホールド時間よりも長く設定する必要があります。ホールド時間がサポートされていないピアデバイスでは、オープンメッセージを介してデバイスとのセッションを確立できますが、グレースフルリスタートが有効になっていると、セッションはフラッピングします。

BGP に関する情報

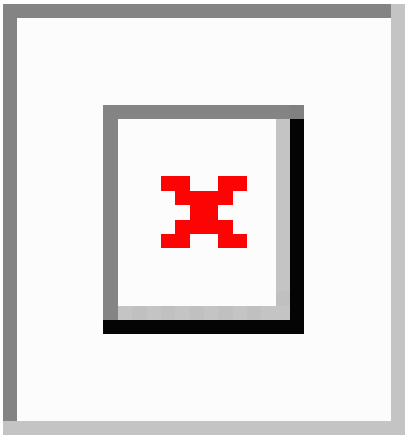
ボーダー ゲートウェイ プロトコル（BGP）は、Exterior Gateway Protocol です。自律システム間で、ループの発生しないルーティング情報交換を保証するドメイン間ルーティング システムを設定するために使用されます。自律システムは、同じ管理下で動作して RIP や OSPF などの Interior Gateway Protocol（IGP）を境界内で実行し、Exterior Gateway Protocol（EGP）を使用して相互接続されるルータで構成されます。BGP バージョン 4 は、インターネット内でドメイン間ルーティングを行うための標準 EGP です。このプロトコルは、RFC 1163、1267、および 1771 で定義されています。

BGP ネットワーク トポロジ

同じ自律システム（AS）に属し、BGP アップデートを交換するルータは内部BGP（IBGP）を実行し、異なる自律システムに属し、BGP アップデートを交換するルータは外部 BGP（EBGP）を実行します。大部分のコンフィギュレーション コマンドは、EBGP と IBGP で同じですが、

ルーティング アップデートが自律システム間で交換されるか（EBGP）、または AS 内で交換されるか（IBGP）という点で異なります。下の図に、EBGP と IBGP の両方を実行しているネットワークを示します。

図 *7: EBGP*、*IBGP*、および複数の自律システム



外部 AS と情報を交換する前に、BGP は AS 内のルータ間で内部 BGP ピアリングを定義し、IGRP や OSPF など AS 内で稼働する IGP に BGP ルーティング情報を再配布して、AS 内のネットワークに到達することを確認します。

BGP ルーティング プロセスを実行するルータは、通常 BGP スピーカーと呼ばれます。BGP はトランスポート プロトコルとして伝送制御プロトコル（TCP）を使用します（特にポート 179）。ルーティング情報を交換するため相互に TCP 接続された 2 つの BGP スピーカーを、ピアまたはネイバーと呼びます。上の図では、ルータ A と B が BGP ピアで、ルータ B と C、ルータ C と D も同様です。ルーティング情報は、宛先ネットワークへの完全パスを示す一連の AS 番号です。BGP はこの情報を使用し、ループのない自律システム マップを作成します。

このネットワークの特徴は次のとおりです。

- ルータ A および B では EBGP が、ルータ B および C では IBGP が稼働しています。EBGP ピアは直接接続されていますが、IBGP ピアは直接接続されていないことに注意してください。IGP が稼働し、2 つのネイバーが相互に到達するかぎり、IBGP ピアを直接接続する必要はありません。
- AS 内のすべての BGP スピーカーは、相互にピア関係を確立する必要があります。つまり、AS 内の BGP スピーカーは、論理的な完全メッシュ型に接続する必要があります。BGP4 は、論理的な完全メッシュに関する要求を軽減する 2 つの技術（連合およびルート リフレクタ）を提供します。
- AS 200 は AS 100 および AS 300 の中継 AS です。つまり、AS 200 は AS 100 と AS 300 間でパケットを転送するために使用されます。

BGP ピアは完全な BGP ルーティング テーブルを最初に交換し、差分更新だけを送信します。BGP ピアはキープアライブ メッセージ（接続が有効であることを確認）、および通知メッセージ（エラーまたは特殊条件に応答）を交換することもできます。

BGPの場合、各ルートはネットワーク番号、情報が通過した自律システムのリスト（自律システム パス）、および他のパス属性リストで構成されます。BGP システムの主な機能は、AS パスのリストに関する情報など、ネットワークの到達可能性情報を他の BGP システムと交換することです。この情報は、AS が接続されているかどうかを判別したり、ルーティング ループをプルーニングしたり、AS レベル ポリシー判断を行うために使用できます。

Cisco IOS が稼働しているルータやデバイスが IBGP ルートを選択または使用するのは、ネクストホップルータで使用可能なルートがあり、IGP から同期信号を受信している（IGP 同期が無効の場合は除く）場合です。複数のルートが使用可能な場合、BGP は属性値に基づいてパスを選択します。BGP 属性については、「BGP 判断属性の設定」の項を参照してください。

BGP バージョン 4 ではクラスレス ドメイン間ルーティング（CIDR）がサポートされているため、集約ルートを作成してスーパーネットを構築し、ルーティング テーブルのサイズを削減できます。CIDR は、BGP 内部のネットワーク クラスの概念をエミュレートし、IP プレフィックスのアドバタイズをサポートします。

NSF 認識

BGP NSF 認識機能は、Network Advantage ライセンスで IPv4 に対してサポートされます。。BGP ルーティングでこの機能を有効にするには、グレースフル リスタートを有効にする必要があります。隣接ルータが NSF 対応で、この機能が有効になっている場合、レイヤ 3 デバイスは、ルータに障害が発生してプライマリルートプロセッサ（RP）がバックアップ RP によって引き継がれる間、または無停止ソフトウェアアップグレードを行うためにプライマリ RP を手動でリロードしている間、隣接ルータからパケットを転送し続けます。

BGP ルーティングに関する情報

BGP ルーティングを有効にするには、BGP ルーティング プロセスを確立し、ローカル ネットワークを定義します。BGP はネイバーとの関係を完全に認識する必要があるため、BGP ネイバーも指定する必要があります。

BGP は、内部および外部の 2 種類のネイバーをサポートします。内部ネイバーは同じ AS 内に、外部ネイバーは異なる AS 内にあります。通常の場合、外部ネイバーは相互に隣接し、1 つのサブネットを共有しますが、内部ネイバーは同じ AS 内の任意の場所に存在します。

スイッチではプライベート AS 番号を使用できます。プライベート AS 番号は通常サービス プロバイダによって割り当てられ、ルートが外部ネイバーにアドバタイズされないシステムに設定されます。プライベート AS 番号の範囲は 64512 ～ 65535 です。AS パスからプライベート AS 番号を削除するように外部ネイバーを設定するには、**neighbor remove-private-as** ルータ コンフィギュレーション コマンドを使用します。この結果、外部ネイバーにアップデートを渡すとき、AS パス内にプライベート AS 番号が含まれている場合は、これらの番号が削除されます。

AS が別の AS からさらに別の AS にトラフィックを渡す場合は、アドバタイズ対象のルートに矛盾が存在しないことが重要です。BGP がルートをアドバタイズしてから、ネットワーク内のすべてのルータが IGP を通してルートを学習した場合、AS は一部のルータがルーティングできなかったトラフィックを受信することがあります。このような事態を避けるため、BGP は

IGP が AS に情報を伝播し、BGP が IGP と同期化されるまで、待機する必要があります。同期化は、デフォルトで有効に設定されています。AS が特定の AS から別の AS にトラフィックを渡さない場合、または自律システム内のすべてのルータで BGP が稼働している場合は、同期化を無効にし、IGP内で伝送されるルート数を少なくして、BGPがより短時間で収束するようにします。

ルーティング ポリシーの変更

ピアのルーティング ポリシーには、インバウンドまたはアウトバウンド ルーティング テーブル アップデートに影響する可能性があるすべての設定が含まれます。BGP ネイバーとして定義された 2 台のルータは、BGP 接続を形成し、ルーティング情報を交換します。このあとでBGP フィルタ、重み、距離、バージョン、またはタイマーを変更する場合、または同様の設定変更を行う場合は、BGPセッションをリセットし、設定の変更を有効にする必要があります。

リセットには、ハード リセットとソフト リセットの 2 種類があります。Cisco IOS Release 12.1 以降では、事前に設定を行わなくても、ソフト リセットを使用できます。事前設定なしにソフト リセットを使用するには、両方の BGP ピアでソフト ルート リフレッシュ機能がサポートされていなければなりません。この機能は、ピアによって TCP セッションが確立されたときに送信される OPEN メッセージに格納されてアドバタイズされます。ソフト リセットを使用すると、BGPルータ間でルート リフレッシュ要求およびルーティング情報を動的に交換したり、それぞれのアウトバウンド ルーティング テーブルをあとで再アドバタイズできます。

- ソフト リセットによってネイバーからインバウンド アップデートが生成された場合、このリセットはダイナミック インバウンド ソフト リセットといいます。
- ソフト リセットによってネイバーに一連のアップデートが送信された場合、このリセットはアウトバウンド ソフト リセットといいます。

ソフト インバウンド リセットが発生すると、新規インバウンド ポリシーが有効になります。ソフトアウトバウンドリセットが発生すると、BGPセッションがリセットされずに、新規ローカル アウトバウンド ポリシーが有効になります。アウトバウンド ポリシーのリセット中に新しい一連のアップデートが送信されると、新規インバウンドポリシーも有効になる場合があります。

下の表に、ハード リセットとソフト リセットの利点および欠点を示します。

表 *26 :* ハード リセットとソフト リセットの利点および欠点

リセット タイプ	利点	欠点
ハード リセット	メモリ オーバーヘッドが発生しません。	ネイバーから提供された BGP、FIB テーブルのプレフィックス す。非推奨
発信ソフト リセット	ルーティング テーブル アップデートが設定、保管されません。	インバウンド ルーティング テーデートがリセットされない。

リセット タイプ	利点	欠点
ダイナミック インバウンド ソフト リセット	BGPセッションおよびキャッシュがクリアされません。 ルーティング テーブル アップデートを保管する必要がなく、メモリ オーバーヘッドが発生しません。	両方のBGPルータでルート能をサポートする必要があ IOS Release 12.1 以降）。

BGP 判断属性

BGPスピーカーが複数の自律システムから受信したアップデートが、同じ宛先に対して異なるパスを示している場合、BGPスピーカーはその宛先に到達する最適パスを1つ選択する必要があります。選択されたパスはBGP ルーティング テーブルに格納され、ネイバーに伝播されます。この判断は、アップデートに格納されている属性値、およびBGP で設定可能な他の要因に基づいて行われます。

BGP ピアはネイバー AS からプレフィックスに対する 2 つの EBGP パスを学習するとき、最適パスを選択して IP ルーティング テーブルに挿入します。BGP マルチパス サポートが有効で、同じネイバー自律システムから複数の EBGP パスを学習する場合、単一の最適パスの代わりに、複数のパスが IP ルーティング テーブルに格納されます。そのあと、パケット スイッチング中に、複数のパス間でパケット単位または宛先単位のロードバランシングが実行されます。**maximum-paths** ルータ コンフィギュレーション コマンドは、許可されるパス数を制御します。

これらの要因により、BGP が最適パスを選択するために属性を評価する順序が決まります。

1. パスで指定されているネクストホップが到達不能な場合、このアップデートは削除されます。BGPネクストホップ属性（ソフトウェアによって自動判別される）は、宛先に到達するために使用されるネクストホップのIPアドレスです。EBGPの場合、通常このアドレスは **neighbor remote-as router** ルータ コンフィギュレーション コマンドで指定されたネイバーのIPアドレスです。ネクストホップの処理を無効にするには、ルートマップまたは **neighbor next-hop-self** ルータ コンフィギュレーション コマンドを使用します。
2. 最大の重みのパスを推奨します（シスコ独自のパラメータ）。ウェイト属性はルータにローカルであるため、ルーティングアップデートで伝播されません。デフォルトでは、ルータ送信元のパスに関するウェイト属性は 32768 で、それ以外のパスのウェイト属性は 0 です。最大の重みのルートを推奨します。重みを設定するには、アクセスリスト、ルートマップ、または **neighbor weight** ルータ コンフィギュレーション コマンドを使用します。
3. ローカル プリファレンス値が最大のルートを推奨します。ローカル プリファレンスはルーティング アップデートに含まれ、同じ AS 内のルータ間で交換されます。ローカル初期設定属性のデフォルト値は100です。ローカルプリファレンスを設定するには、**bgp default local-preference** ルータ コンフィギュレーション コマンドまたはルートマップを使用します。
4. ローカル ルータ上で稼働する BGP から送信されたルートを推奨します。

5. AS パスが最短のルートを推奨します。
6. 送信元タイプが最小のルートを推奨します。内部ルートまたは IGP は、EGP によって学習されたルートよりも小さく、EGP で学習されたルートは、未知の送信元のルートまたは別の方法で学習されたルートよりも小さくなります。
7. 想定されるすべてのルートについてネイバー AS が同じである場合は、MED メトリック属性が最小のルートを推奨します。MED を設定するには、ルートマップまたは **default-metric** ルータ コンフィギュレーション コマンドを使用します。IBGP ピアに送信されるアップデートには、MED が含まれます。
8. 内部（IBGP）パスより、外部（EBGP）パスを推奨します。
9. 最も近い IGP ネイバー（最小の IGP メトリック）を通って到達できるルートを推奨します。ルータは、AS 内の最短の内部パス（BGP のネクストホップへの最短パス）を使用し、宛先に到達するためです。
10. 次の条件にすべて該当する場合は、このパスのルートを IP ルーティング テーブルに挿入してください。
    - 最適ルートと目的のルートがともに外部ルートである
    - 最適ルートと目的のルートの両方が、同じネイバー自律システムからのルートである
    - maximum-paths が有効である
11. マルチパスが有効でない場合は、BGP ルータ ID の IP アドレスが最小であるルートを推奨します。通常、ルータ ID はルータ上の最大の IP アドレスまたはループバック（仮想）アドレスですが、実装に依存することがあります。

ルート マップ

BGP 内でルート マップを使用すると、ルーティング情報を制御、変更したり、ルーティング ドメイン間でルートを再配布する条件を定義できます。各ルート マップには、ルート マップを識別する名前（マップ タグ）およびオプションのシーケンス番号が付いています。

BGP フィルタリング

BGP アドバタイズメントをフィルタリングするには、**as-path access-list** グローバル コンフィギュレーション コマンドや **neighbor filter-list** ルータ コンフィギュレーション コマンドなどの AS パスフィルタを使用します。**neighbor distribute-list** ルータ コンフィギュレーション コマンドとアクセスリストを併用することもできます。distribute-list フィルタはネットワーク番号に適用されます。**distribute-list** コマンドの詳細については、「ルーティングアップデートのアドバタイズおよび処理の制御」の項を参照してください。

ネイバー単位でルートマップを使用すると、アップデートをフィルタリングしたり、さまざまな属性を変更したりできます。ルート マップは、インバウンド アップデートまたはアウトバ

ウンド アップデートのいずれかに適用できます。ルート マップを渡すルートだけが、アップデート内で送信または許可されます。着信および発信の両方のアップデートで、AS パス、コミュニティ、およびネットワーク番号に基づくマッチングがサポートされています。AS パスのマッチングには **match as-path access-list** ルートマップコマンド、コミュニティに基づくマッチングには **match community-list** ルートマップコマンド、ネットワークに基づくマッチングには **ip access-list** グローバル コンフィギュレーション コマンドが必要です。

BGP フィルタリングのプレフィックス リスト

neighbor distribute-list ルータ コンフィギュレーション コマンドを含む多数の BGP ルート フィルタリング コマンドでは、アクセスリストの代わりにプレフィックスリストを使用できます。プレフィックスリストを使用すると、大規模リストのロードおよび検索パフォーマンスが改善し、差分更新がサポートされ、コマンドライン インターフェイス（CLI）設定が簡素化され、柔軟性が増すなどの利点が生じます。

プレフィックス リストによるフィルタリングでは、アクセス リストの照合の場合と同様に、プレフィックス リストに記載されたプレフィックスとルートのプレフィックスが照合されます。一致すると、一致したルートが使用されます。プレフィックスが許可されるか、または拒否されるかは、次に示すルールに基づいて決定されます。

- 空のプレフィックス リストはすべてのプレフィックスを許可します。
- 特定のプレフィックスがプレフィックス リストのどのエントリとも一致しなかった場合、実質的に拒否されたものと見なされます。
- 指定されたプレフィックスと一致するエントリがプレフィックス リスト内に複数存在する場合は、シーケンス番号が最小であるプレフィックス リスト エントリが識別されます。

デフォルトでは、シーケンス番号は自動生成され、5 ずつ増分します。シーケンス番号の自動生成を無効にした場合は、エントリごとにシーケンス番号を指定する必要があります。シーケンス番号を指定する場合の増分値に制限はありません。増分値が 1 の場合は、このリストに追加エントリを挿入できません。増分値が大きい場合は、値がなくなることがあります。

BGP コミュニティ フィルタリング

BGP コミュニティ フィルタリングは、COMMUNITIES 属性の値に基づいてルーティング情報の配信を制御する BGP の方法の 1 つです。この属性によって、宛先はコミュニティにグループ化され、コミュニティに基づいてルーティング判断が適用されます。この方法を使用すると、ルーティング情報の配信制御を目的とする BGP スピーカーの設定が簡単になります。

コミュニティは、共通するいくつかの属性を共有する宛先のグループです。各宛先は複数のコミュニティに属します。AS 管理者は、宛先が属するコミュニティを定義できます。デフォルトでは、すべての宛先が一般的なインターネット コミュニティに属します。コミュニティは、過渡的でグローバルなオプションの属性である、COMMUNITIES 属性（1 ～ 4294967200 の数値）によって識別されます。事前に定義された既知のコミュニティの一部を、次に示します。

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

- **no-export**：EBGP ピアにこのルートをアドバタイズしません。
- **no-advertise**：どのピア（内部または外部）にもこのルートをアドバタイズしません。
- **local-as**：ローカルな AS 外部のピアにこのルートをアドバタイズしません。

コミュニティに基づき、他のネイバーに許可、送信、配信するルーティング情報を制御できます。BGP スピーカーは、ルートを学習、アドバタイズ、または再配布するときに、ルートのコミュニティを設定、追加、または変更します。ルートを集約すると、作成された集約内の COMMUNITIES 属性に、すべての初期ルートの全コミュニティが含まれます。

コミュニティ リストを使用すると、ルート マップの match 句で使用されるコミュニティ グループを作成できます。さらに、アクセス リストの場合と同様、一連のコミュニティ リストを作成することもできます。ステートメントは一致が見つかるまでチェックされ、1 つのステートメントが満たされると、テストは終了します。

BGP ネイバーおよびピア グループ

通常、BGP ネイバーの多くは同じアップデート ポリシー（同じアウトバウンド ルート マップ、配信リスト、フィルタ リスト、アップデート送信元など）を使用して設定されます。アップデート ポリシーが同じネイバーをピア グループにまとめると設定が簡単になり、アップデートの効率が高まります。多数のピアを設定した場合は、この方法を推奨します。

BGP ピア グループを設定するには、ピア グループを作成し、そこにオプションを割り当てて、ピア グループ メンバーとしてネイバーを追加します。ピア グループを設定するには、**neighbor** ルータ コンフィギュレーション コマンドを使用します。デフォルトでは、ピア グループ メンバーは remote-as（設定されている場合）、version、update-source、out-route-map、out-filter-list、out-dist-list、minimum-advertisement-interval、next-hop-self など、ピア グループの設定オプションをすべて継承します。すべてのピア グループ メンバーは、ピア グループに対する変更を継承します。また、アウトバウンド アップデートに影響しないオプションを無効にするように、メンバーを設定することもできます。

集約ルート

クラスレス ドメイン間ルーティング（CIDR）を使用すると、集約ルート（またはスーパーネット）を作成して、ルーティング テーブルのサイズを最小化できます。BGP 内に集約ルートを設定するには、集約ルートを BGP に再配布するか、または BGP ルーティング テーブル内に集約エントリを作成します。BGP テーブル内に特定のエントリがさらに 1 つまたは複数存在する場合は、BGP テーブルに集約アドレスが追加されます。

ルーティング ドメイン コンフェデレーション

IBGP メッシュを削減する方法の 1 つは、自律システムを複数のサブ自律システムに分割して、単一の自律システムとして認識される単一の連合にグループ化することです。各自律システムは内部で完全にメッシュ化されていて、同じコンフェデレーション内の他の自律システムとの間には数本の接続があります。異なる自律システム内にあるピアでは EBGP セッションが使用

されますが、ルーティング情報は IBGP ピアと同様な方法で交換されます。具体的には、ネクスト ホップ、MED、およびローカル プリファレンス情報は維持されます。すべての自律システムで単一の IGP を使用できます。

BGP ルート リフレクタ

BGP では、すべての IBGP スピーカーを完全メッシュ構造にする必要があります。外部ネイバーからルートを受信したルータは、そのルートをすべての内部ネイバーにアドバタイズする必要があります。ルーティング情報のループを防ぐには、すべての IBGP スピーカーを接続する必要があります。内部ネイバーは、内部ネイバーから学習されたルートを他の内部ネイバーに送信しません。

ルートリフレクタを使用すると、学習されたルートをネイバーに渡す場合に他の方法が使用されるため、すべての IBGP スピーカーを完全メッシュ構造にする必要はありません。IBGP ピアをルート リフレクタに設定すると、その IBGP ピアは IBGP によって学習されたルートを一連の IBGP ネイバーに送信するようになります。ルート リフレクタの内部ピアには、クライアント ピアと非クライアント ピア（AS 内の他のすべてのルータ）の 2 つのグループがあります。ルート リフレクタは、これらの 2 つのグループ間でルートを反映させます。ルート リフレクタおよびクライアント ピアは、クラスタを形成します。非クライアント ピアは相互に完全メッシュ構造にする必要がありますが、クライアントピアはその必要はありません。クラスタ内のクライアントは、そのクラスタ外の IBGP スピーカーと通信しません。

アドバタイズされたルートを受信したルートリフレクタは、ネイバーに応じて、次のいずれかのアクションを実行します。

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

通常、クライアントのクラスタにはルート リフレクタが 1 つあり、クラスタはルート リフレクタのルータ ID で識別されます。冗長性を高めて、シングル ポイントでの障害を回避するには、クラスタに複数のルートリフレクタを設定する必要があります。このように設定した場合は、ルート リフレクタが同じクラスタ内のルート リフレクタからのアップデートを認識できるように、クラスタ内のすべてのルート リフレクタに同じクラスタ ID（4 バイト）を設定する必要があります。クラスタを処理するすべてのルートリフレクタは完全メッシュ構造にし、一連の同一なクライアント ピアおよび非クライアント ピアを設定する必要があります。

ルート ダンプニング

ルート フラップ ダンプニングは、インターネットワーク内でフラッピング ルートの伝播を最小化するための BGP 機能です。ルートの状態が使用可能、使用不可能、使用可能、使用不可能という具合に、繰り返し変化する場合、ルートはフラッピングと見なされます。ルートダンプニングが有効の場合は、フラッピングしているルートにペナルティ値が割り当てられます。

ルートの累積ペナルティが、設定された制限値に到達すると、ルートが稼働している場合であっても、BGPはルートのアドバタイズメントを抑制します。再使用限度は、ペナルティと比較される設定可能な値です。ペナルティが再使用限度より小さくなると、起動中の抑制されたルートのアドバタイズメントが再開されます。

IBGPによって取得されたルートには、ダンプニングが適用されません。このポリシーにより、IBGP ピアのペナルティが AS 外部のルートよりも大きくなることはありません。

条件付き BGP ルートの注入

BGP を通じてアドバタイズされるルートは、通常、使用されるルートの数が最小化され、グローバル ルーティング テーブルのサイズが小さくなるように集約されます。しかし、共通のルート集約では、より具体的なルーティング情報（より正確であるが、パケットを宛先に転送するために必要なわけではない）がわかりにくくなってしまいます。ルーティングの精度は、共通のルート集約により低下します。これは、トポロジ的に大きな領域に広がる複数のアドレスやホストを表すプレフィックスを 1 つのルートに正確に反映させることはできないからです。シスコ ソフトウェアには、プレフィックスを BGP 由来とする方法がいくつか用意されています。BGP 条件付きルート注入機能の導入以前は、既存の方法として、再配布や**network** または **aggregate-address** コマンドが使用されていました。ただし、これらの方法は、より具体的なルーティング情報（開始されるルートと一致するもの）がルーティング テーブルまたは BGP テーブルのいずれかに存在することを前提にしています。

BGP の条件付きルートの注入により、一致するものがなくても、プレフィックスを BGP ルーティング テーブルにすることができます。この機能を使って、管理ポリシーやトラフィック エンジニアリング情報に基づいて、より具体的なルートを生成することができます。これにより、設定された条件が満たされた場合にだけ BGP ルーティング テーブルに注入される、より具体的なルートへのパケットの転送をさらに厳密に制御できるようになります。この機能を有効にすると、条件に応じて、あまり具体的ではないプレフィックスにより具体的なプレフィックスを注入または置き換えることにより、共通のルート集約の精度を高めることができるようになります。元のプレフィックスと同じ、またはより具体的なプレフィックスだけが注入されます。BGP 条件付きルート注入を有効にするには、**bgp inject-map exist-map** コマンドを使用します。また、BGP 条件付きルート注入では、2 つのルート マップ（注入マップと存在マップ）を使用して、1つ（または複数）のより具体的なプレフィックスがBGPルーティングテーブルに注入されます。存在マップは、BGP スピーカーが追跡するプレフィックスを指定します。注入マップは、ローカル BGP テーブルで作成され、このテーブルにインストールされるプレフィックスを定義します。

（注） 注入マップおよび存在マップで一致となるプレフィックスはルート マップ句ごとに1つだけです。さらにプレフィックスを注入するには、ルート マップ句を追加で設定する必要があります。複数のプレフィックスが使用されている場合は、一致する最初のプレフィックスが使用されます。

BGP Peer テンプレート

構成管理など、ピア グループの制約の一部に対応するため、BGP アップデート グループ コンフィギュレーションをサポートする BGP ピア テンプレートが導入されました。

ピア テンプレートは、ポリシーを共有するネイバーに適用可能なコンフィギュレーション パターンです。ピア テンプレートは再利用が可能で、継承がサポートされているため、ネットワーク オペレータはピア テンプレートを使用して、ポリシーを共有している BGP ネイバーに対して異なるネイバー コンフィギュレーションをグループ化し適用できます。また、ネットワーク オペレータは、別のピア テンプレートからコンフィギュレーションを継承できるというピア テンプレートの機能を使用して、非常に複雑なコンフィギュレーション パターンを定義できるようになります。

ピア テンプレートには 2 種類あります。

- ピア セッション テンプレート。アドレス ファミリ モードおよび NLRI コンフィギュレーション モードすべてに共通する一般的なセッション コマンドのコンフィギュレーションをグループ化し、適用するために使用されます。
- ピア ポリシー テンプレート。特定のアドレス ファミリおよびNLRI コンフィギュレーション モードで適用されるコマンドのコンフィギュレーションをグループ化し、適用するために使用されます。

ピア テンプレートにより、柔軟性が高まり、ネイバー コンフィギュレーションの機能が強化されます。また、ピア テンプレートはピア グループ コンフィギュレーションに代わるものを提供し、ピア グループの制約の一部を解決します。ピアテンプレートを使用した BGP ピアデバイスも、自動アップデート グループ コンフィギュレーションの恩恵を受けています。BGP ピア テンプレートが設定され、BGP ダイナミック アップデート ピア グループがサポートされたことにより、ネットワーク オペレータは BGP でピア グループを設定する必要がなくなります。また、ネットワークはコンフィギュレーションの柔軟性が高まり、コンバージェンスが高速化されたことによる恩恵を受けます。

（注） BGP ネイバーを、ピア グループとピア テンプレートの両方と連動するようには設定できません。BGP ネイバーは、1 つのピア グループだけに属するように設定するか、またはピア テンプレートからポリシーを継承するように設定します。

ピア ポリシー テンプレートには、次の制約事項が適用されます。

- ピア ポリシー テンプレートは、直接的、または間接的に、最高 8 個のピア ポリシー テンプレートを継承できます。
- BGP ネイバーを、ピア グループとピア テンプレートの両方と連動するようには設定できません。BGP ネイバーは、1 つのピア グループだけに属するように設定するか、またはピア テンプレートだけからポリシーを継承するように設定できます。

ピア テンプレートでの継承

継承機能は、ピア テンプレート操作の重要なコンポーネントです。ピア テンプレートでの継承は、たとえば、ファイルとディレクトリ ツリーなど、一般的なコンピューティングで見られるノードとツリーの構造に似ています。ピア テンプレートは、別のピア テンプレートから直接、または間接的にコンフィギュレーションを継承することができます。直接継承されたピアテンプレートは、構造体のツリーを表します。間接継承されたピア テンプレートはツリーのノードを表します。個々のノードもまた継承をサポートしているため、ブランチを作成して、そこから直接継承されたピアテンプレートすなわちツリーの起点へ連なる全ての間接継承されたピアテンプレートの設定を適用することができます。

この構造により、ネイバーのグループに通常、再適用されるコンフィギュレーション文を繰り返す必要がなくなります。これは、共通のコンフィギュレーション文を一度適用しておくと、その後は共通のコンフィギュレーションを持つネイバー グループに適用されるピア グループにより間接継承されるからです。ノードとツリー内部の別々の箇所で重複するコンフィギュレーション文は、ツリーの起点で直接継承したテンプレートによりフィルタ処理されます。直接継承されたテンプレートは、重複する間接継承された文を直接継承された文で上書きします。

継承によりネイバー コンフィギュレーションのスケーラビリティと柔軟性がさらに広がり、複数のピアテンプレート コンフィギュレーションを連ねることで、共通のコンフィギュレーション文を継承する単純なコンフィギュレーションを作成したり、共通に継承されるコンフィギュレーションとともに非常に限定的なコンフィギュレーション文を適用する複雑なコンフィギュレーションを作成したりできるようになります。ピア セッション テンプレートおよびピア ポリシーテンプレートでの継承の設定についての詳細は、これ以降のセクションで説明します。

BGP ネイバーが継承したピア テンプレートを使用する場合、特定のテンプレートに関連付けられているポリシーを判断するのが難しいことがあります。**show ip bgptemplate peer-policy** コマンドに、特定のテンプレートに関連付けられているローカルポリシーおよび継承されたポリシーの詳しいコンフィギュレーションを表示するためのキーワード **detail** が追加されました。

ピア セッション テンプレート

ピア セッション テンプレートは、一般的なセッション コマンドのコンフィギュレーションをグループ化し、セッション コンフィギュレーション要素を共有するネイバーのグループに適用するために使用されます。異なるアドレス ファミリで設定されているネイバーに共通する一般的なセッション コマンドは、同じピア セッション テンプレートに設定できます。ピア セッション テンプレートの作成と設定は、ピア セッション コンフィギュレーション モードで行います。ピア セッション テンプレートで設定できるのは、一般的なセッション コマンドだけです。次の一般的なセッション コマンドは、ピア セッション テンプレートでサポートされています。

- **description**
- **disable-connected-check**
- **ebgp-multihop**
- **exit peer-session**

- **inherit peer-session**
- **local-as**
- **password**
- **remote-as**
- **shutdown**
- **timers**
- **translate-update**
- **update-source**
- **version**

一般的なセッション コマンドをピア セッションで一度設定しておくと、ピア セッション テンプレートの直接適用、またはピア セッション テンプレートの間接継承によって、多数のネイバーに適用できます。ピア セッション テンプレートのコンフィギュレーションにより、自律システム内のすべてのネイバーに共通に適用される一般的なセッション コマンドのコンフィギュレーションが簡素化されます。

ピア セッション テンプレートは、直接継承と間接継承をサポートします。一度にピアの設定に使用できるピア セッション テンプレートは 1 つだけです。また、このピア セッション テンプレートは、間接継承されたピア セッション テンプレートを 1 つだけ含むことができます。

（注） 1 つのピア セッション テンプレートを使って、複数の継承文を設定しようとすると、エラーメッセージが表示されます。

この動作により、BGP ネイバーは 1 つのセッション テンプレートだけを直接継承し、最高 7 個のピア セッション テンプレートを間接継承できます。したがって、1 つのネイバーに最高 8 個のピア セッション コンフィギュレーション（直接継承されたピア セッション テンプレートのコンフィギュレーションと最高 7 個の間接継承されたピア セッション テンプレートのコンフィギュレーション）を適用できます。継承されたピア セッション コンフィギュレーションは、ブランチの最後のノードが最初に評価されて適用され、ツリーの起点で直接適用されたピア セッション テンプレートが最後に適用されます。直接適用されたピア セッション テンプレートは、継承されたピア セッション テンプレート コンフィギュレーションよりも優先されます。継承されたピア セッション テンプレートで重複するコンフィギュレーション文はすべて、直接適用されたピア セッション テンプレートにより上書きされます。したがって、基本セッション コマンドが異なる値で再び適用される場合は、後の値が優先され、間接継承されたテンプレートに設定されていた前の値は上書きされます。次に、この機能を使用した例を示します。

次の例では、一般セッション コマンド **remote-as 1** がピア セッション テンプレート SESSION-TEMPLATE-ONE に適用されます。

```
template peer-session SESSION-TEMPLATE-ONE
```

```
 remote-as 1
 exit peer-session
```

ピア セッション テンプレートは、一般的なセッション コマンドだけをサポートします。特定のアドレス ファミリ、または NLRI コンフィギュレーション モードだけのために設定される BGP ポリシー コンフィギュレーション コマンドは、ピア ポリシー テンプレートで設定されます。

ピア ポリシー テンプレート

ピア ポリシー テンプレートは、特定のアドレス ファミリおよび NLRI コンフィギュレーション モードで適用されるコマンドのコンフィギュレーションをグループ化し、適用するために使用されます。ピア ポリシー テンプレートの作成と設定は、ピア ポリシー コンフィギュレーション モードで行います。特定のアドレス ファミリ専用に設定される BGP ポリシー コマンドは、ピア ポリシー テンプレートで設定されます。ピア ポリシー テンプレートでは、次の BGP ポリシー コマンドがサポートされています。

- **advertisement-interval**
- **allowas-in**
- **as-override**
- **capability**
- **default-originate**
- **distribute-list**
- **dmzlink-bw**
- **exit-peer-policy**
- **filter-list**
- **inherit peer-policy**
- **maximum-prefix**
- **next-hop-self**
- **next-hop-unchanged**
- **prefix-list**
- **remove-private-as**
- **route-map**
- **route-reflector-client**
- **send-community**
- **send-label**
- **soft-reconfiguration**
- **unsuppress-map**

- **weight**

ピア ポリシー テンプレートは、特定のアドレス ファミリに属するネイバーに設定される BGP ポリシー コマンドの設定に使用されます。ピア セッション テンプレートと同様、ピア ポリシーテンプレートを一度設定しておくと、直接適用、または継承を通じて、多数のネイバーにピア ポリシーテンプレートを適用することができます。ピア ポリシー テンプレートの設定により、自律システム内のすべてのネイバーに適用される BGP ポリシー コマンドの設定が簡略化されます。

ピア セッション テンプレートと同様、ピア ポリシー テンプレートは継承をサポートしています。しかし、多少の違いはあります。直接適用されたピア ポリシー テンプレートは、最大 7 つのピア ポリシー テンプレートから設定を直接的または間接継承できます。したがって、合計 8 つのピア ポリシーテンプレートをネイバーまたはネイバー グループに適用できます。ルート マップと同じように、継承されたピア ポリシー テンプレートにはシーケンス番号が設定されます。また、ルート マップと同じように、継承されたピア ポリシー テンプレートは、最も低いシーケンス番号を持つ **inherit peer-policy** 文が最初に評価され、最も高いシーケンス番号のものが最後に評価されます。ただし、ピア ポリシー テンプレートはルート マップのように折りたたむことはできません。シーケンスはすべて評価されます。異なる値を使って、BGP ポリシー コマンドが再適用された場合は、シーケンス番号の小さいものから順に、前の値がすべて上書きされます。

直接適用されたピア ポリシーテンプレートと、シーケンス番号が最も大きい **inherit peer-policy** 文のプライオリティは常に最も高く、最後に適用されます。これ以降のピア テンプレートに再適用されるコマンドは、必ず、前の値を上書きします。この動作は、個々のポリシー コンフィギュレーション コマンドを繰り返さずとも、共通のポリシー コンフィギュレーションは大規模なネイバー グループに適用し、特定のポリシー コンフィギュレーションは特定のネイバーやネイバー グループだけに適用できるように設計されています。

ピア ポリシー テンプレートは、ポリシー コンフィギュレーション コマンドだけをサポートします。特定のアドレス ファミリ用に設定される BGP ポリシー コンフィギュレーション コマンドは、ピア ポリシー テンプレートで設定されます。

ピア ポリシー テンプレートの設定により、BGP 設定が簡略化され、柔軟性が向上します。特定のポリシーを 1 回設定すれば、何回も参照できます。ピア ポリシーは最大 8 レベルの継承をサポートするため、非常に具体的で複雑な BGP ポリシーも作成できます。

BGP ルート マップ ネクスト ホップ セルフ

BGP ルート マップ ネクスト ホップ セルフ機能は、bgp next-hop unchanged と bgp next-hop unchanged allpaths の設定を選択的にオーバーライドする方法を提供します。これらの設定はアドレス ファミリに対してグローバルに適用されます。ルートによっては、これは適切でない場合があります。たとえば、スタティック ルートは、自身をネクスト ホップとして再配布する必要がある一方で、接続ルート、および内部ボーダー ゲートウェイ プロトコル（IBGP）または外部ボーダー ゲートウェイ プロトコル（EBGP）を介して学習されたルートは、引き続きネクスト ホップを変更せずに再配布する場合があります。

BGP ルート マップ ネクスト ホップ セルフ機能は、bgp next-hop unchanged 設定と bgp next-hop unchanged allpaths 設定をオーバーライドする新しい ip next-hop self 設定を構成できるように、既存のルート マップ インフラストラクチャを変更します。

ip next-hop self 設定は、VPNv4 および VPNv6 アドレス ファミリにのみ適用されます。BGP 以外のプロトコルによって配布されるルートは影響を受けません。

新しい bgp route-map priority 設定を使用すると、bgp next-hop unchanged と bgp next-hop unchanged allpaths の設定よりもルート マップが優先されることを BGP に通知できます。bgp route-map priority 設定は、BGP にのみ影響します。bgp next-hop unchanged または bgp next-hop unchanged allpaths 設定を構成していない場合、bgp route-map priority 設定は効果がありません。

BGP の設定方法

BGP のデフォルト設定

下の表に、BGP のデフォルト設定を示します。

表 *27 : BGP* のデフォルト設定

機能	デフォルト設定
集約アドレス	無効：未定義
AS パス アクセス リスト	未定義
自動サマリー	ディセーブル。
最適パス	• ルータはルートを選択する場合に *as-path* を考慮し、外部 BGP ピ似ルートは比較しません。 • ルータ ID の比較：無効
BGP コミュニティ リスト	• 番号：未定義。コミュニティ番号を示す特定の値を許可すると、いないその他すべてのコミュニティ番号は、暗黙の拒否にデフォれます。 • フォーマット：シスコ デフォルト フォーマット（32 ビット番号
BGP 連合 ID/ピア	• ID：未設定 • ピア：識別なし
BGP 高速外部フォールオーバー	有効
BGP ローカル初期設定	100。指定できる範囲は 0～4294967295 です（大きな値を推奨）。

機能	デフォルト設定
BGP ネットワーク	指定なし。バックドア ルートのアドバタイズなし
BGP ルート ダンプニング	デフォルトでは、無効です。有効の場合は、次のようになります。 • 半減期は 15 分 • 再使用は 750（10 秒増分） • 抑制は 2000（10 秒増分） • 最大抑制時間は半減期の 4 倍（60 分）
BGP ルータ ID	ループバック インターフェイスに IP アドレスが設定されている場 バック インターフェイスの IP アドレス、またはルータの物理イン 対して設定された最大の IP アドレス
デフォルトの情報送信元（プロトコルまたはネットワーク再配布）	ディセーブル。
デフォルト メトリック	自動メトリック変換（組み込み）
ディスタンス	• 外部ルート アドミニストレーティブ ディスタンス：20（有効 • 内部ルート アドミニストレーティブ ディスタンス：200（有効 • ローカル ルート アドミニストレーティブ ディスタンス：200 255）
ディストリビュート リスト	• 入力（アップデート中に受信されたネットワークをフィルタリ • 出力（アップデート中のネットワークのアドバタイズを抑制）
内部ルート再配布	無効
IP プレフィックス リスト	未定義
Multi Exit Discriminator（MED）	• 常に比較：無効。異なる自律システム内のネイバーからのパス を比較しません。 • 最適パスの比較：無効 • 最悪パスである MED の除外：無効 • 決定的な MED 比較：無効

機能	デフォルト設定
ネイバー	• アドバタイズメント インターバル：外部ピアの場合は 30 秒、内部 は 5 秒 • ロギング変更：有効 • 条件付きアドバタイズ：無効 • デフォルト送信元：ネイバーに送信されるデフォルト ルートはな • 説明：なし • ディストリビュート リスト：未定義 • 外部 BGP マルチホップ：直接接続されたネイバーだけを許可 • フィルタ リスト：使用しない • 受信したプレフィックスの最大数：制限なし • ネクスト ホップ（BGP ネイバーのネクスト ホップとなるルータ） • パスワード：無効 • ピア グループ：定義なし、割り当てメンバーなし • プレフィックス リスト：指定なし • リモート AS（ネイバー BGP テーブルへのエントリ追加）：ピア • プライベート AS 番号の削除：無効 • ルート マップ：ピアへの適用なし • コミュニティ属性送信：ネイバーへの送信なし。 • シャットダウンまたはソフト再設定：無効 • タイマー：60 秒、ホールドタイム：180 秒 • アップデート送信元：最適ローカル アドレス • バージョン：BGP バージョン 4 • 重み：BGP ピアによって学習されたルート：0、ローカル ルータ れたルート：32768
NSF[1] 認識	無効にされた NSF 認識は、グレースフルリスタートを有効にすること Network Advantage ライセンスを実行するスイッチ上で IPv4 に対して有 す。[2]。有効な場合、レイヤ 3 スイッチでは、ハードウェアやソフトウ 中に、隣接する NSF 対応ルータからのパケットを転送し続けることが
ルート リフレクタ	未設定

機能	デフォルト設定
同期化（BGP および IGP）	無効
テーブル マップ アップデート	無効
タイマー	キープアライブ：60 秒、ホールドタイム：180 秒

1 Nonstop Forwarding
2

BGP ルーティングのイネーブル化

始める前に

（注） EIGRP を有効にするには、スタンドアロンスイッチまたはアクティブスイッチで Network Advantage ライセンスを実行している必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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 routing** 例： `Device(config)# ip routing`	IP ルーティングを有効にします。
ステップ 4	**router bgp** *autonomous-system* 例： `Device(config)# router bgp 45000`	BGP ルーティング プロセスを有効にして AS 番号を割り当て、ルータ コンフィギュレーション モードを開始します。指定できる AS 番号は 1～65535 です。64512～65535 は、プライベート AS 番号専用です。

	コマンドまたはアクション	目的
ステップ 5	**network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*] 例： `Device(config-router)# network 10.108.0.0`	このASに対してローカルとなるようにネットワークを設定し、BGP テーブルにネットワークを格納します。
ステップ 6	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *number* 例： `Device(config-router)# neighbor 10.108.1.2 remote-as 65200`	BGP ネイバー テーブルに設定を追加し、IP アドレスによって識別されるネイバーが、指定された AS に属することを示します。 EBGP の場合、通常ネイバーは直接接続されており、IP アドレスは接続のもう一方の端におけるインターフェイスのアドレスです。 IBGP の場合、IP アドレスにはルータ インターフェイス内の任意のアドレスを指定できます。
ステップ 7	**neighbor** {*ip-address* | *peer-group-name*} **remove-private-as** 例： `Device(config-router)# neighbor 172.16.2.33 remove-private-as`	（任意）発信ルーティング アップデート内の AS パスからプライベート AS 番号を削除します。
ステップ 8	**synchronization** 例： `Device(config-router)# synchronization`	（任意）BGP と IGP の同期化を有効にします。
ステップ 9	**auto-summary** 例： `Device(config-router)# auto-summary`	（任意）自動ネットワーク サマライズを有効にします。IGP から BGP にサブネットが再配布された場合、ネットワーク ルートだけが BGP テーブルに挿入されます。
ステップ 10	**bgp graceful-restart** 例： `Device(config-router)# bgp graceful-start`	（任意）NSF 認識をスイッチで有効にします。NSF 認識はデフォルトでは無効です。
ステップ 11	**end** 例： `Device(config-router)# end`	特権 EXEC モードに戻ります。
ステップ 12	**show ip bgp network** *network-number* 例：	設定を確認します。

	コマンドまたはアクション	目的
	`Device# show ip bgp network 10.108.0.0`	
ステップ 13	**show ip bgp neighbor** 例： `Device# show ip bgp neighbor`	NSF 認識（グレースフル リスタート）がネイバーで有効にされていることを確認します。スイッチおよびネイバーでNSF認識が有効になっている場合、次のメッセージが表示されます。Graceful Restart Capability: advertised and received スイッチでNSF認識が有効になっていて、ネイバーで有効になっていない場合、次のメッセージが表示されます。Graceful Restart Capability: advertised
ステップ 14	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ルーティング ポリシー変更の管理

BGP ピアがルート リフレッシュ機能をサポートするかどうかを学習して、BGP セッションをリセットするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**show ip bgp neighbors** 例： `Device# show ip bgp neighbors`	ネイバーがルートリフレッシュ機能をサポートするかどうかを表示します。サポートされている場合は、ルータに関する次のメッセージが表示されます。 *Received route refresh capability from peer*
ステップ 2	**clear ip bgp** {***** | *address* | *peer-group-name*} 例： `Device# clear ip bgp *`	指定された接続上でルーティングテーブルをリセットします。 • すべての接続をリセットする場合は、アスタリスク（*）を入力します。 • 特定の接続をリセットする場合は、IPアドレスを入力します。 • ピア グループをリセットする場合は、ピア グループ名を入力します。

	コマンドまたはアクション	目的
ステップ 3	**clear ip bgp** {***** | *address* | *peer-group-name*} **soft out** 例： `Device# clear ip bgp * soft out`	（任意）指定された接続上でインバウンドルーティングテーブルをリセットするには、アウトバウンドソフト リセットを実行します。このコマンドは、ルートリフレッシュがサポートされている場合に使用してください。 • すべての接続をリセットする場合は、アスタリスク（*）を入力します。 • 特定の接続をリセットする場合は、IP アドレスを入力します。 • ピア グループをリセットする場合は、ピア グループ名を入力します。
ステップ 4	**show ip bgp** 例： `Device# show ip bgp`	ルーティング テーブルおよび BGP ネイバーに関する情報をチェックし、リセットされたことを確認します。
ステップ 5	**show ip bgp neighbors** 例： `Device# show ip bgp neighbors`	ルーティング テーブルおよび BGP ネイバーに関する情報をチェックし、リセットされたことを確認します。

BGP 判断属性の設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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。

	コマンドまたはアクション	目的
ステップ 3	**router bgp** *autonomous-system* 例： `Device(config)# router bgp 4500`	BGP ルーティング プロセスを有効にして AS 番号を割り当て、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 4	**bgp best-path as-path ignore** 例： `Device(config-router)# bgp bestpath as-path ignore`	（任意）ルート選択中にASパス長を無視するようにルータを設定します。
ステップ 5	**neighbor** {*ip-address* | *peer-group-name*} **next-hop-self** 例： `Device(config-router)# neighbor 10.108.1.1 next-hop-self`	（任意）ネクストホップ アドレスの代わりに使用される特定の IP アドレスを入力し、ネイバーへの BGP アップデートに関するネクストホップの処理を無効にします。
ステップ 6	**neighbor** {*ip-address* | *peer-group-name*} **weight** *weight* 例： `Device(config-router)# neighbor 172.16.12.1 weight 50`	（任意）ネイバー接続に重みを割り当てます。指定できる値は0～65535です。最大の重みのルートを推奨します。別の BGP ピアから学習されたルートのデフォルトの重みは 0 です。ローカル ルータから送信されたルートのデフォルトの重みは32768です。
ステップ 7	**default-metric** *number* 例： `Device(config-router)# default-metric 300`	（任意）推奨パスを外部ネイバーに設定するようにMEDメトリックを設定します。MEDを持たないすべてのルータも、この値に設定されます。指定できる範囲は 1 ～ 4294967295 です。最小値を推奨します。
ステップ 8	**bgp bestpath med missing-as-worst** 例： `Device(config-router)# bgp bestpath med missing-as-worst`	（任意）MED がない場合は無限の値が指定されていると見なし、MED 値を持たないパスが最も望ましくないパスになるように、スイッチを設定します。
ステップ 9	**bgp always-compare med** 例： `Device(config-router)# bgp always-compare-med`	（任意）異なるAS内のネイバーからのパスに対して、MEDを比較するようにスイッチを設定します。デフォルトでは、MED は同じ AS 内のパス間でだけ比較されます。
ステップ 10	**bgp bestpath med confed** 例： `Device(config-router)# bgp bestpath med confed`	（任意）連合内の異なるサブASによってアドバタイズされたパスから特定のパスを選択する場合に、MED を考慮するようにスイッチを設定します。

	コマンドまたはアクション	目的
ステップ 11	**bgp deterministic med** 例： `Device(config-router)# bgp deterministic med`	（任意）同じAS内の異なるピアによってアドバタイズされたルートから選択する場合に、MED 変数を考慮するようにスイッチを設定します。
ステップ 12	**bgp default local-preference** *value* 例： `Device(config-router)# bgp default local-preference 200`	（任意）デフォルトのローカル プリファレンス値を変更します。指定できる範囲は 0 ～ 4294967295 で、デフォルト値は 100 です。最大のローカル プリファレンス値を推奨します。
ステップ 13	**maximum-paths** *number* 例： `Device(config-router)# maximum-paths 8`	（任意）IP ルーティング テーブルに追加するパスの数を設定します。デフォルトでは、最適パスだけがルーティング テーブルに追加されます。指定できる範囲は 1 ～ 16 です。複数の値を指定すると、パス間のロード バランシングが可能になります。スイッチ ソフトウェア では最大32の等コスト ルートが許可されていますが、スイッチ ハードウェア はルートあたり 17 パス以上は使用しません。
ステップ 14	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 15	**show ip bgp** 例： `Device# show ip bgp`	ルーティング テーブルおよびBGP ネイバーに関する情報をチェックし、リセットされたことを確認します。
ステップ 16	**show ip bgp neighbors** 例： `Device# show ip bgp neighbors`	ルーティング テーブルおよびBGP ネイバーに関する情報をチェックし、リセットされたことを確認します。
ステップ 17	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ルート マップによる BGP フィルタリングの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] 例： `Device(config)# route-map set-peer-address 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 4	**set ip next-hop** *ip-address* [*...ip-address*] [**peer-address**] 例： `Device(config)# set ip next-hop 10.1.1.3`	（任意）ネクストホップ処理を無効にするようにルート マップを設定します。 • インバウンド ルート マップの場合は、一致するルートのネクスト ホップをネイバー ピア アドレスに設定し、サードパーティのネクスト ホップを上書きします。 • BGP ピアのアウトバウンド ルート マップの場合は、ネクスト ホップをローカル ルータのピア アドレスに設定して、ネクスト ホップ計算を無効にします。
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show route-map** [*map-name*] 例： `Device# show route-map`	設定を確認するため、設定されたすべてのルート マップ、または指定されたルートマップだけを表示します。

	コマンドまたはアクション	目的
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ネイバーによる BGP フィルタリングの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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* 例： `Device(config)# router bgp 109`	BGP ルーティング プロセスを有効にして AS 番号を割り当て、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 4	**neighbor** {*ip-address* | *peer-group name*} **distribute-list** {**access-list-number** | *name*} {**in** | **out**} 例： `Device(config-router)# neighbor 172.16.4.1 distribute-list 39 in`	（任意）アクセス リストの指定に従って、ネイバーに対して送受信される BGP ルーティング アップデートをフィルタリングします。 （注） **neighbor prefix-list** ルータ コンフィギュレーション コマンドを使用して、アップデートをフィルタリングすることもできますが、両方のコマンドを使用して同じ BGP ピアを設定することはできません。
ステップ 5	**neighbor** {*ip-address* | *peer-group name*} **route-map** *map-tag* {**in** | **out**} 例： `Device(config-router)# neighbor 172.16.70.24 route-map internal-map in`	（任意）ルート マップを適用し、着信または発信ルートをフィルタリングします。

	コマンドまたはアクション	目的
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show ip bgp neighbors** 例： `Device# show ip bgp neighbors`	設定を確認します。
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

アクセス リストおよびネイバーによる BGP フィルタリングの設定

BGP 自律システム パスに基づいて着信および発信の両方のアップデートにアクセス リスト フィルタを指定して、フィルタリングすることもできます。各フィルタは、正規表現を使用するアクセス リストです。この方法を使用するには、自律システム パスのアクセス リストを定義し、特定のネイバーとの間のアップデートに適用します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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 as-path access-list** *access-list-number* {**permit** | **deny**} *as-regular-expressions* 例： `Device(config)# ip as-path access-list 1 deny _65535_`	BGP-related アクセス リストを定義します。

	コマンドまたはアクション	目的
ステップ 4	**router bgp** *autonomous-system* 例： `Device(config)# router bgp 110`	BGP 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 5	**neighbor** {*ip-address* | *peer-group name*} **filter-list** {*access-list-number* | *name*} {**in** | **out** | **weight** *weight*} 例： `Device(config-router)# neighbor 172.16.1.1 filter-list 1 out`	アクセス リストに基づいて、BGP フィルタを確立します。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show ip bgp neighbors** [**paths** *regular-expression*] 例： `Device# show ip bgp neighbors`	設定を確認します。
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

BGP フィルタリング用のプレフィックス リストの設定

コンフィギュレーションエントリを削除する場合は、シーケンス番号を指定する必要はありません。**Show** コマンドの出力には、シーケンス番号が含まれます。

コマンド内でプレフィックス リストを使用する場合は、あらかじめプレフィックス リストを設定しておく必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	**configure terminal** 例： `Device# 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** | **permit** *network/len* [**ge** *ge-value*] [**le** *le-value*] 例： `Device(config)# ip prefix-list BLUE permit 172.16.1.0/24`	一致条件に合わせてアクセスを **deny** または **permit** するプレフィックスリストを作成します。シーケンス番号を指定することもできます。少なくとも 1 つの **permit** または **deny** 句を入力する必要があります。 • *network/len* は、ネットワーク番号およびネットワーク マスクの長さ（ビット単位）です。 •（任意）**ge** および **le** の値は、一致させるプレフィックス長を指定します。指定する *ge-value* および l*e-value* は次の条件を満たしている必要があります。*len* < *ge-value* < *le-value* < 32
ステップ 4	**ip prefix-list** *list-name* **seq** *seq-value* **deny** | **permit** *network/len* [**ge** *ge-value*] [**le** *le-value*] 例： `Device(config)# ip prefix-list BLUE seq 10 permit 172.24.1.0/24`	（任意）プレフィックス リストにエントリを追加し、そのエントリにシーケンス番号を割り当てます。
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show ip prefix list** [**detail** | **summary**] *name* [*network/len*] [**seq** *seq-num*] [**longer**] [**first-match**] 例： `Device# show ip prefix list summary test`	プレフィックス リストまたはプレフィックス リストエントリに関する情報を表示して、設定を確認します。
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

BGP コミュニティ フィルタリングの設定

デフォルトでは、COMMUNITIES 属性はネイバーに送信されません。COMMUNITIES 属性が特定の IP アドレスのネイバーに送信されるように指定するには、**neighbor send-community** ルータ コンフィギュレーション コマンドを使用します。

手順の概要

1. **enable**
2. **configure terminal**
3. **ip community-list** *community-list-number* {**permit** | **deny**} *community-number*
4. **router bgp** *autonomous-system*
5. **neighbor** {*ip-address* | *peer-group name*} **send-community**
6. **set comm-list** *list-num* **delete**
7. **exit**
8. **ip bgp-community new-format**
9. **end**
10. **show ip bgp community**
11. **copy running-config startup-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： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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 community-list** *community-list-number* {**permit**	**deny**} *community-number* 例： `Device(config)# ip community-list 1 permit 50000:10`	コミュニティ リストを作成し、番号を割り当てます。 • *community-list-number* は 1 ～ 99 の整数です。この値は、コミュニティの 1 つ以上の許可または拒否グループを識別します。 • *community-number* は、**set community** ルートマップ コンフィギュレーション コマンドで設定される番号です。
ステップ 4	**router bgp** *autonomous-system* 例：	BGP 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
	`Device(config)# router bgp 108`	
ステップ 5	**neighbor** {*ip-address* | *peer-group name*} **send-community** 例： `Device(config-router)# neighbor 172.16.70.23 send-community`	この IP アドレスのネイバーに送信する COMMUNITIES 属性を指定します。
ステップ 6	**set comm-list** *list-num* **delete** 例： `Device(config-router)# set comm-list 500 delete`	（任意）ルート マップで指定された標準または拡張コミュニティ リストと一致する着信または発信アップデートのコミュニティ属性から、コミュニティを削除します。
ステップ 7	**exit** 例： `Device(config-router)# end`	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 8	**ip bgp-community new-format** 例： `Device(config)# ip bgp-community new format`	（任意）AA:NN の形式で、BGP コミュニティを表示、解析します。 BGP コミュニティは、2 つの部分からなる 2 バイト長形式で表示されます。シスコのデフォルトのコミュニティ形式は、NNAA です。BGP に関する最新の RFC では、コミュニティは AA:NN の形式をとります。最初の部分は AS 番号で、その次の部分は 2 バイトの数値です。
ステップ 9	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 10	**show ip bgp community** 例： `Device# show ip bgp community`	設定を確認します。
ステップ 11	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

BGP ネイバーおよびピア グループの設定

各ネイバーに設定オプションを割り当てるには、ネイバーの IP アドレスを使用し、次に示すルータ コンフィギュレーション コマンドのいずれかを指定します。ピア グループにオプションを割り当てるには、ピア グループ名を使用し、いずれかのコマンドを指定します。**neighbor shutdown** ルータ コンフィギュレーション コマンドを使用して、コンフィギュレーション情報を削除せずに、BGP ピア、またはピアグループを削除することができます。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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*	BGP 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** *peer-group-name* **peer-group**	BGP ピア グループを作成します。
ステップ 5	**neighbor** *ip-address* **peer-group** *peer-group-name*	BGP ネイバーをピア グループのメンバにします。
ステップ 6	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *number*	BGP ネイバーを指定します。**remote-as** *number* を使用してピアグループが設定されていない場合は、このコマンドを使用し、EBGP ネイバーを含むピアグループを作成します。指定できる範囲は 1 ～ 65535 です。
ステップ 7	**neighbor** {*ip-address* | *peer-group-name*} **description** *text*	（任意）ネイバーに説明を関連付けます。
ステップ 8	**neighbor** {*ip-address* | *peer-group-name*} **default-originate** [**route-map** *map-name*]	（任意）BGP スピーカー（ローカル ルータ）にネイバーへのデフォルト ルート 0.0.0.0 の送信を許可して、このルートがデフォルト ルートとして使用されるようにします。
ステップ 9	**neighbor** {*ip-address* | *peer-group-name*} **send-community**	（任意）この IP アドレスのネイバーに送信する COMMUNITIES 属性を指定します。

	コマンドまたはアクション	目的
ステップ 10	**neighbor** {*ip-address* | *peer-group-name*} **update-source** *interface*	（任意）内部 BGP セッションに、TCP 接続に関するすべての操作インターフェイスの使用を許可します。
ステップ 11	**neighbor** {*ip-address* | *peer-group-name*} **ebgp-multihop**	（任意）ネイバーがセグメントに直接接続されていない場合でも、BGP セッションを使用可能にします。マルチホップ ピア アドレスへの唯一のルートがデフォルトルート（0.0.0.0）の場合、マルチホップ セッションは確立されません。
ステップ 12	**neighbor** {*ip-address* | *peer-group-name*} **local-as** *number*	（任意）ローカル AS として使用する AS 番号を指定します。指定できる範囲は 1 ～ 65535 です。
ステップ 13	**neighbor** {*ip-address* | *peer-group-name*} **advertisement-interval** *seconds*	（任意）BGP ルーティング アップデートを送信する最小インターバルを設定します。
ステップ 14	**neighbor** {*ip-address* | *peer-group-name*} **maximum-prefix** *maximum* [*threshold*]	（任意）ネイバーから受信できるプレフィックス数を制御します。指定できる範囲は 1 ～ 4294967295 です。*threshold*（任意）は、警告メッセージが生成される基準となる最大値（パーセンテージ）です。デフォルトは 75% です。
ステップ 15	**neighbor** {*ip-address* | *peer-group-name*} **next-hop-self**	（任意）ネイバー宛ての BGP アップデートに関して、ネクストホップでの処理を無効にします。
ステップ 16	neighbor {*ip-address* | *peer-group-name*} password *string*	（任意）TCP 接続での MD5 認証を BGP ピアに設定します。両方の BGP ピアに同じパスワードを設定する必要があります。そうしないと、BGP ピア間に接続が作成されません。
ステップ 17	**neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}	（任意）着信または発信ルートにルート マップを適用します。
ステップ 18	**neighbor** {*ip-address* | *peer-group-name*} **send-community**	（任意）この IP アドレスのネイバーに送信する COMMUNITIES 属性を指定します。
ステップ 19	**neighbor** {*ip-address* | *peer-group-name*} **timers** *keepalive holdtime*	（任意）ネイバーまたはピア グループ用のタイマーを設定します。 • *keepalive* インターバルは、キープアライブ メッセージがピアに送信される間隔です。指定できる範囲は 1 ～ 4294967295 秒です。デフォルト値は 60 秒です。 • *holdtime* は、キープアライブ メッセージを受信しなかった場合、ピアが非アクティブと宣言されるまでのインターバルです。指定できる範囲

	コマンドまたはアクション	目的
		は 1 ～ 4294967295 秒です。デフォルト値は 180 秒です。
ステップ 20	**neighbor** {*ip-address* | *peer-group-name*} **weight** *weight*	（任意）ネイバーからのすべてのルートに関する重みを指定します。
ステップ 21	**neighbor** {*ip-address* | *peer-group-name*} **distribute-list** {*access-list-number* | *name*} {**in** | **out**}	（任意）アクセスリストの指定に従って、ネイバーに対して送受信される BGP ルーティング アップデートをフィルタリングします。
ステップ 22	**neighbor** {*ip-address* | *peer-group-name*} **filter-list** *access-list-number* {**in** | **out** | **weight** *weight*}	（任意）BGP フィルタを確立します。
ステップ 23	**neighbor** {*ip-address* | *peer-group-name*} **version** *value*	（任意）ネイバーと通信するときに使用する BGP バージョンを指定します。
ステップ 24	**neighbor** {*ip-address* | *peer-group-name*} **soft-reconfiguration inbound**	（任意）受信したアップデートのストアを開始するようにソフトウェアを設定します。
ステップ 25	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 26	**show ip bgp neighbors**	設定を確認します。
ステップ 27	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ルーティング テーブルでの集約アドレスの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例：	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
	`Device# configure terminal`	
ステップ 3	**router bgp** *autonomous-system* 例： `Device(config)# router bgp 106`	BGP 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	**aggregate-address** *address mask* 例： `Device(config-router)# aggregate-address 10.0.0.0 255.0.0.0`	BGP ルーティング テーブル内に集約エントリを作成します。集約ルートはASからのルートとしてアドバタイズされます。情報が失われた可能性があることを示すため、アトミック集約属性が設定されます。
ステップ 5	**aggregate-address** *address mask* **as-set** 例： `Device(config-router)# aggregate-address 10.0.0.0 255.0.0.0 as-set`	（任意）AS 設定パス情報を生成します。このコマンドは、この前のコマンドと同じルールに従う集約エントリを作成します。ただし、アドバタイズされるパスは、すべてのパスに含まれる全要素で構成される AS_SET です。多くのパスを集約するときは、このキーワードを使用しないでください。このルートは絶えず取り消され、アップデートされます。
ステップ 6	**aggregate-address** *address-mask* **summary-only** 例： `Device(config-router)# aggregate-address 10.0.0.0 255.0.0.0 summary-only`	（任意）サマリー アドレスだけをアドバタイズします。
ステップ 7	**aggregate-address** *address mask* **suppress-map** *map-name* 例： `Device(config-router)# aggregate-address 10.0.0.0 255.0.0.0 suppress-map map1`	（任意）選択された、より具体的なルートを抑制します。
ステップ 8	**aggregate-address** *address mask* **advertise-map** *map-name* 例： `Device(config-router)# aggregate-address 10.0.0.0 255.0.0.0 advertise-map map2`	（任意）ルート マップによって指定された設定に基づいて集約を生成します。
ステップ 9	**aggregate-address** *address mask* **attribute-map** *map-name* 例：	（任意）ルート マップで指定された属性を持つ集約を生成します。

	コマンドまたはアクション	目的
	`Device(config-router)# aggregate-address 10.0.0.0 255.0.0.0 attribute-map map3`	
ステップ 10	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 11	**show ip bgp neighbors** [**advertised-routes**] 例： `Device# show ip bgp neighbors`	設定を確認します。
ステップ 12	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ルーティング ドメイン連合の設定

自律システムのグループの自律システム番号として機能する連合 ID を指定する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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* 例： `Device(config)# router bgp 100`	BGP 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	**bgp confederation identifier** *autonomous-system* 例： `Device(config)# bgp confederation identifier 50007`	BGP 連合 ID を設定します。
ステップ 5	**bgp confederation peers** *autonomous-system* [*autonomous-system ...*] 例： `Device(config)# bgp confederation peers 51000 51001 51002`	連合に属する AS、および特殊な EBGP ピアとして処理する AS を指定します。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show ip bgp neighbor** 例： `Device# show ip bgp neighbor`	設定を確認します。
ステップ 8	**show ip bgp network** 例： `Device# show ip bgp network`	設定を確認します。
ステップ 9	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーションファイルに設定を保存します。

BGP ルート リフレクタの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	**configure terminal** 例： `Device# 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* 例： `Device(config)# router bgp 101`	BGP 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*} **route-reflector-client** 例： `Device(config-router)# neighbor 172.16.70.24 route-reflector-client`	ローカル ルータを BGP ルート リフレクタとして、指定されたネイバーをクライアントとして、それぞれ設定します。
ステップ 5	**bgp cluster-id** *cluster-id* 例： `Device(config-router)# bgp cluster-id 10.0.1.2`	（任意）クラスタに複数のルート リフレクタが存在する場合、クラスタ ID を設定します。
ステップ 6	**no bgp client-to-client reflection** 例： `Device(config-router)# no bgp client-to-client reflection`	（任意）クライアント間のルート反映を無効にします。デフォルトでは、ルート リフレクタ クライアントからのルートは、他のクライアントに反映されます。ただし、クライアントが完全メッシュ構造の場合、ルート リフレクタはルートをクライアントに反映させる必要がありません。
ステップ 7	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 8	**show ip bgp** 例： `Device# show ip bgp`	設定を確認します。送信元 ID およびクラスタ リスト属性を表示します。
ステップ 9	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ルート ダンプニングの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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* 例： `Device(config)# router bgp 100`	BGP 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	**bgp dampening** 例： `Device(config-router)# bgp dampening`	BGP ルート ダンプニングを有効にします。
ステップ 5	**bgp dampening** *half-life reuse suppress max-suppress* [**route-map** *map*] 例： `Device(config-router)# bgp dampening 30 1500 10000 120`	（任意）ルート ダンプニング係数のデフォルト値を変更します。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show ip bgp flap-statistics** [{ **regexp** *regexp*} | { **filter-list** *list*} | {*address mask* [**longer-prefix**]}] 例： `Device# show ip bgp flap-statistics`	（任意）フラッピングしているすべてのパスのフラップを監視します。ルートの抑制が終了し、安定状態になると、統計情報が削除されます。
ステップ 8	**show ip bgp dampened-paths** 例：	（任意）抑制されるまでの時間を含めて、ダンプニングされたルートを表示します。

	コマンドまたはアクション	目的
	`Device# show pi bgp dampened-paths`	
ステップ 9	**clear ip bgp flap-statistics** [{ **regexp** *regexp*} | { **filter-list** *list*} | {*address mask* [**longer-prefix**]} 例： `Device# clear ip bgp flap-statistics`	（任意）BGP フラップ統計情報を消去して、ルートがダンプニングされる可能性を小さくします。
ステップ 10	**clear ip bgp dampening** 例： `Device# clear ip bgp dampening`	（任意）ルートダンプニング情報を消去して、ルートの抑制を解除します。
ステップ 11	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

BGP ルートの条件付き注入

標準のルート集約を通じて選択された具体性にかけるプレフィックスではなく、より具体的なプレフィックスを BGP ルーティング テーブルに注入するには、この作業を実行します。より具体的なプレフィックスを使用すると、集約されたルートを使う場合よりも、よりきめ細かなトラフィック エンジニアリングや管理制御を行うことができます。

始める前に

この作業は、BGP ピアに対して、IGP がすでに設定されていることを前提にしています。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **bgp inject-map** *inject-map-name* **exist-map** *exist-map-name* [**copy-attributes**]
5. **exit**
6. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
7. **match ip address** {*access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...* | *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]}
8. **match ip route-source** {*access-list-number* | *access-list-name*} [*access-list-number...*| *access-list-name...*]
9. **exit**
10. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]

11. **set ip address** {*access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...* | *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]}
12. **set community** {*community-number* [**additive**] [*well-known-community*] | **none**}
13. **exit**
14. **ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network/length* | **permit** *network/length*} [**ge** *ge-value*] [**le** *le-value*]
15. 作成される各プレフィックス リストについて、ステップ 14 を繰り返します。
16. **exit**
17. **show ip bgp injected-paths**

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： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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： `Device(config)# router bgp 40000`	指定したルーティング プロセスのルータ 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	**bgp inject-map** *inject-map-name* **exist-map** *exist-map-name* [**copy-attributes**] 例： `Device(config-router)# bgp inject-map ORIGINATE exist-map LEARNED_PATH`	条件付きルート注入のために、注入マップと存在マップを指定します。 • 注入したルートが集約ルートの属性を継承することを指定するには、**copy-attributes** キーワードを使用します。
ステップ 5	**exit** 例： `Device(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 6	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] 例： `Device(config)# route-map LEARNED_PATH 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。

	コマンドまたはアクション	目的
ステップ 7	**match ip address** {*access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...* | *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]} 例： `Device(config-route-map)# match ip address prefix-list SOURCE`	より具体的なルートの注入先となる集約ルートを指定します。 • この例では、ルートのソースの再配布に、プレフィックス リスト SOURCE が使用されています。
ステップ 8	**match ip route-source** {*access-list-number* | *access-list-name*} [*access-list-number...*| *access-list-name...*] 例： `Device(config-route-map)# match ip route-source prefix-list ROUTE_SOURCE`	ルートのソースを再配布するための一致条件を指定します。 • この例では、ルートのソースの再配布に、プレフィックス リスト ROUTE_SOURCE が使用されています。 （注） ルート ソースは、**neighbor remote-as** コマンドで設定されたネイバー アドレスです。より具体的なルートの注入先とな注入る集約ルートを指定します。
ステップ 9	**exit** 例： `Device(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 10	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] 例： `Device(config)# route-map ORIGINATE 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。
ステップ 11	**set ip address** {*access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...* | *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]} 例： `Device(config-route-map)# set ip address prefix-list ORIGINATED_ROUTES`	注入されるルートを指定します。 この例では、ルートのソースの再配布に、プレフィックス リスト originated_routes が使用されています。
ステップ 12	**set community** {*community-number* [**additive**] [*well-known-community*] | **none**} 例： `Device(config-route-map)# set community 14616:555 additive`	注入されたルートの BGP コミュニティ属性を設定します。

	コマンドまたはアクション	目的
ステップ 13	**exit** 例： `Device(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 14	**ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network/length* | **permit** *network/length*} [**ge** *ge-value*] [**le** *le-value*] 例： `Device(config)# ip prefix-list SOURCE permit 10.1.1.0/24`	プレフィックス リストを設定します。 この例では、プレフィックス リスト SOURCE は、ネットワーク 10.1.1.0/24 からのルートを許可するように設定されています。
ステップ 15	作成される各プレフィックス リストについて、ステップ 14 を繰り返します。	--
ステップ 16	**exit** 例： `Device(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 モードに戻ります。
ステップ 17	**show ip bgp injected-paths** 例： `Device# show ip bgp injected-paths`	（任意）注入されたパスに関する情報を表示します。

ピア セッション テンプレートの設定

次の作業では、ピアセッションテンプレートを作成し、設定します。

基本的なピア セッション テンプレートの設定

一般的な BGP ルーティング セッション コマンドを使って、この次に説明する 2 つの作業のうち 1 つを使用して、多数のネイバーに適用できる基本的なピア セッション テンプレートを作成するには、この作業を実行します。

（注） ステップ 5 と 6 のコマンドは任意で、サポートされている一般的なセッション コマンドのいずれとでも置き換えが可能です。

（注） ピア セッション テンプレートには、次の制約事項が適用されます。

- ピア セッション テンプレートが直接継承できるセッション テンプレートは 1 つだけです。また、継承されたセッション テンプレートはそれぞれ、間接継承されたセッション テンプレートを 1 つ含むことができます。したがって、ネイバー、またはネイバー グループの設定には、直接適用されたピア セッション テンプレートを 1 個だけと、間接継承されたピア セッション テンプレートを 7 個使用できます。
- BGP ネイバーを、ピア グループとピア テンプレートの両方と連動するようには設定できません。BGP ネイバーは、1 つのピア グループだけに属するように設定するか、またはピア テンプレートだけからポリシーを継承するように設定できます。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **template peer-session** *session-template-name*
5. **remote-as** *autonomous-system-number*
6. **timers** *keepalive-interval hold-time*
7. **end**
8. **show ip bgp template peer-session** [*session-template-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： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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： `Device(config)# router bgp 101`	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。

	コマンドまたはアクション	目的
ステップ 4	**template peer-session** *session-template-name* 例： `Device(config-router)# template peer-session INTERNAL-BGP`	セッション テンプレート コンフィギュレーション モードを開始して、ピア セッション テンプレートを作成します。
ステップ 5	**remote-as** *autonomous-system-number* 例： `Device(config-router-stmp)# remote-as 202`	（任意）指定された自律システムでリモート ネイバーとのピアリングを設定します。 （注） ここでは、サポートされている一般セッション コマンドならどれでも使用できます。サポートされているコマンドの一覧については、「制限事項」の項を参照してください。
ステップ 6	**timers** *keepalive-interval hold-time* 例： `Device(config-router-stmp)# timers 30 300`	（任意）BGP キープアライブとホールド タイマーを設定します。 ホールド タイムは、少なくともキープアライブ タイムの 2 倍の長さが必要です。 （注） ここでは、サポートされている一般セッション コマンドならどれでも使用できます。サポートされているコマンドの一覧については、「制限事項」の項を参照してください。
ステップ 7	**end** 例： `Device(config-router)# end`	セッション テンプレート コンフィギュレーション モードを終了して、特権 EXEC モードに戻ります。
ステップ 8	**show ip bgp template peer-session** [*session-template-name*] 例： `Device# show ip bgp template peer-session`	ローカルに設定されたピア セッション テンプレートを表示します。 *session-template-name* 引数を使用して、ピア ポリシー テンプレートが 1 つだけ表示されるように、出力をフィルタ処理できます。また、このコマンドは、標準出力修飾子すべてをサポートしています。

inherit peer-session コマンドを使用したピア セッション テンプレートの継承の設定

この作業は、**inherit peer-session** コマンドを使用して、ピア セッション テンプレートの継承を設定します。これは、ピア セッション テンプレートを作成、設定し、別のピア セッション テンプレートからコンフィギュレーションを継承できるようにします。

（注） ステップ 5 と 6 のコマンドは任意で、サポートされている一般的なセッション コマンドのいずれとでも置き換えが可能です。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **template peer-session** *session-template-name*
5. **description** *text-string*
6. **update-source** *interface-type interface-number*
7. **inherit peer-session** *session-template-name*
8. **end**
9. **show ip bgp template peer-session** [*session-template-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： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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： `Device(config)# router bgp 101`	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	**template peer-session** *session-template-name* 例： `Device(config-router)# template peer-session CORE1`	セッション テンプレート コンフィギュレーション モードを開始して、ピア セッション テンプレートを作成します。
ステップ 5	**description** *text-string* 例： `Device(config-router-stmp)# description CORE-123`	（任意）説明を設定します。 text-string には最大 80 文字を使用できます。

	コマンドまたはアクション	目的
		（注） ここでは、サポートされている一般セッション コマンドならどれでも使用できます。サポートされているコマンドの一覧については、「制限事項」の項を参照してください。
ステップ 6	**update-source** *interface-type interface-number* 例： `Device(config-router-stmp)# update-source loopback 1`	（任意）ルーティング テーブル アップデートを受信するための特定のソース、またはインターフェイスを選択するようにルータを設定します。 この例では、ループバック インターフェイスを使用します。このコンフィギュレーションの利点は、ループバック インターフェイスはフラッピングしているインターフェイスの影響を受けにくいところにあります。 （注） ここでは、サポートされている一般セッション コマンドならどれでも使用できます。サポートされているコマンドの一覧については、「制限事項」の項を参照してください。
ステップ 7	**inherit peer-session** *session-template-name* 例： `Device(config-router-stmp)# inherit peer-session INTERNAL-BGP`	別のピア セッション テンプレートのコンフィギュレーションを継承するように、このピア セッション テンプレートを設定します。 この例では、INTERNAL-BGP からコンフィギュレーションを継承するようにピア セッション テンプレートを設定しています。このテンプレートはネイバーに適用可能で、コンフィギュレーション INTERNAL-BGP は間接的に適用されます。その他のピア セッション テンプレートは直接適用できません。ただし、直接継承されたテンプレートは最高 7 個の間接継承されたピア セッション テンプレートを持つことができます。
ステップ 8	**end** 例： `Device(config-router)# end`	セッション テンプレート 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 9	**show ip bgp template peer-session** [*session-template-name*] 例： `Device# show ip bgp template peer-session`	ローカルに設定されたピア セッション テンプレートを表示します。 オプションの *session-template-name* 引数を使用して、ピア ポリシー テンプレートが 1 つだけ表示される

	コマンドまたはアクション	目的
		ように、出力をフィルタ処理できます。また、このコマンドは、標準出力修飾子すべてをサポートしています。

neighbor inherit peer-session コマンドを使用したピア セッション テンプレートの継承の設定

この作業では、**neighbor inherit peer-session** コマンドを使用して、ピア セッション テンプレートをネイバーに送信し、指定されたピア セッション テンプレートからコンフィギュレーションを継承させるようにデバイスを設定します。次の手順に従って、ピア セッション テンプレート コンフィギュレーションをネイバーに送信し、継承させます。

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. **neighbor** *ip-address* **inherit peer-session** *session-template-name*
6. **end**
7. **show ip bgp template peer-session** [*session-template-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： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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： `Device(config)# router bgp 101`	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* **remote-as** *autonomous-system-number* 例：	指定されたネイバーを使ってピアリング セッションを設定します。 手順 5 の neighbor inherit 文を動作させるには、**remote-as** 文を明示的に使用する必要があります。

	コマンドまたはアクション	目的
	`Device(config-router)# neighbor 172.16.0.1 remote-as 202`	ピアリングが設定されていない場合、手順 5 で指定されたネイバーはセッションテンプレートを受け付けません。
ステップ 5	**neighbor** *ip-address* **inherit peer-session** *session-template-name* 例： `Device(config-router)# neighbor 172.16.0.1 inherit peer-session CORE1`	ネイバーがコンフィギュレーションを継承できるように、このネイバーにピア セッション テンプレートを送信します。 この例では、ピア セッション テンプレート CORE1 を 172.16.0.1 ネイバーに送信し、継承させるようにデバイスを設定しています。このテンプレートはネイバーに適用できます。また、別のピアセッション テンプレートが CORE1 で間接継承された場合、間接継承されたコンフィギュレーションも適用されます。その他のピア セッション テンプレートは直接適用できません。ただし、直接継承されたテンプレートも、さらに最高 7 個の間接継承されたピア セッション テンプレートを継承することができます。
ステップ 6	**end** 例： `Device(config-router)# end`	ルータ 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 7	**show ip bgp template peer-session** [*session-template-name*] 例： `Device# show ip bgp template peer-session`	ローカルに設定されたピア セッション テンプレートを表示します。 オプションの *session-template-name* 引数を使用して、ピア ポリシー テンプレートが 1 つだけ表示されるように、出力をフィルタ処理できます。また、このコマンドは、標準出力修飾子すべてをサポートしています。

ピア ポリシー テンプレートの設定

次の作業では、ピアポリシーテンプレートを作成し、設定します。

基本的なピア ポリシー テンプレートの設定

BGP ポリシー コンフィギュレーション コマンドを使って、この次に説明する 2 つの作業のうち 1 つを使用して、多数のネイバーに適用できる基本的なピア ポリシー テンプレートを作成するには、この作業を実行します。

（注） ステップ 5 ～ 7 のコマンドは任意で、サポートされている BGP ポリシー コンフィギュレーション コマンドのいずれとでも置き換えが可能です。

（注） ピア ポリシー テンプレートには、次の制約事項が適用されます。

- ピア ポリシー テンプレートは、直接的、または間接的に、最高 8 個のピア ポリシー テンプレートを継承できます。
- BGP ネイバーを、ピア グループとピア テンプレートの両方と連動するようには設定できません。BGP ネイバーは、1 つのピア グループだけに属するように設定するか、またはピア テンプレートだけからポリシーを継承するように設定できます。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **template peer-policy** *policy-template-name*
5. **maximum-prefix** *prefix-limit* [*threshold*] [**restart** *restart-interval* | **warning-only**]
6. **weight** *weight-value*
7. **prefix-list** *prefix-list-name* {**in** | **out**}
8. **end**

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： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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： `Device(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。

	コマンドまたはアクション	目的
ステップ 4	**template peer-policy** *policy-template-name* 例： `Device(config-router)# template peer-policy GLOBAL`	ポリシーテンプレートコンフィギュレーションモードを開始し、ピア ポリシー テンプレートを作成します。
ステップ 5	**maximum-prefix** *prefix-limit* [*threshold*] [**restart** *restart-interval* | **warning-only**] 例： `Device(config-router-ptmp)# maximum-prefix 10000`	（任意）このピアがネイバーから受け入れるプレフィックスの最大数を設定します。 （注） ここでは、サポートされている BGP ポリシー コンフィギュレーション コマンドならどれでも使用できます。サポートされているコマンドの一覧については、「ピア ポリシー テンプレート」の項を参照してください。
ステップ 6	**weight** *weight-value* 例： `Device(config-router-ptmp)# weight 300`	（任意）このネイバーから送信されるルートのデフォルトの重みを設定します。 （注） ここでは、サポートされている BGP ポリシー コンフィギュレーション コマンドならどれでも使用できます。サポートされているコマンドの一覧については、「ピア ポリシー テンプレート」の項を参照してください。
ステップ 7	**prefix-list** *prefix-list-name* {**in** | **out**} 例： `Device(config-router-ptmp)# prefix-list NO-MARKETING in`	（任意）ルータにより受信、またはルータから送信されるプレフィックスをフィルタします。 この例のプレフィックス リストは、インバウンド内部アドレスをフィルタします。 （注） ここでは、サポートされている BGP ポリシー コンフィギュレーション コマンドならどれでも使用できます。サポートされているコマンドの一覧については、「ピア ポリシー テンプレート」の項を参照してください。
ステップ 8	**end** 例： `Device(config-router-ptmp)# end`	ポリシーテンプレートコンフィギュレーションモードを終了して、特権 EXEC モードに戻ります。

inherit peer-policy コマンドを使用したピア ポリシー テンプレートの継承の設定

この作業は、**inherit peer-policy** コマンドを使用して、ピア ポリシー テンプレートの継承を設定します。これは、ピア ポリシー テンプレートを作成、設定し、別のピア ポリシー テンプレートからコンフィギュレーションを継承できるようにします。

（注） ステップ5と6のコマンドは任意で、サポートされているBGPポリシーコンフィギュレーション コマンドのいずれとでも置き換えが可能です。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **template peer-policy** *policy-template-name*
5. **route-map** *map-name* {**in**| **out**}
6. **inherit peer-policy** *policy-template-name sequence-number*
7. **end**
8. **show ip bgp template peer-policy** [*policy-template-name*[**detail**]]

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： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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： `Device(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	**template peer-policy** *policy-template-name* 例： `Device(config-router)# template peer-policy NETWORK1`	ポリシーテンプレートコンフィギュレーションモードを開始し、ピア ポリシー テンプレートを作成します。

	コマンドまたはアクション	目的
ステップ 5	**route-map** *map-name* {**in**| **out**} 例： `Device(config-router-ptmp)# route-map ROUTE in`	（任意）指定されたルート マップをインバウンド ルート、またはアウトバウンド ルートに適用します。 （注） ここでは、サポートされている BGP ポリシー コンフィギュレーション コマンドならどれでも使用できます。
ステップ 6	**inherit peer-policy** *policy-template-name sequence-number* 例： `Device(config-router-ptmp)# inherit peer-policy GLOBAL 10`	別のピア ポリシーテンプレートのコンフィギュレーションを継承するように、このピア ポリシー テンプレートを設定します。 • *sequence-number* 引数は、ピア ポリシー テンプレートの評価順序を設定します。ルート マップのシーケンス番号と同様、最も小さいシーケンス番号が最初に評価されます。 • この例では、GLOBAL からコンフィギュレーションを継承するようにピア ポリシー テンプレートを設定しています。これらの手順で作成されたテンプレートをネイバーに適用すると、コンフィギュレーション GLOBAL も間接継承され、適用されます。GLOBAL からはさらに最高 6 個のピア ポリシー テンプレートが間接継承され、合計 8 個のピア ポリシー テンプレートが直接適用、および間接継承されます。 • 他のテンプレートで、これより小さいシーケンス番号が設定されていなければ、この例のこのテンプレートが最初に評価されます。
ステップ 7	**end** 例： `Device(config-router-ptmp)# end`	ポリシーテンプレート コンフィギュレーション モードを終了して、特権 EXEC モードに戻ります。
ステップ 8	**show ip bgp template peer-policy** [*policy-template-name*[**detail**]] 例： `Device# show ip bgp template peer-policy NETWORK1 detail`	ローカルに設定されたピア ポリシー テンプレートを表示します。 • *policy-template-name* 引数を使用して、ピア ポリシー テンプレートが 1 つだけ表示されるように、出力をフィルタ処理できます。また、このコマンドは、標準出力修飾子すべてをサポートしています。

	コマンドまたはアクション	目的
		• 詳細なポリシー情報を表示するには、**detail** キーワードを使用します。

例

次の例は、**show ip bgp template peer-policy** コマンドに **detail** キーワードを付けた場合の出力で、NETWORK1 というポリシーの詳細が表示されています。この例の出力からは、GLOBAL テンプレートが継承されたことがわかります。ルートマップおよびプレフィックス リスト コンフィギュレーションの詳細も表示されています。

```
Device# show ip bgp template peer-policy NETWORK1 detail
Template:NETWORK1, index:2.
Local policies:0x1, Inherited polices:0x80840
This template inherits:
  GLOBAL, index:1, seq_no:10, flags:0x1
Locally configured policies:
  route-map ROUTE in
Inherited policies:
  prefix-list NO-MARKETING in
  weight 300
  maximum-prefix 10000
 Template:NETWORK1 <detail>
Locally configured policies:
  route-map ROUTE in
route-map ROUTE, permit, sequence 10
  Match clauses:
    ip address prefix-lists: DEFAULT
ip prefix-list DEFAULT: 1 entries
   seq 5 permit 10.1.1.0/24
  Set clauses:
  Policy routing matches: 0 packets, 0 bytes
Inherited policies:
  prefix-list NO-MARKETING in
ip prefix-list NO-MARKETING: 1 entries
   seq 5 deny 10.2.2.0/24
```

neighbor inherit peer-policy コマンドを使用したピア ポリシー テンプレートの継承の設定

この作業では、**neighbor inherit peer-policy** コマンドを使用して、ピアポリシーテンプレートをネイバーに送信し、継承させるようにデバイスを設定します。次の手順に従って、ピア ポリシー テンプレート コンフィギュレーションをネイバーに送信し、継承させます。

BGP ネイバーが複数レベルのピア テンプレートを使用する場合、ネイバーに適用されているポリシーを判断するのが難しいことがあります。**show ip bgp neighbors** コマンドの **policy** および **detail** キーワードは、指定されたネイバーに継承されたポリシーおよび直接設定されたポリシーを表示します。

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. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
6. **neighbor** *ip-address* **inherit peer-policy** *policy-template-name*
7. **end**
8. **show ip bgp neighbors** [*ip-address*[**policy** [**detail**]]]

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： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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： `Device(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* **remote-as** *autonomous-system-number* 例： `Device(config-router)# neighbor 192.168.1.2 remote-as 40000`	指定されたネイバーを使ってピアリング セッションを設定します。 • 手順 6 の **neighbor inherit** 文を動作させるには、**remote-as** 文を明示的に使用する必要があります。ピアリングが設定されていない場合、手順 6 で指定されたネイバーはセッション テンプレートを受け付けません。
ステップ 5	**address-family ipv4** [**multicast**	**unicast**	**vrf** *vrf-name*] 例： `Device(config-router)# address-family ipv4 unicast`	アドレス ファミリ固有のコマンド コンフィギュレーションを使用するようにネイバーを設定するために、アドレス ファミリ コンフィギュレーション モードを開始します。
ステップ 6	**neighbor** *ip-address* **inherit peer-policy** *policy-template-name* 例： `Device(config-router-af)# neighbor 192.168.1.2 inherit peer-policy GLOBAL`	ネイバーが設定を継承できるように、ピア ポリシー テンプレートをこのネイバーに送信します。 この例では、ピア ポリシー テンプレート GLOBAL を 192.168.1.2 ネイバーに送信し、継承させるようにルータを設定しています。このテンプレートはネイバーに適用できます。また、別のピア ポリシー テンプレートが GLOBAL から間接継承された場合、

	コマンドまたはアクション	目的
		間接継承されたコンフィギュレーションも適用されます。GLOBAL からは、さらに最高 7 個のピア ポリシー テンプレートを間接継承できます。
ステップ 7	**end** 例： `Device(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 モードに戻ります。
ステップ 8	**show ip bgp neighbors** [*ip-address*[**policy** [**detail**]]] 例： `Device# show ip bgp neighbors 192.168.1.2 policy`	ローカルに設定されたピア ポリシー テンプレートを表示します。 • *policy-template-name* 引数を使用して、ピア ポリシー テンプレートが 1 つだけ表示されるように、出力をフィルタ処理できます。また、このコマンドは、標準出力修飾子すべてをサポートしています。 • このネイバーに適用されているポリシーをアドレス ファミリごとに表示するには、**policy** キーワードを使用します。 • 詳細なポリシー情報を表示するには、**detail** キーワードを使用します。

例

次の出力例に表示されているのは、192.168.1.2 にあるネイバーに適用されたポリシーです。この出力には、継承されたポリシーと、このネイバーデバイスで設定されたポリシーの両方が表示されています。継承されたポリシーは、ピア グループ、またはピア ポリシー テンプレートからネイバーが継承したポリシーです。

```
Device# show ip bgp neighbors 192.168.1.2 policy
Neighbor: 192.168.1.2, Address-Family: IPv4 Unicast
Locally configured policies:
 route-map ROUTE in
Inherited polices:
 prefix-list NO-MARKETING in
 route-map ROUTE in
 weight 300
 maximum-prefix 10000
```

BGP ルートマップの next-hop self の設定

ip next-hop self 設定を追加し、bgp next-hop unchanged 設定と bgp next-hop unchanged allpaths 設定をオーバーライドして、既存のルート マップを変更するには、この作業を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag* **permit** *sequence-number*
4. **match source-protocol** *source-protocol*
5. **set ip next-hop self**
6. **exit**
7. **route-map** *map-tag* **permit** *sequence-number*
8. **match route-type internal**
9. **match route-type external**
10. **match source-protocol** *source-protocol*
11. **exit**
12. **router bgp** *autonomous-system-number*
13. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
14. **address-family vpnv4**
15. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**
16. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **next-hop unchanged allpaths**
17. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-map** *map-name* **out**
18. **exit**
19. **address-family ipv4** [**unicast** | **multicast**| **vrf** *vrf-name*]
20. **bgp route-map priority**
21. **redistribute** *protocol*
22. **redistribute** *protocol*
23. **exit-address-family**
24. **end**

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： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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	**route-map** *map-tag* **permit** *sequence-number* 例： `Device(config)# route-map static-nexthop-rewrite permit 10`	ルーティング プロトコル間でルートを再配布する条件を定義し、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。

	コマンドまたはアクション	目的
ステップ 4	**match source-protocol** *source-protocol* 例： `Device(config-route-map)# match source-protocol static`	送信元プロトコルに基づいて、Enhanced Interior Gateway Routing Protocol（EIGRP）の外部ルートを照合します。
ステップ 5	**set ip next-hop self** 例： `Device(config-route-map)# set ip next-hop self`	自身をネクスト ホップとするようにローカル ルート（BGP の場合のみ）を設定します。
ステップ 6	**exit** 例： `Device(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、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-tag* **permit** *sequence-number* 例： `Device(config)# route-map static-nexthop-rewrite permit 20`	ルーティング プロトコル間でルートを再配布する条件を定義し、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 route-type internal** 例： `Device(config-route-map)# match route-type internal`	指定されたタイプのルートを再配布します。
ステップ 9	**match route-type external** 例： `Device(config-route-map)# match route-type external`	指定されたタイプのルートを再配布します。
ステップ 10	**match source-protocol** *source-protocol* 例： `Device(config-route-map)# match source-protocol connected`	送信元プロトコルに基づいて、Enhanced Interior Gateway Routing Protocol（EIGRP）の外部ルートを照合します。
ステップ 11	**exit** 例： `Device(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、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	**router bgp** *autonomous-system-number* 例：	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。

	コマンドまたはアクション	目的
	`Device(config)# router bgp 45000`	
ステップ 13	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *autonomous-system-number* 例： `Device(config-router)# neighbor 172.16.232.50 remote-as 65001`	BGP ネイバー テーブルまたはマルチプロトコル BGP ネイバー テーブルにエントリを追加します。
ステップ 14	**address-family vpnv4** 例： `Device(config-router)# address-family vpnv4`	VPNv4 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。
ステップ 15	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate** 例： `Device(config-router-af)# neighbor 172.16.232.50 activate`	ボーダー ゲートウェイ プロトコル（BGP）ネイバーとの情報交換を有効にします。
ステップ 16	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **next-hop unchanged allpaths** 例： `Device(config-router-af)# neighbor 172.16.232.50 next-hop unchanged allpaths`	マルチホップとして設定されている外部 EBGP ピアで、ネクスト ホップを変更せずに伝播できるようにします。
ステップ 17	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-map** *map-name* **out** 例： `Device(config-router-af)# neighbor 172.16.232.50 route-map static-nexthop-rewrite out`	発信ルートにルート マップを適用します。
ステップ 18	**exit** 例： `Device(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 19	**address-family ipv4** [**unicast** | **multicast**| **vrf** *vrf-name*] 例： `Device(config-router)# address-family ipv4 unicast vrf inside`	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。

	コマンドまたはアクション	目的
ステップ 20	**bgp route-map priority** 例： `Device(config-router-af)# bgp route-map priority`	ローカル BGP ルーティング プロセスについてルート マップを優先することを設定します。
ステップ 21	**redistribute** *protocol* 例： `Device(config-router-af)# redistribute static`	ルートを 1 つのルーティング ドメインから他のルーティング ドメインに再配布します。
ステップ 22	**redistribute** *protocol* 例： `Device(config-router-af)# redistribute connected`	ルートを 1 つのルーティング ドメインから他のルーティング ドメインに再配布します。
ステップ 23	**exit-address-family** 例： `Device(config-router-af)# exit address-family`	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 モードを開始します。
ステップ 24	**end** 例： `Device(config-router)# end`	ルータ 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。

BGP の設定例

例：条件付き BGP ルートの注入の設定

次の出力例は、**show ip bgp injected-paths** コマンドを入力したときに表示される出力に類似しています。

```
Device# show ip bgp injected-paths

BGP table version is 11, local router ID is 10.0.0.1
Status codes:s suppressed, d damped, h history, * valid, > best, i -
internal
Origin codes:i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 172.16.0.0       10.0.0.2                               0 ?
*> 172.17.0.0/16    10.0.0.2                               0 ?
```

例：ピア セッション テンプレートの設定

次の例は、セッション テンプレート コンフィギュレーション モードで、INTERNAL-BGP という名前のピア セッション テンプレートを作成します。

```
router bgp 45000
 template peer-session INTERNAL-BGP
 remote-as 50000
 timers 30 300
 exit-peer-session
```

次の例は、ピア セッション テンプレート CORE1 を作成します。この例は、INTERNAL-BGP というピア セッション テンプレートのコンフィギュレーションを継承します。

```
router bgp 45000
 template peer-session CORE1
 description CORE-123
 update-source loopback 1
 inherit peer-session INTERNAL-BGP
 exit-peer-session
```

次の例は、CORE1 ピア セッション テンプレートを継承するように、192.168.3.2 ネイバーを設定します。192.168.3.2 ネイバーも、ピア セッション テンプレート INTERNAL-BGP から間接的にコンフィギュレーションを継承します。neighbor inherit 文を動作させるには、**remote-as** 文を明示的に使用する必要があります。ピアリングが設定されていない場合、指定されたネイバーはセッション テンプレートを受け付けません。

```
router bgp 45000
 neighbor 192.168.3.2 remote-as 50000
 neighbor 192.168.3.2 inherit peer-session CORE1
```

例：ピア ポリシー テンプレートの設定

次の例は、GLOBAL という名前のピア ポリシー テンプレートを作成し、ポリシー テンプレート コンフィギュレーション モードを開始します。

```
router bgp 45000
 template peer-policy GLOBAL
  weight 1000
  maximum-prefix 5000
  prefix-list NO_SALES in
  exit-peer-policy
```

次の例は、PRIMARY-IN という名前のピア ポリシー テンプレートを作成し、ポリシー テンプレート コンフィギュレーション モードを開始します。

```
router bgp 45000
 template peer-policy PRIMARY-IN
  prefix-list ALLOW-PRIMARY-A in
  route-map SET-LOCAL in
  weight 2345
  default-originate
  exit-peer-policy
```

次の例は、ピア ポリシー テンプレート CUSTOMER-A を作成します。このピア ポリシー テンプレートは、PRIMARY-IN および GLOBAL という名前のピア ポリシー テンプレートからコンフィギュレーションを継承するように設定されています。

```
router bgp 45000
 template peer-policy CUSTOMER-A
  route-map SET-COMMUNITY in
  filter-list 20 in
  inherit peer-policy PRIMARY-IN 20
  inherit peer-policy GLOBAL 10
  exit-peer-policy
```

次の例は、アドレス ファミリ モードでピア ポリシー テンプレート CUSTOMER-A を継承するように 192.168.2.2 ネイバーを設定します。この例は上の例の続きと仮定しており、上のピア ポリシー テンプレート CUSTOMER-A は PRIMARY-IN および GLOBAL という名前のテンプレートからコンフィギュレーションを継承しているため、192.168.2.2 ネイバーもピア ポリシー テンプレート PRIMARY-IN および GLOBAL から間接継承します。

```
router bgp 45000
 neighbor 192.168.2.2 remote-as 50000
 address-family ipv4 unicast
  neighbor 192.168.2.2 inherit peer-policy CUSTOMER-A
  end
```

例：BGP ルート マップの next-hop self の設定

この項では、BGP ルート マップの next-hop self を設定する方法の例を示します。

この例では、bgp next-hop unchanged と bgp next-hop unchanged allpaths の設定をオーバーライドするネットワークを照合するルート マップを設定します。次に、next-hop self を設定します。その後、指定したアドレス ファミリに対して bgp route-map priority を設定して、指定済みのルート マップが bgp next-hop unchanged と bgp next-hop unchanged allpaths の設定よりも優先されるようにします。この設定により、スタティック ルートは自身をネクスト ホップとして再配布されますが、接続されたルートおよび IBGP または EBGP を介して学習されたルートは引き続きネクスト ホップを変更せずに再配布されます。

```
route-map static-nexthop-rewrite permit 10
 match source-protocol static
 set ip next-hop self
route-map static-nexthop-rewrite permit 20
 match route-type internal
 match route-type external
 match source-protocol connected
!
router bgp 65000
 neighbor 172.16.232.50 remote-as 65001
 address-family vpnv4
   neighbor 172.16.232.50 activate
   neighbor 172.16.232.50 next-hop unchanged allpaths
   neighbor 172.16.232.50 route-map static-nexthop-rewrite out
 exit-address-family
 address-family ipv4 unicast vrf inside
   bgp route-map priority
   redistribute static
```

```
   redistribute connected
 exit-address-family
end
```

BGP のモニタリングおよびメンテナンス

特定のキャッシュ、テーブル、またはデータベースのすべての内容を削除できます。この作業は、特定の構造の内容が無効になった場合、または無効である疑いがある場合に必要となります。

BGP ルーティング テーブル、キャッシュ、データベースの内容など、特定の統計情報を表示できます。さらに、リソースの利用率を取得したり、ネットワーク問題を解決するための情報を使用することもできます。さらに、ノードの到達可能性に関する情報を表示し、デバイスのパケットが経由するネットワーク内のルーティング パスを検出することもできます。

下の図に、BGP を消去および表示するために使用する特権 EXEC コマンドを示します。

表 *28 : IP BGP* の *clear* および *show* コマンド

clear ip bgp *address*	特定の BGP 接続をリセットします。
clear ip bgp *	すべての BGP 接続をリセットします。
clear ip bgp peer-group *tag*	BGP ピア グループのすべてのメンバを削除しま
show ip bgp *prefix*	プレフィックスがアドバタイズされるピア グル ピア グループに含まれないピアを表示します。 プやローカル プレフィックスなどのプレフィッ 示されます。
show ip bgp cidr-only	サブネットおよびスーパーネット ネットワーク すべての BGP ルートを表示します。
show ip bgp community [*community-number*] [**exact**]	指定されたコミュニティに属するルートを表示
show ip bgp community-list *community-list-number* [**exact-match**]	コミュニティ リストで許可されたルートを表示
show ip bgp filter-list *access-list-number*	指定された AS パス アクセス リストによって照 トを表示します。
show ip bgp inconsistent-as	送信元の AS と矛盾するルートを表示します。
show ip bgp regexp *regular-expression*	コマンドラインに入力された特定の正規表現と スを持つルートを表示します。
show ip bgp	BGP ルーティング テーブルの内容を表示します

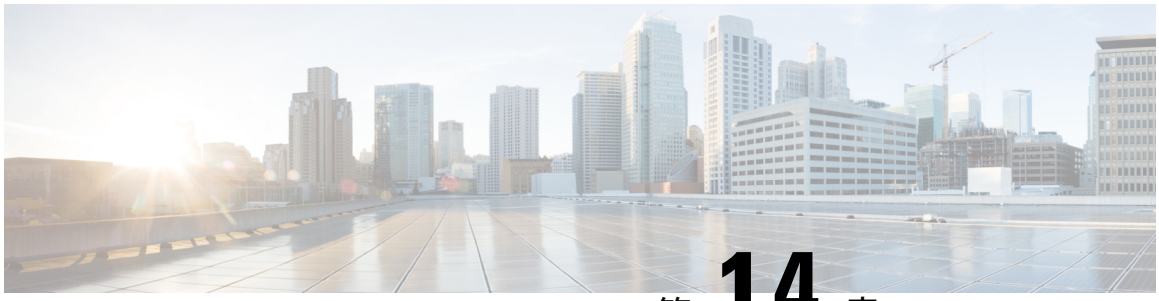
show ip bgp neighbors [*address*]	各ネイバーとの BGP 接続および TCP 接続に関する 表示します。
show ip bgp neighbors [*address*] [**advertised-routes** | **dampened-routes** | **flap-statistics** | **paths** *regular-expression* | **received-routes** | **routes**]	特定の BGP ネイバーから取得されたルートを表示
show ip bgp paths	データベース内のすべての BGP パスを表示します
show ip bgp peer-group [*tag*] [**summary**]	BGP ピア グループに関する情報を表示します。
show ip bgp summary	BGP 接続すべての状況を表示します。

bgp log-neighbor changes コマンドは、デフォルトでは有効です。そのため、BGP ネイバーのリセット、起動、またはダウン時に生成されるメッセージをログに記録できます。

ボーダー ゲートウェイ プロトコルの機能情報

表 *29 :* ボーダー ゲートウェイ プロトコルの機能情報

機能名	リリース	機能情報
ボーダー ゲートウェイ プロトコル	Cisco IOS XE Everest 16.5.1a	この機能が導入されました。
条件付き BGP ルートの挿入	Cisco IOS XE Gibraltar 16.11.1	この機能が導入されました。
BGP ピア テンプレート	Cisco IOS XE Gibraltar 16.11.1	この機能が導入されました。
BGP ルートマップ ネクスト ホップ セルフ	Cisco IOS XE Gibraltar 16.11.1	この機能が導入されました。



第 14 章

4 バイト ASN に対する BGP サポートの設定

4 バイト ASN に対する BGP サポートに関する情報

BGP 自律システム番号の形式

RFC 4271『*A Border Gateway Protocol 4 (BGP-4)*』に記述されているように、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（AS）番号は 1 ～ 65535 の範囲の 2 オクテットの数値でした。現在は、AS 番号の需要増加に伴い、Internet Assigned Numbers Authority（IANA）によって割り当てられる AS 番号は 65536 ～ 4294967295 の範囲の 4 オクテットの番号になりました。RFC 5396『*Textual Representation of Autonomous System (AS) Numbers*』には、AS 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 AS 番号をその 10 進数値で表します。たとえば、65526 は 2 バイト AS 番号、234567 は 4 バイト AS 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 AS 番号は 10 進数で、4 バイト AS 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 AS 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 AS 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

4 オクテット（4 バイト）の AS 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 AS 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。次の表は、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 AS 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 *30 : asdot* だけを使用する *4* バイト *AS* 番号形式

書式	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 ～ 655354 バイト：1.0 ～ 65535.65535	2 バイト：1 ～ 655354 バイト：1.0 ～ 65535.65535

asplain をデフォルトとする AS 番号形式

シスコ実装の 4 バイト AS 番号では asplain がデフォルトの AS 番号表示形式として使用されていますが、4 バイト AS 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 AS 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 AS 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** コマンド出力を変更して、4 バイトの自律システム番号を asdot 形式で表示する場合は、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 形式が有効にされている場合、正規表現の 4 バイト AS 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。次の表に示すように、4 バイト AS 番号は asplain と asdot のどちらにも設定できますが、**show** コマンド出力と正規表現を使用した 4 バイト AS 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 AS 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** コマンドを有効にした後、**clear ip bgp *** コマンドを入力してすべての BGP セッションに対してハード リセットを開始する必要があります。

（注） 4 バイト AS 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 AS 番号を使用できます。4 バイト AS 番号に設定された形式にかかわらず、2 バイト AS 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 *31 : asplain* をデフォルトとする *4* バイト *AS* 番号形式

書式	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 ～ 655354 バイト：65536 ～ 4294967295	2 バイト：1 ～ 655354 バイト：65536 ～ 4294967295

書式	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 ～ 655354 バイト：1.0 ～ 65535.65535	2 バイト：1 ～ 655354 バイト：65536 ～ 4294967295

表 *32 : asdot* を使用する *4* バイト *AS* 番号形式

書式	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 ～ 655354 バイト：65536 ～ 4294967295	2 バイト：1 ～ 655354 バイト：1.0 ～ 65535.65535
asdot	2 バイト：1 ～ 655354 バイト：1.0 ～ 65535.65535	2 バイト：1 ～ 655354 バイト：1.0 ～ 65535.65535

予約済みおよびプライベートの AS 番号

シスコが採用している BGP は、RFC 4893 をサポートしています。RFC 4893 は、2 バイト AS 番号から 4 バイト AS 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。新しい予約済み（プライベート）AS 番号（23456）は RFC 4893 により作成された番号で、Cisco IOS CLI ではこの番号を AS 番号として設定できません。

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 AS 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 AS 番号レジストリに記載されています。予約済み 2 バイト AS 番号は 64496 ～ 64511 の連続したブロック、予約済み 4 バイト AS 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 AS 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 AS 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 AS 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 AS 番号を削除しません。ISP がプライベート AS 番号をフィルタ処理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 AS 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 AS 番号の登録申込など、AS 番号に関する情報については、http://www.iana.org/ を参照してください。

シスコが採用している 4 バイト自律システム番号

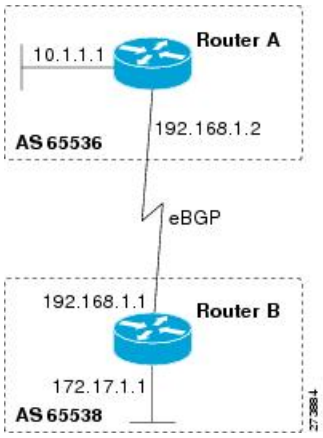
シスコが採用している 4 バイト自律システム（AS）番号は、AS 番号の正規表現のマッチングおよび出力表示形式のデフォルトとして asplain（たとえば、65538）を使用していますが、RFC 5396 に記載されているとおり、4 バイト AS 番号を asplain 形式および asdot 形式の両方で設定できます。4 バイト AS 番号の正規表現マッチングと出力表示のデフォルトを asdot 形式に変更するには、**bgp asnotation dot** コマンドの後に **clear ip bgp *** コマンドを実行し、現在の BGP セッションをすべてハード リセットします。4 バイト AS 番号形式の詳細については、「BGP 自律システム番号の形式」の項を参照してください。

シスコが採用している 4 バイト AS 番号は、設定形式、正規表現とのマッチング、および出力表示として、asdot（たとえば、1.2）だけを使用しています。asplain はサポートしていません。4 バイト番号を使用する 2 つの自律システム内の BGP ピアの例については、下の図を参照してください。asdot 表記法を使用して設定された、異なる 4 バイトの自律システムにある 3 つのネイバー ピアの間での設定例については、「例：BGP ルーティング プロセスと 4 バイト自律システム番号を使用したピアの設定」を参照してください。

シスコは、BGP が 2 バイト AS 番号から 4 バイト AS 番号へ段階的に移行できるように開発された RFC 4893 もサポートしています。スムーズな移行を確実に行うには、4 バイト AS 番号を使用して識別される AS 内の BGP スピーカーをすべて、4 バイト AS 番号をサポートするようにアップグレードすることを推奨します。

（注） 新しいプライベート AS 番号（23456）は RFC 4893 により作成された番号で、Cisco IOS CLI ではこの番号を AS 番号として設定できません。

図 *8 : 4* バイト番号を使用する *2* つの自律システム内の *BGP* ピア



4 バイト ASN に対する BGP サポートの設定方法

BGP ルーティング プロセスと 4 バイト自律システム番号を使用したピアの設定

4 バイト自律システム（AS）番号を使用する AS にボーダー ゲートウェイ プロトコル（BGP）ピアが配置されているときに、BGP ルーティング プロセスおよび BGP ピアを設定するには、この作業を実行します。ここで設定するアドレス ファミリは、デフォルトの IPv4 ユニキャスト アドレス ファミリで、設定は上の図（「シスコが採用している 4 バイト自律システム番号」の項）のルータ A で行われています。この作業にある 4 バイト AS 番号は、デフォルトの asplain（10 進数値）形式にフォーマットされています。たとえば、上の図にあるルータ B の AS 番号は 65538 です。BGP ピアとなりうるネイバー ルータすべてについて、必ず、この作業を実行してください。

始める前に

（注） デフォルトでは、ルータ コンフィギュレーション モードで **neighbor remote-as** コマンドを使用して定義したネイバーは、IPv4 ユニキャスト アドレス プレフィックスだけを交換します。IPv6 プレフィックスなど、その他のアドレス プレフィックス タイプを交換するには、そのプレフィックス タイプについて、アドレス ファミリ コンフィギュレーション モードで **neighbor activate** コマンドを使用してネイバーをアクティブ化する必要もあります。

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. 必要に応じて、手順 4 を繰り返し、その他の BGP ネイバーを定義します。
6. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
7. **neighbor** *ip-address* **activate**
8. 必要に応じて、手順 7 を繰り返し、その他の BGP ネイバーをアクティブ化します。
9. **network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*]
10. **end**
11. **show ip bgp** [*network*] [*network-mask*]
12. **show ip bgp summary**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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： `Device(config)# router bgp 65538`	指定したルーティング プロセスのルータ 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、4 バイト AS 番号 65538 は asplain 表記法で定義されています。
ステップ 4	**neighbor** *ip-address* **remote-as** *autonomous-system-number* 例： `Device(config-router)# neighbor 192.168.1.2 remote-as 65536`	指定された AS のネイバーの IP アドレスを、ローカル デバイスの 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、4 バイト AS 番号 65536 は asplain 表記法で定義されています。
ステップ 5	必要に応じて、手順 4 を繰り返し、その他の BGP ネイバーを定義します。	--
ステップ 6	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] 例： `Device(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、デバイスは 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 プレフィックスを指定します。 • **vrf** キーワードおよび *vrf-name* 引数では、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 Virtual Routing and Forwarding（VRF; 仮想ルーティングおよび転送）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	**neighbor** *ip-address* **activate** 例： `Device(config-router-af)# neighbor 192.168.1.2 activate`	ネイバーが IPv4 ユニキャスト アドレス ファミリのプレフィックスをローカル デバイスと交換できるようにします。
ステップ 8	必要に応じて、手順 7 を繰り返し、その他の BGP ネイバーをアクティブ化します。	--
ステップ 9	**network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*] 例： `Device(config-router-af)# network 172.17.1.0 mask 255.255.255.0`	（任意）この AS にローカルとしてネットワークを指定し、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決定します。
ステップ 10	**end** 例： `Device(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 モードに戻ります。
ステップ 11	**show ip bgp** [*network*] [*network-mask*] 例： `Device# show ip bgp 10.1.1.0`	（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。 （注） この例では、このタスクに適用可能な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 12	**show ip bgp summary** 例： `Device# show ip bgp summary`	（任意）BGP 接続すべての状況を表示します。

次の例は、上の図のルータ B で実行された **show ip bgp** コマンドの出力ですが、ここにはルータ A で 192.168.1.2 にある BGP ネイバーから学習されたネットワーク 10.1.1.0 に対する BGP ルーティング テーブル エントリと、デフォルトの asplain 形式で表した 4 バイト AS 番号 65536 が表示されています。

```
RouterB# show ip bgp 10.1.1.0

BGP routing table entry for 10.1.1.0/24, version 2
Paths: (1 available, best #1)
```

```
Advertised to update-groups:
2
65536
192.168.1.2 from 192.168.1.2 (10.1.1.99)
Origin IGP, metric 0, localpref 100, valid, external, best
```

次の例は、**show ip bgp summary** コマンドの出力ですが、ここには、上の図のルータ B でこの作業を設定した後で、ルータ A にある BGP ネイバー 192.168.1.2 の 4 バイト AS 番号が 65536 であることが表示されています。

```
RouterB# show ip bgp summary

BGP router identifier 172.17.1.99, local AS number 65538
BGP table version is 3, main routing table version 3
2 network entries using 234 bytes of memory
2 path entries using 104 bytes of memory
3/2 BGP path/bestpath attribute entries using 444 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 806 total bytes of memory
BGP activity 2/0 prefixes, 2/0 paths, scan interval 60 secs
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  Stated
192.168.1.2     4       65536       6       6        3    0    0 00:01:33        1
```

4 バイト自律システム番号で使用される出力および正規表現とのマッチング形式のデフォルトを変更

4 バイト自律システム（AS）番号のデフォルト出力形式を asplain 形式から asdot 表記法形式に変更するには、この作業を実行します。4 バイト AS 番号の出力形式の変化を表示するには、**show ip bgp summary** コマンドを使用します。

手順の概要

1. **enable**
2. **show ip bgp summary**
3. **configure terminal**
4. **router bgp** *autonomous-system-number*
5. **bgp asnotation dot**
6. **end**
7. **clear ip bgp ***
8. **show ip bgp summary**
9. **show ip bgp regexp** *regexp*
10. **configure terminal**
11. **router bgp** *autonomous-system-number*
12. **no bgp asnotation dot**
13. **end**
14. **clear ip bgp ***

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**show ip bgp summary** 例： `Device# show ip bgp summary`	すべてのボーダーゲートウェイ プロトコル（BGP）接続のステータスを表示します。
ステップ 3	**configure terminal** 例： `Device# 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 4	**router bgp** *autonomous-system-number* 例： `Device(config)# router bgp 65538`	指定したルーティング プロセスのルータ 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、4 バイト AS 番号 65538 は asplain 表記法で定義されています。
ステップ 5	**bgp asnotation dot** 例： `Device(config-router)# bgp asnotation dot`	BGP 4 バイト AS 番号のデフォルト出力形式を asplain（10 進数値）からドット表記法に変更します。 （注） 4 バイト AS 番号は、asplain 形式、または asdot 形式を使用して設定できます。このコマンドの影響を受けるのは、**show** コマンドの出力、または正規表現のマッチングだけです。
ステップ 6	**end** 例： `Device(config-router)# 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 モードに戻ります。
ステップ 7	**clear ip bgp *** 例： `Device# clear ip bgp *`	現在の BGP セッションをすべてクリアし、リセットします。 • この例では、4 バイト AS 番号形式の変更がすべての BGP セッションに反映されていることを確認するために、ハード リセットが実行されています。

	コマンドまたはアクション	目的
		（注） この例では、このタスクに適用可能な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	**show ip bgp summary** 例： `Device# show ip bgp summary`	BGP 接続すべての状況を表示します。
ステップ 9	**show ip bgp regexp** *regexp* 例： `Device# show ip bgp regexp ^1\.0$`	AS パスの正規表現と一致するルートを表示します。 • この例では、4 バイトの AS パスをマッチングする正規表現は、asdot 形式で設定されています。
ステップ 10	**configure terminal** 例： `Device# 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 11	**router bgp** *autonomous-system-number* 例： `Device(config)# router bgp 65538`	指定したルーティング プロセスのルータ 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、4 バイト AS 番号 65538 は asplain 表記法で定義されています。
ステップ 12	**no bgp asnotation dot** 例： `Device(config-router)# no bgp asnotation dot`	BGP 4 バイト AS 番号のデフォルト出力形式を asplain（10 進数値）にリセットします。 （注） 4 バイト AS 番号は、asplain 形式、または asdot 形式を使用して設定できます。このコマンドの影響を受けるのは、**show** コマンドの出力、または正規表現のマッチングだけです。
ステップ 13	**end** 例： `Device(config-router)# end`	ルータ コンフィギュレーション モードを終了して、特権 EXEC モードに戻ります。
ステップ 14	**clear ip bgp *** 例：	現在の BGP セッションをすべてクリアし、リセットします。

	コマンドまたはアクション	目的
	`Device# clear ip bgp *`	• この例では、4 バイト AS 番号形式の変更がすべての BGP セッションに反映されていることを確認するために、ハード リセットが実行されています。 （注） この例では、このタスクに適用可能な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。

例

次の **show ip bgp summary** コマンドの出力は、4 バイト AS 番号のデフォルト asplain 形式を示しています。ここで、asplain 形式で表された 4 バイト AS 番号 65536 および 65550 に注意してください。

```
Router# show ip bgp summary

BGP router identifier 172.17.1.99, local AS number 65538
BGP table version is 1, main routing table version 1
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  Statd
192.168.1.2     4        65536       7       7        1    0    0 00:03:04      0
192.168.3.2     4        65550       4       4        1    0    0 00:00:15      0
```

bgp asnotation dot コマンドの設定後（これに、現在の BGP セッションをすべてハード リセットする **clear ip bgp *** コマンドが続きます）、出力は、次の **show ip bgp summary** コマンドの出力に示すように、asdot 表記法の形式に変換されます。asdot 形式で表された 4 バイト AS 番号 1.0 および 1.14 に注意してください。これらは AS 番号 65536 と 65550 を asdot 変換したものです。

```
Router# show ip bgp summary

BGP router identifier 172.17.1.99, local AS number 1.2
BGP table version is 1, main routing table version 1
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  Statd
192.168.1.2     4          1.0       9       9        1    0    0 00:04:13      0
192.168.3.2     4         1.14       6       6        1    0    0 00:01:24      0
```

bgp asnotation dot コマンドの設定後（これに、現在の BGP セッションをすべてハード リセットする **clear ip bgp *** コマンドが続きます）、4 バイトの AS パスで使用される正規表現とのマッチング形式は asdot 表記法の形式に変更されます。4 バイト AS 番号は、asplain 形式または asdot 形式のいずれかを使用して、正規表現で設定できますが、現在のデフォルト形式を使用して設定された 4 バイト AS 番号だけがマッチングされます。下の先頭の例では、**show ip bgp regexp** コマンドは、asplain 形式で表された 4 バイト AS 番号を使って設定されています。現在のデフォルト形式は asdot 形式なのでマッチングは失敗し、何も出力されません。asdot 形式を使用した 2 番目の例では、

マッチングは成功し、4 バイトの AS パスに関する情報が asdot 表記法を使って表示されます。

（注） このasdot表記法で使用されているピリオドは、シスコの正規表現では特殊文字です。特殊な意味を取り除くには、ピリオドの前にバックスラッシュをつけます。

```
Router# show ip bgp regexp ^65536$

Router# show ip bgp regexp ^1\.0$

BGP table version is 2, 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 1.0 i
```

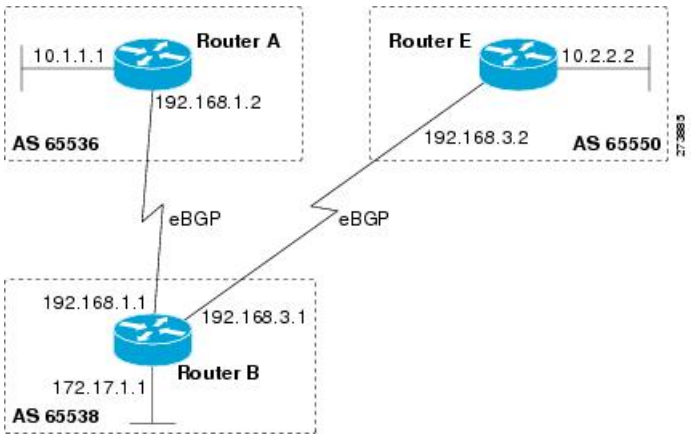
4 バイト ASN に対する BGP サポートの設定例

例：BGP ルーティング プロセスと 4 バイト自律システム番号を使用したピアの設定

asplain 形式

次に示すのは、下の図におけるボーダー ゲートウェイ プロトコル（BGP）プロセスを使ったルータ A、B、E のコンフィギュレーションの例で、このプロセスは、asplain 表記法を使用して設定された別々の 4 バイト自律システムのルータ A、B、E にある 3 つのネイバー ピアの間に設定されています。IPv4 ユニキャスト ルートはすべてのピアと交換されます。

図 *9 : asplain* 形式の *4* バイト自律システム番号を使用する *BGP* ピア



ルータ A

```
router bgp 65536
bgp router-id 10.1.1.99
no bgp default ipv4-unicast
bgp fast-external-fallover
bgp log-neighbor-changes
timers bgp 70 120
neighbor 192.168.1.1 remote-as 65538
!
address-family ipv4
neighbor 192.168.1.1 activate
no auto-summary
no synchronization
network 10.1.1.0 mask 255.255.255.0
exit-address-family
```

ルータ B

```
router bgp 65538
bgp router-id 172.17.1.99
no bgp default ipv4-unicast
bgp fast-external-fallover
bgp log-neighbor-changes
timers bgp 70 120
neighbor 192.168.1.2 remote-as 65536
neighbor 192.168.3.2 remote-as 65550
neighbor 192.168.3.2 description finance
!
address-family ipv4
neighbor 192.168.1.2 activate
neighbor 192.168.3.2 activate
no auto-summary
no synchronization
network 172.17.1.0 mask 255.255.255.0
exit-address-family
```

ルータ E

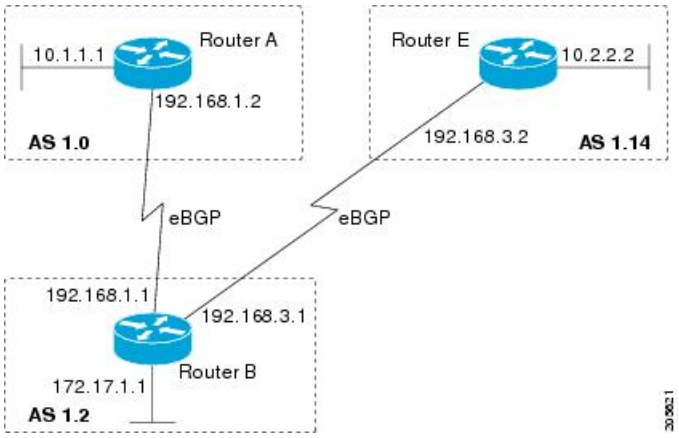
```
router bgp 65550
bgp router-id 10.2.2.99
no bgp default ipv4-unicast
bgp fast-external-fallover
bgp log-neighbor-changes
timers bgp 70 120
neighbor 192.168.3.1 remote-as 65538
!
address-family ipv4
neighbor 192.168.3.1 activate
no auto-summary
no synchronization
network 10.2.2.0 mask 255.255.255.0
exit-address-family
```

asdot 形式

次に示すのは、下の図における BGP プロセスを使ったルータ A、B、E のコンフィギュレーションを作成する方法の例で、このプロセスは、デフォルトの asdot 形式を使用して設定され

た別々の 4 バイト自律システムのルータ A、B、E にある 3 つのネイバー ピアの間に設定されています。IPv4 ユニキャスト ルートはすべてのピアと交換されます。

図 *10 : asdot* 形式の *4* バイト自律システム番号を使用する *BGP* ピア



ルータ A

```
router bgp 1.0
bgp router-id 10.1.1.99
no bgp default ipv4-unicast
bgp fast-external-fallover
bgp log-neighbor-changes
timers bgp 70 120
neighbor 192.168.1.1 remote-as 1.2
!
address-family ipv4
neighbor 192.168.1.1 activate
no auto-summary
no synchronization
network 10.1.1.0 mask 255.255.255.0
exit-address-family
```

ルータ B

```
router bgp 1.2
bgp router-id 172.17.1.99
no bgp default ipv4-unicast
bgp fast-external-fallover
bgp log-neighbor-changes
timers bgp 70 120
neighbor 192.168.1.2 remote-as 1.0
neighbor 192.168.3.2 remote-as 1.14
neighbor 192.168.3.2 description finance
!
address-family ipv4
neighbor 192.168.1.2 activate
neighbor 192.168.3.2 activate
no auto-summary
no synchronization
network 172.17.1.0 mask 255.255.255.0
exit-address-family
```

ルータ E

```
router bgp 1.14
bgp router-id 10.2.2.99
no bgp default ipv4-unicast
bgp fast-external-fallover
bgp log-neighbor-changes
timers bgp 70 120
neighbor 192.168.3.1 remote-as 1.2
!
address-family ipv4
neighbor 192.168.3.1 activate
no auto-summary
no synchronization
network 10.2.2.0 mask 255.255.255.0
exit-address-family
```

例：4バイトのBGP自律システム番号を使用したVRFおよび拡張コミュニティの設定

次に、4 バイト自律システム番号 65537 を使用するルートターゲットを使って VRF を作成する方法、およびルートターゲットに、ルートマップにより許可されたルートの拡張コミュニティ値 65537:100 を設定する例を示します。

```
ip vrf vpn_red
rd 64500:100
route-target both 65537:100
exit
route-map red_map permit 10
set extcommunity rt 65537:100
end
```

コンフィギュレーションの完了後、**show route-map** コマンドを使用して、拡張コミュニティが、4 バイト自律システム番号 65537 を含むルート ターゲットに設定されていることを確認します。

```
RouterB# show route-map red_map
route-map red_map, permit, sequence 10
Match clauses:
Set clauses:
extended community RT:65537:100
Policy routing matches: 0 packets, 0 bytes
```

4 バイト自律システム番号の RD サポート

次の例は、4 バイト AS 番号 65536 を含むルート識別子、および 4 バイト自律システム番号 65537 を含むルート ターゲットを使用して、VRF を作成する方法を示しています。

```
ip vrf vpn_red
rd 65536:100
route-target both 65537:100
exit
```

コンフィギュレーションの完了後、**show vrf** コマンドを使用して、4 バイト AS 番号ルート識別子が 65536:100 に設定されていることを確認します。

```
RouterB# show vrf vpn_red
Current configuration : 36 bytes
vrf definition x
rd 65536:100
!
```

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、4 バイト自律システム番号 1.1 を使用するルートターゲットを使って VRF を作成する方法、およびルートターゲットに、ルートマップにより許可されたルートの拡張コミュニティ値 1.1:100 を設定する例を示します。

（注） 次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です。

```
ip vrf vpn_red
rd 64500:100
route-target both 1.1:100
exit
route-map red_map permit 10
set extcommunity rt 1.1:100
end
```

コンフィギュレーションの完了後、**show route-map** コマンドを使用して、拡張コミュニティが、4 バイト自律システム番号 1.1 を含むルート ターゲットに設定されていることを確認します。

```
RouterB# show route-map red_map
route-map red_map, permit, sequence 10
Match clauses:
Set clauses:
extended community RT:1.1:100
Policy routing matches: 0 packets, 0 bytes
```

4 バイト自律システム番号の RD サポートの asdot デフォルト形式

次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です。

```
ip vrf vpn_red
rd 1.0:100
route-target both 1.1:100
exit
```

4 バイト ASN に対する BGP サポートに関する追加情報

関連資料

関連項目	マニュアル タイトル
BGP コマンド	『*Cisco IOS IP Routing: BGP Command Reference*』

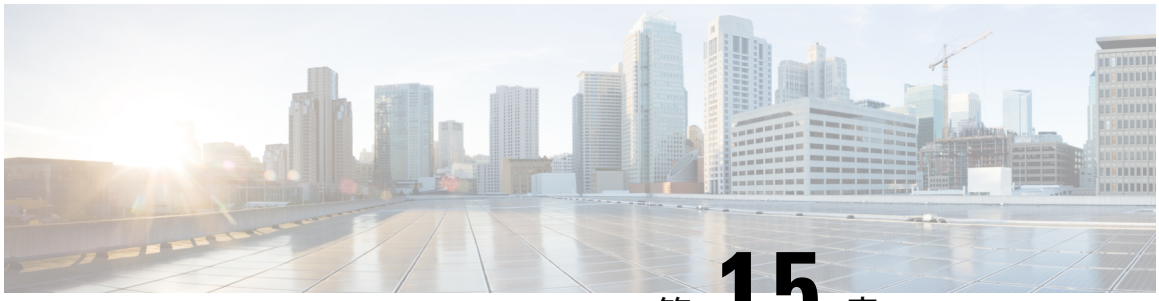
標準および RFC

標準/RFC	タイトル
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*』
RFC 5668	『*4-Octet AS Specific BGP Extended Community*』

4 バイト ASN に対する BGP サポートの機能履歴と機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。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。

リリース	変更内容
Cisco IOS XE Gibraltar 16.11.1	この機能が導入されました。



第 15 章

BGP ネクストホップ非変更の設定

外部 BGP（eBGP）セッションでは、デフォルトで、ルータがルートの送信時に BGP ルートのネクスト ホップ属性を（自身のアドレスに）変更します。BGP ネクスト ホップ非変更機能では、ネクスト ホップ属性を変更せずに BGP によって eBGP マルチホップ ピアにアップデートを送信できます。

BGP ネクストホップ非変更に関する制約事項

BGP ネクスト ホップ非変更機能は、マルチホップ eBGP ピア間だけで設定できます。直接接続されたネイバーにこの機能を設定しようとすると、次のエラー メッセージが表示されます。

```
%BGP: Can propagate the nexthop only to multi-hop EBGP neighbor
```

BGP ネクスト ホップ非変更

外部 BGP（eBGP）セッションでは、デフォルトで、ルータがルートの送信時に BGP ルートのネクスト ホップ属性を（自身のアドレスに）変更します。BGP ネクスト ホップ非変更機能が設定されている場合、BGP はネクスト ホップ属性を変更せずに eBGP マルチホップ ピアにルートを送信します。ネクスト ホップ属性は変更されません。

（注） ルータがルートを送信するとき、BGP ルートのネクストホップ属性を変更するルータのデフォルト動作の例外があります。ネクスト ホップが eBGP ピアのピアリング アドレスと同じサブネットにある場合、ネクスト ホップは変更されません。これは、サード パーティのネクスト ホップと呼ばれます。

BGP ネクスト ホップ非変更機能により、ネットワークの設計および移行を柔軟に実行できます。これは、マルチホップとして設定された eBGP ピア間だけで使用できます。2 つの自律システム間のさまざまなシナリオで使用できます。たとえば、同じ IGP を共有する複数の自律システムが接続される場合、または少なくともルータに互いのネクストホップに到達するための別の方法がある（このため、ネクストホップを変更しないままにできる）場合などが挙げられます。

この機能の一般的な用途は、RR 間で VPNv4 のマルチホップ MP-eBGP を持つマルチプロトコル ラベル スイッチング（MPLS）Inter-AS を設定することです。

この機能のもう 1 つの一般的な用途は、RFC4364、Section 10 で定義されている VPNv4 Inter-AS オプション C の設定です。この設定では、VPNv4 ルートは、自律システム間で（異なる自律システムの RR 間で）渡されます。RR は複数ホップ離れており、**neighbor next-hop unchanged** が設定されています。異なる自律システムの PE によって、その PE 間に LSP が確立されます（一般的な IGP 経由によって、または ASBR 間のラベル付きルート（1 ホップ離れた異なる自律システムからのルート）経由で PE に接続されたネクストホップのアドバタイズによって）。PE は、LSP 経由で別の AS 内の PE のネクスト ホップに到達でき、したがって VRF RIB に VPNv4 ルートをインストールできます。

BGP ネクスト ホップ非変更の設定方法

次の手順には、BGPネクストホップ非変更を設定する手順が含まれています。

eBGP ピアの BGP ネクスト ホップ非変更の設定

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family** {**ipv4** | **ipv6** | **l2vpn** | **nsap** | **rtfilter** | **vpnv4** | **vpnv6**}
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**
7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **ebgp-multihop** *ttl*
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **next-hop-unchanged**
9. **end**
10. **show ip bgp**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例：	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
	`Device> enable`	
ステップ 2	**configure terminal** 例： `Device# 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** *as-number* 例： `Device(config)# router bgp 65535`	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	**address-family** {**ipv4** | **ipv6** | **l2vpn** | **nsap** | **rtfilter** | **vpnv4** | **vpnv6**} 例： `Device(config-router-af)# address-family vpnv4`	アドレス ファミリ コンフィギュレーション モードを開始して、アドレス ファミリ固有の設定を受け入れるように BGP ピアを設定します。
ステップ 5	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* 例： `Device(config-router-af)# neighbor 10.0.0.100 remote-as 65600`	エントリを BGP ネイバー テーブルに追加します。
ステップ 6	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate** 例： `Device(config-router-af)# neighbor 10.0.0.100 activate`	ピアとの情報交換をイネーブルにします。
ステップ 7	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **ebgp-multihop** *ttl* 例： `Device(config-router-af)# neighbor 10.0.0.100 ebgp-multihop 255`	ローカル ルータを設定して、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との接続を受け入れて開始するようにします。
ステップ 8	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **next-hop-unchanged** 例： `Device(config-router-af)# neighbor 10.0.0.100 next-hop-unchanged`	ネクスト ホップ属性を変更せずに指定された eBGP ピアに BGP アップデートを送信するようにルータを設定します。

	コマンドまたはアクション	目的
ステップ 9	**end** 例： `Device(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 10	**show ip bgp** 例： `Device# show ip bgp`	（任意）BGP ルーティング テーブルのエントリを表示します。 • 出力には、選択されたアドレスについて **neighbor next-hop-unchanged** コマンドが設定されているかどうかが示されます。

ルートマップを使用した BGP ネクスト ホップ非変更の設定

eBGP ネイバーに対する発信ルートマップの設定

ルートマップを定義し、ネイバーに対する発信ポリシーを適用するには、**set ip next-hop unchanged** コマンドを使用します。

次の設定では、プレフィックス 1.1.1.1 のネクストホップは eBGP ネイバー 15.1.1.2 への送信時に変更されません。

```
enable
config terminal
router bgp 2
 bgp log-neighbor-changes
 neighbor 15.1.1.2 remote-as 3
 neighbor 15.1.1.2 ebgp-multihop 10
 !
 address-family ipv4
  neighbor 15.1.1.2 activate
  neighbor 15.1.1.2 route-map A out
  exit address-family
!
route-map A permit 10
 match ip address 1
 set ip next-hop unchanged
!
access-list 1 permit 1.1.1.1
end
```

eBGP ネイバーへの送信時における iBGP および eBGP パス プレフィックスのネクストホップ非変更の設定

eBGP ネイバーへの送信時に iBGP および eBGP パス プレフィックスのネクストホップを変更しないよう設定するには、**next-hop-unchanged allpaths** コマンドを使用します。

次の設定では、iBGP パス プレフィックスでも eBGP パス プレフィックスでも、ネクストホップは eBGP ネイバー 15.1.1.2 への送信時に変更されません。

```
enable
config terminal
router bgp 2
 bgp log-neighbor-changes
 neighbor 15.1.1.2 remote-as 3
 neighbor 15.1.1.2 ebgp-multihop 10
!
address-family ipv4
 neighbor 15.1.1.2 activate
 neighbor 15.1.1.2 next-hop-unchanged allpaths
 exit address-family
!
end
```

例：eBGP ピアの BGP ネクスト ホップ非変更

次に、リモート AS にマルチホップ eBGP ピア 10.0.0.100 を設定する例を示します。ローカル ルータがそのピアにアップデートを送信する場合、ネクスト ホップ属性を変更せずにアップデートを送信します。

```
router bgp 65535
 address-family ipv4
 neighbor 10.0.0.100 remote-as 65600
 neighbor 10.0.0.100 activate
 neighbor 10.0.0.100 ebgp-multihop 255
 neighbor 10.0.0.100 next-hop-unchanged
 end
```

（注） IPv4、IPv6、VPNv4、VPNv6、L2VPN など、すべてのアドレス ファミリが **next-hop unchanged** コマンドをサポートしています。ただし、アドレス ファミリ L2VPN BGP VPLS シグナリングについては、正常に機能させるためには **next-hop self** コマンドを使用する必要があります。

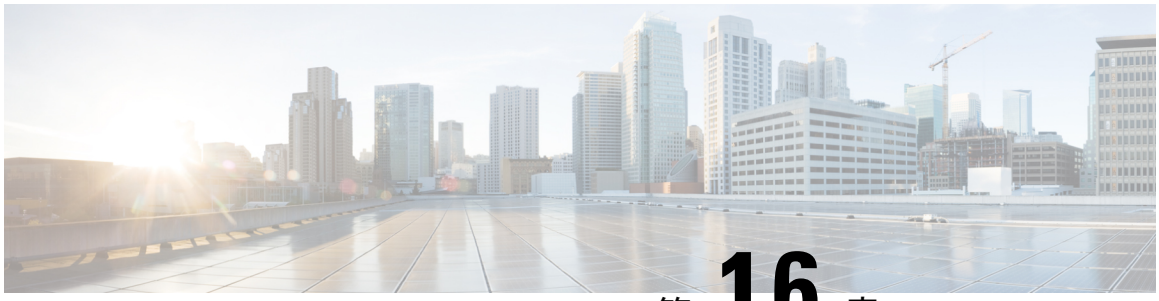
BGP ネクスト ホップ非変更機能の情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。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。

プラットフォームのサポートおよびシスコ 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 にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 *33 : BGP* ネクスト ホップ非変更機能の情報

機能名	リリース	機能情報
BGP ネクスト ホップ非変更	Cisco IOS XE Gibraltar 16.11.1	BGP ネクスト ホップ非変更機能では、ネクストホップ属性を変更せずにBGPによってeBGPマルチホップ ピアにアップデートを送信できます。



第 16 章

IS-IS ルーティングの設定

IS-IS ルーティングに関する情報

Integrated Intermediate System-to-Intermediate System（IS-IS）は、ISO ダイナミック ルーティング プロトコルの一つです（ISO 105890 を参照）。IS-IS をイネーブルするには、IS-IS ルーティングプロセスを作成し、それをネットワークではなく特定のインターフェイスに割り当てる必要があります。マルチエリア IS-IS コンフィギュレーション シンタックスを使用することで、レイヤ 3 デバイスごとに複数の IS-IS ルーティングプロトコルを指定できます。その後、IS-IS ルーティングプロセスのインスタンスごとにパラメータを設定する必要があります。

小規模の IS-IS ネットワークは、ネットワーク内にすべてのデバイスが含まれる単一のエリアとして構築されます。このネットワークは、その規模が大きくなるにしたがって、ローカルエリアに接続されたままの、接続済みのレベル 2 デバイスのセットで構成されるバックボーンエリア内に再編成されます。ローカル エリアの内部では、デバイスがすべてのシステム ID に到達する方法を認識しています。エリア間では、デバイスはバックボーンへの到達方法を認識しており、バックボーン デバイスは他のエリアに到達する方法を認識しています。

デバイスは、ローカルエリア内でルーティングを実行するために、レベル 1 の隣接関係を確立します（ステーションルーティング）。デバイスは、レベル 2 隣接関係を確立して、レベル 1 エリア間でルーティングを実行します（エリアルーティング）。

1 つの Cisco デバイスは、最大 29 エリアのルーティングに参加でき、バックボーンでレベル 2 ルーティングを実行できます。一般に、ルーティング プロセスごとに 1 つのエリアに対応します。デフォルトでは、設定されているルーティングプロセスの最初のインスタンスが、レベル 1 ルーティングと レベル 2 ルーティングの両方を実行します。追加のデバイスインスタンスを設定できます。このインスタンスは、自動的にレベル 1 エリアとして扱われます。IS-IS ルーティング プロセスの各インスタンスごとに個別にパラメータを設定する必要があります。

IS-IS マルチエリア ルーティングでは、シスコの各装置に対して最大 29 個の レベル 1 エリアを定義できますが、レベル 2 ルーティングを実行するプロセスは 1 つだけ設定できます。レベル2ルーティングが任意のプロセス上に設定されている場合、追加のプロセスは、すべて自動的にレベル 1 に設定されます。同時に、このプロセスがレベル 1 ルーティングを実行するように設定することもできます。デバイスインスタンスにレベル2ルーティングが必要でない場合は、 グローバル コンフィギュレーション モードで **is-type** コマンドを使用してレベル 2 の機能を削除します。別のデバイスインスタンスをレベル2デバイスとして設定する場合にも **is-type** コマンドを使用します。

IS-IS 認証

無許可のデバイスがリンクステートデータベースに誤ったルーティング情報を挿入することを防ぐために、インターフェイスごとにプレーンテキストのパスワードを設定するとともにIS-IS エリアごとにエリアパスワードを設定するか、IS-IS 認証を設定することができます。

プレーンテキストのパスワードは、無許可のユーザーに対するセキュリティを提供しません。プレーンテキストのパスワードを設定すると、無許可のネットワーキングデバイスがルータと隣接関係を形成することを防ぐことができます。このパスワードはプレーンテキストで交換されるため、アクセスして IS-IS パケットを表示できるエージェントによって参照されます。

新しい IS-IS 認証方式には、プレーンテキストパスワード設定コマンドに比べて次のような利点があります。

- ソフトウェア設定が表示されるときにパスワードが暗号化されます。
- パスワードの管理や変更がより容易になります。
- ネットワークの運用を中断させることなく、新しいパスワードに変更できます。
- 中断なしで認証を移行できます。

認証モード（IS-IS 認証またはプレーンテキストパスワード）は、特定の範囲（IS-IS インスタンスもしくはインターフェイス）またはレベルのいずれかで設定できますが、両方を設定することはできません。ただし、異なる範囲およびレベルに対して、異なるモードを設定することができます。混合モードが設定されている場合は、異なるモードには異なるキーを使用して、プロトコルデータユニット（PDU）で暗号化されたパスワードが危険にさらされないようにする必要があります。

クリアテキスト認証

IS-IS クリアテキスト認証は **area-password** コマンドまたは **domain-password** コマンドによって提供される機能と同じ機能を提供します。

HMAC-MD5 認証

IS-IS は、クリアテキスト認証より安全性の高いメッセージダイジェストアルゴリズム 5（MD5）認証をサポートしています。

ハッシュメッセージ認証コード（HMAC）は暗号学的ハッシュ関数を使用するメッセージ認証符号（MAC）のためのメカニズムです。HMAC-MD5 認証では、各 IS-IS PDU に HMAC-MD5 ダイジェストを追加します。ダイジェストによって、不正なルーティング メッセージがネットワーク ルーティング ドメインに入り込むのを防御できるため、IS-IS ルーティング プロトコル レベルでの認証が可能になります。

HMAC-MD5 認証の利点は次のとおりです。

- パスワードは、ルーティングメッセージを中断させずに新しいパスワードに変更できます。
- 中断なしで認証を移行できます。デバイスは、認証情報のない PDU や古い認証情報を持つ PDU を受け入れ、現在の認証情報を持つ PDU を送信します。このような移行は、認証なしの状態からあるタイプの認証に移行するとき、認証タイプを変更するとき、また認証キーを変更するときに便利です。

HMAC-SHA 認証

IS-IS では、MD5 認証またはクリアテキスト認証よりも安全性の高いセキュア ハッシュ アルゴリズム（SHA）認証（SHA-1、SHA-256、SHA-384、および SHA-512）がサポートされています。

HMAC-SHA 認証方式を有効にすると、共通ネットワークに接続されているすべてのデバイスで共有秘密キーが設定されます。各パケットでは、このキーを使用して、パケットに追加されるメッセージダイジェストを生成および検証します。メッセージダイジェストはパケットおよび秘密キーの単方向機能です。

ヒットレス アップグレード

使用するセキュリティ認証をあるタイプから別のタイプに移行する前に、次の手順を実行する必要があります。

1. すべてのデバイスに、その新しい認証タイプをサポートする新しいイメージをロードする必要があります。デバイスは、すべてのデバイスが新しい認証方式をサポートする新しいイメージでロードされ、さらにすべてのデバイスがその新しい認証方式を使用するように設定されるまで、元の認証方式を使用し続けます。
2. 現在のキーと新しいキーの両方を含むキーチェーンを追加します。たとえば、HMAC-MD5 から HMAC-SHA1-20 に移行する場合、現在のキーは HMAC-MD5 であり、新しいキーは HMAC-SHA1-20 です。IS-IS が現在のキーを送信しつづけるように、現在のキーが新しいキーよりも send-lifetime フィールドの終了日が遅いことを確認してください。IS-IS が両方のキーを受け入れるように、両方のキーの accept-lifetime 値を infinite に設定してください。
3. 手順 2 が完了したら、リンクまたはエリア内のすべてのデバイスについて、現在のキーをキーチェーンから削除できます。

NSF 認識

統合型 IS-IS ノンストップ フォワーディング（NSF）認識機能は IPv4G でサポートされています。この機能により、NSF を認識する顧客宅内機器（CPE）デバイスが、NFS 対応デバイスによるパケットのノンストップ フォワーディングを実現します。ローカルデバイスでは、必ずしも NSF を実行している必要はありませんが、その NSF を認識機能により、スイッチオーバープロセス時にルーティングデータベースの完全性と精度、および隣接 NSF 対応デバイス上のリンクステートデータベースが保持できます。

統合型 IS-IS ノンストップ フォワーディング（NSF）認識機能は自動的に有効になり、設定は不要です。

IS-IS グローバル パラメータ

次に、設定可能なオプションの IS-IS グローバルパラメータを示します。

- ルートマップによって制御されるデフォルトルートを設定することで、デフォルトルートを IS-IS ルーティングドメイン内に強制的に設定できます。ルートマップで設定可能な、その他のフィルタリングオプションも指定できます。
- 内部チェックサムエラーとともに受信された IS-IS リンクステートパケット（LSP）を無視したり、破損した LSP を消去するようにデバイスを設定できます。これにより、LSP の発信側は、LSP を再生成します。
- エリアおよびドメインにパスワードを割り当てられます。
- ルーティングテーブルでサマリーアドレスによって表される（経路集約に基づいた）集約アドレスを作成できます。他のルーティングプロトコルから学習したルートも集約できます。サマリーをアドバタイズするのに使用されるメトリックは、すべての個別ルートにおける最小のメトリックです。
- 過負荷ビットを設定できます。
- LSP リフレッシュインターバルおよび LSP がリフレッシュなしでデバイスデータベース内にとどまることができる最大時間を設定できます。
- LSP 生成に対するスロットリング タイマー、最短パス優先計算、および部分ルート計算を設定できます。
- IS-IS 隣接関係（アジャセンシー）がステートを変更（アップまたはダウン）する際に、デバイスがログメッセージを生成するように設定できます。
- ネットワーク内のリンクが、1500 バイト未満の最大伝送ユニット（MTU）サイズの場合、それでもルーティングが行われるように LSP MTU の値を低くできます。
- **partition avoidance** コマンドを使用して、レベル 1-2 境界デバイス、隣接レベル 1 デバイス、およびエンドホスト間で完全な接続が失われた場合に、エリアがパーティション化されるのを防ぐことができます。

IS-IS インターフェイス パラメータ

任意で、特定のインターフェイス固有の IS-IS パラメータを、付加されている他のデバイスとは別に設定できます。ただし、デフォルト値（乗数およびタイムインターバルなど）を変更する場合、複数のデバイスおよびインターフェイス上でもこれを変更する必要があります。ほとんどのインターフェイスパラメータは、レベル1、レベル2、またはその両方で設定できます。

設定可能なインターフェイスレベルのパラメータは次のとおりです。

- インターフェイスのデフォルトメトリック：Quality of Service（QoS）ルーティングが実行されない場合に、IS-IS メトリックの値として使用され、割り当てられます。
- helloインターバル（インターフェイスから送信されるhelloパケットの間隔）またはデフォルトのhelloパケット乗数：インターフェイス上で使用されて、IS-IS helloパケットで送信されるホールド タイムを決定します。ホールド タイムは、ネイバーがダウンしていると宣言するまでに、別の hello パケットを待機する時間を決定します。これにより、障害リンクまたはネイバーが検出される速さも決定し、ルートを再計算できるようになります。hello パケットが頻繁に失われ、IS-IS 隣接に無用な障害が発生する場合は、hello 乗数を変更してください。hello 乗数を大きくし、それに対応して hello インターバルを小さくすると、リンク障害を検出するのに必要な時間を増やすことなく、helloプロトコルの信頼性を高めることができます。
- その他のタイム インターバル：
  - Complete Sequence Number PDU（CSNP）インターバル：CSNP は、データベースの同期を維持するために指定デバイスによって送信されます。
  - 再送信インターバル：これは、ポイントツーポイントリンクの IS-IS LSP の再送信間隔です。
  - IS-IS LSP 再送信スロットルインターバル：これは、IS-IS LSP がポイントツーポイントリンク上で再送信される最大レート（パケット間のミリ秒数）です。この間隔は、同じ LSP の連続した再送信の間隔である再送信インターバルとは異なります。
- 指定デバイスの選択の優先順位：マルチアクセスネットワークで必要な隣接数を削減し、その代わりに、ルーティング プロトコル トラフィックの量およびトポロジデータベースのサイズを削減できます。
- インターフェイス回線タイプ：指定されたインターフェイス上のネイバーに必要な隣接タイプです。
- インターフェイスのパスワード認証。

IS-IS の設定方法

ここでは、インターフェイスで IS-IS を有効にする方法、IS-IS グローバルパラメータを設定する方法、および IS-IS インターフェイスパラメータを設定する方法について説明します。

IS-IS のデフォルト設定

表 *34 : IS-IS* のデフォルト設定

機能	デフォルト設定
リンクステート PDU（LSP）エラーを無視	イネーブル。
IS-IS タイプ	従来型の IS-IS：ルータは、レベル 1（ステーション）とレベア）両方のルータとして機能します。 マルチエリア IS-IS：IS-IS ルーティングプロセスの最初のインレベル 1-2 ルータです。残りのインスタンスは、レベル 1 ル
デフォルト情報送信元	ディセーブル。
IS-IS 隣接関係のステート変更を記録	ディセーブル。
LSP 生成スロットリング タイマー	連続した 2 つのオカレンス間の最大インターバル：5000 ミリ 初期 LSP 生成遅延：50 ミリ秒 最初と 2 番目の LSP 生成の間のホールド時間：200 ミリ秒
LSP 最大ライフ タイム（リフレッシュなし）	LSP パケットが削除されるまで 1200 秒（20 分）
LSP リフレッシュ インターバル	900 秒（15 分）ごと
最大 LSP パケット サイズ	1497 バイト
NSF 認識	イネーブル。レイヤ3デバイスでは、ハードウェアやソフト中に、隣接するノンストップ フォワーディング対応ルータがトを転送し続けることができます。
部分ルート計算（PRC）スロットリング タイマー	最大 PRC 待機インターバル：5000 ミリ秒 トポロジの変更後の初期 PRC 計算遅延：50 ミリ秒 最初と 2 番目の PRC 計算の間のホールド時間：200 ミリ秒
パーティション回避	ディセーブル。
パスワード	エリアまたはドメインのパスワードが定義されておらず、認証ブルになっています。
過負荷ビットの設定	ディセーブル。有効の際に引数が入力されない場合、過負荷ビちに設定され、**no set-overload-bit** コマンドが入力されるまでままになります。

機能	デフォルト設定
Shortest Path First（SPF）スロットリング タイマー	連続した SFP 間の最大インターバル：5000 ミリ秒 トポロジの変更後の初期 SFP 計算：200 ミリ秒 最初と 2 番目の SFP 計算の間のホールド時間：50 ミリ秒
サマリー アドレス	ディセーブル

IS-IS ルーティングのイネーブル化

IS-IS をイネーブルにするには、各ルーティングプロセスに名前とネットワーク エンティティ タイトル（NET）を指定します。インターフェイス上でIS-ISルーティングをイネーブルにし、ルーティングプロセスの各インスタンスに対してエリアを指定します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 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： `Device#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	**clns routing** 例： `Device(config)#clns routing`	デバイス上で ISO コネクションレス型ルーティングをイネーブルに設定します。
ステップ 4	**router isis** [*area tag*] 例： `Device(config)#router isis tag1`	指定したルーティングプロセスに対して IS-IS ルーティングをイネーブルにし、IS-IS ルーティング コンフィギュレーション モードを開始します。 （任意）*area tag* 引数を使用して、IS-IS ルータが割り当てられているエリアを特定します。複数の IS-IS エリアを設定する場合は、値を入力します。 最初に設定された IS-IS インスタンスは、デフォルトでレベル 1-2 です。後のインスタンスは、自動的にレベル 1 に設定されます。グローバル コンフィギュレーション モードで **is-type** コマンドを使用してルーティングのレベルを変更できます。

	コマンドまたはアクション	目的
ステップ 5	**net** *network-entity-title* 例： `Device(config-router)#net 47.0004.004d.0001.0001.0c11.1111.00`	ルーティング プロセスに NET を設定します。マルチエリア IS-IS を設定する場合は、各ルーティング プロセスに NET を指定します。NET およびアドレスの名前を指定します。
ステップ 6	**is-type** {**level-1** | **level-1-2** | **level-2-only**} 例： `Device(config-router)#is-type level-2-only`	（任意）レベル 1（ステーション）ルータ、マルチエリアルーティング用のレベル 2（エリア）ルータ、または両方（デフォルト）として機能するようにルータを設定します。 • **level 1**：ステーションルータとしてだけ機能します。 • **level 1-2**：ステーションルータおよびエリアルータの両方として機能します。 • **level 2**：エリアルータとしてだけ機能します。
ステップ 7	**exit** 例： `Device(config-router)#end`	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 8	**interface** *interface-id* 例： `Device(config)#interface gigabitethernet 1/0/1`	IS-IS をルーティングするインターフェイスを指定し、インターフェイスコンフィギュレーションモードを開始します。インターフェイスがまだレイヤ 3 インターフェイスとして設定されていない場合は、**no switchport** コマンドを入力してインターフェイスをレイヤ 3 モードに設定します。
ステップ 9	**ip router isis** [*area tag*] 例： `Device(config-if)#ip router isis tag1`	インターフェイスに IS-IS ルーティングプロセスを設定し、エリア指示子をルーティングプロセスに割り当てます。
ステップ 10	**ip address** *ip-address-mask* 例： `Device(config-if)#ip address 10.0.0.5 255.255.255.0`	インターフェイスの IP アドレスを定義します。インターフェイスのいずれかで IS-IS ルーティングが設定されている場合は、IS-IS がイネーブルになっているエリアに含まれるすべてのインターフェイスに IP アドレスが必要です。
ステップ 11	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
ステップ 12	**show isis** [*area tag*] **database detail** 例： `Device#show isis database detail`	入力を確認します。

IS-IS グローバル パラメータの設定

グローバル IS-IS パラメータを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 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： `Device#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 isis** 例： `Device(config)#router isis`	IS-IS ルーティング プロトコルを指定し、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	**default-information originate** [**route-map** *map-name*] 例： `Device(config-router)#default-information originate route-map map1`	（任意）デフォルト ルートを IS-IS ルーティング ドメインに強制的に設定します。**route-map** *map-name* コマンドを入力すると、にルーティング ロセスによって有効なルートマップのデフォルト ルートが生成されます。
ステップ 5	**ignore-lsp-errors** 例： `Device(config-router)#ignore-lsp-errors`	（任意）LSP を消去する代わりに、内部チェックサムにエラーがある LSP を無視するようにデバイスを設定します。このコマンドは、デフォルトでイネーブルになっています（破損した LSP はドロップされます）。破損した LSP を消去するには、ルータ コンフィギュレーション モードで **no ignore-lsp-errors** コマンドを入力します。

	コマンドまたはアクション	目的
ステップ 6	**area-password** *password* 例： `Device(config-router)#area-password 1password`	（任意）レベル 1（ステーションルータレベル）LSP に挿入されるエリア認証パスワードを設定します。
ステップ 7	**domain-password** *password* 例： `Device(config-router)#domain-password 2password`	（任意）レベル 2（エリアルータレベル）LSP に挿入されるルーティングドメイン認証パスワードを設定します。
ステップ 8	**summary-address** *address mask* [**level-1** | **level-1-2** | **level-2**] 例： `Device(config-router)#summary-address 10.1.0.0 255.255.0.0 level-2`	（任意）所定のレベルのアドレスのサマリーを作成します。
ステップ 9	**set-overload-bit** [**on-startup** {*seconds* | **wait-for-bgp**}] 例： `Device(config-router)#set-overload-bit on-startup wait-for-bgp`	（任意）デバイスに問題がある場合に、他のデバイスが最短パス優先（SPF）計算でこのデバイスを無視するように過負荷ビットを設定します。 • （任意）**on-startup**：スタートアップ時だけ過負荷ビットを設定します。**on-startup** が指定されない場合、過負荷ビットが即座に設定され、**no set-overload-bit** コマンドを入力するまで設定されたままになります。**on-startup** が指定されている場合は、秒数または **wait-for-bgp** のどちらかを入力する必要があります。 • *seconds*：**on-startup** キーワードが設定されている場合、システム起動時に過負荷ビットが設定されて、指定した秒数の間設定されたままになります。指定できる範囲は 5 ～ 86400 秒です。 • **wait-for-bgp**：**on-startup** キーワードが設定されている場合、過負荷ビットがシステム起動時に設定され、BGP が収束するまで設定されたままになります。BGP が収束されたことが IS-IS に通知されない場合、IS-IS は 10 分後に過負荷ビットをオフにします。
ステップ 10	**lsp-refresh-interval** *seconds* 例： `Device(config-router)#lsp-refresh-interval 1080`	（任意）LSP リフレッシュインターバル（秒）を設定します。範囲は 1 ～ 65535 秒です。デフォルトでは、LSP リフレッシュを 900 秒（15 分）ごとに送信します。

	コマンドまたはアクション	目的
ステップ 11	**max-lsp-lifetime** *seconds* 例： `Device(config-router)#max-lsp-lifetime 1000`	（任意）LSPパケットがリフレッシュされずにルータ データベース内に存続する最大時間を設定します。範囲は1～65535秒です。デフォルト値は1200秒（20分）です。指定された時間間隔のあと、LSPパケットは削除されます。
ステップ 12	**lsp-gen-interval** [**level-1** | **level-2**] *lsp-max-wait* [*lsp-initial-wait lsp-second-wait*] 例： `Device(config-router)#lsp-gen-interval level-2 2 50 100`	（任意）IS-IS 生成スロットリング タイマーを設定します。 • *lsp-max-wait*：生成される LAP の連続した 2 つのオカレンス間の最大インターバル（ミリ秒）。指定できる範囲は1～120 ミリ秒です。デフォルト値は 5000 ミリ秒です。 • *lsp-initial-wait*：最初の LSP 生成遅延（ミリ秒）。指定できる範囲は 1 ～ 10000 ミリ秒です。デフォルト値は 50 ミリ秒です。 • *lsp-second-wait*：最初と 2 番目の LSP 生成間（ミリ秒）のホールド時間。指定できる範囲は 1 ～ 10000 ミリ秒です。デフォルト値は 200 ミリ秒です。
ステップ 13	**spf-interval** [**level-1** | **level-2**] *spf-max-wait* [*spf-initial-wait spf-second-wait*] 例： `Device(config-router)#spf-interval level-2 5 10 20`	（任意）IS-IS SPF スロットリングタイマーを設定します。 • *spf-max-wait*：連続する SFP 間（ミリ秒）の最大インターバル。指定できる範囲は1～120 ミリ秒です。デフォルト値は 5000 ミリ秒です。 • *spf-initial-wait*：トポロジ変更後の最初の SFP 計算（ミリ秒）。指定できる範囲は 1 ～ 10000 ミリ秒です。デフォルト値は 50 ミリ秒です。 • *spf-second-wait*：最初と 2 番目の SFP 計算間（ミリ秒）のホールド時間。指定できる範囲は1～10000 ミリ秒です。デフォルト値は 200 ミリ秒です。
ステップ 14	**prc-interval** *prc-max-wait* [*prc-initial-wait prc-second-wait*] 例： `Device(config-router)#prc-interval 5 10 20`	（任意）IS-IS PRC スロットリングタイマーを設定します。 • *prc-max-wait*：2 つの連続する PRC 計算間の最大インターバル（ミリ秒）。指定できる範囲は 1 ～ 120 ミリ秒です。デフォルト値は 5000 ミリ秒です。

	コマンドまたはアクション	目的
		• *prc-initial-wait*：トポロジ変更後の最初の PRC 計算遅延（ミリ秒）。指定できる範囲は 1 ～ 10,000 ミリ秒です。デフォルト値は 50 ミリ秒です。 • *prc-second-wait*：最初と 2 番目の PRC 計算間（ミリ秒）のホールド タイム。指定できる範囲は 1 ～ 10,000 ミリ秒です。デフォルト値は 200 ミリ秒です。
ステップ 15	**log-adjacency-changes** [**all**] 例： `Device(config-router)#log-adjacency-changes all`	（任意）IS-IS 隣接ステート変更をログするようルータを設定します。End System-to-Intermediate System PDU および LSP など、IS-IS hello に関連しないイベントにより生成されたすべての変更をログに含めるには、**all** を入力します。
ステップ 16	**lsp-mtu** *size* 例： `Device(config-router)#lsp mtu 1560`	（任意）最大 LSP パケットサイズ（バイト）を指定します。指定できる範囲は 128 ～ 4352 バイトです。デフォルト値は 1497 バイトです。 （注） ネットワーク内のリンクで MTU サイズが縮小された場合、ネットワーク内のすべてのデバイスで LSP MTU サイズを変更する必要があります。
ステップ 17	**partition avoidance** 例： `Device(config-router)#partition avoidance`	（任意）境界ルータ、すべての隣接レベル 1 ルータ、およびエンド ホスト間で、フル接続が切断された場合、IS-IS レベル 1-2 境界ルータがレベル 1 エリア プレフィックスをレベル 2 バックボーンにアドバタイズしないようにします。
ステップ 18	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。

IS-IS インターフェイス パラメータの設定

IS-IS インターフェイス固有のパラメータを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 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： `Device#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	**interface** *interface-id* 例： `Device(config)#interface gigabitethernet 1/0/1`	設定するインターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。インターフェイスがまだレイヤ 3 インターフェイスとして設定されていない場合は、**no switchport** コマンドを入力してインターフェイスをレイヤ 3 モードに設定します。
ステップ 4	**isis metric** *default-metric* [**level-1** | **level-2**] 例： `Device(config-if)#isis metric 15`	（任意）指定したインターフェイスにメトリック（またはコスト）を設定します。指定できる範囲は 0 ～ 63 です。デフォルトは 10 です。レベルが入力されない場合は、レベル 1 ルータとレベル 2 ルータの両方にデフォルト値が適用されます。
ステップ 5	**isis hello-interval** {*seconds* | **minimal**} [**level-1** | **level-2**] 例： `Device(config-if)#isis hello-interval minimal`	（任意）デバイスが hello パケットを送信する間隔を指定します。デフォルトでは、hello インターバル *seconds* の 3 倍の値が、送信される hello パケットの *holdtime* としてアドバタイズされます。hello インターバルが狭まると、トポロジ変更の検出も速くなりますが、ルーティング トラフィック量は増大します。 • **minimal**：結果として得られるホールドタイムが 1 秒になるように、hello 乗数に基づいて hello 間隔が計算されます。 • *seconds*：指定できる範囲は 1 ー 65535 です。デフォルトは 10 秒です。
ステップ 6	**isis hello-multiplier** *multiplier* [**level-1** | **level-2**] 例： `Device(config-if)#isis hello-multiplier 5`	（任意）ネイバーが見落とすことができる IS-IS hello パケット数の最大値を指定します。見落とされたパケット数がこの値を超えると、デバイスは隣接がダウンしていると宣言します。指定できる範囲は 3 ～ 1000 です。デフォルトは 3 です。

	コマンドまたはアクション	目的
		（注） hello 乗数を小さくすると、高速コンバージェンスとなりますが、ルーティングが不安定になる場合があります。
ステップ 7	**isis csnp-interval** *seconds* [**level-1** | **level-2**] 例： `Device(config-if)#isis csnp-interval 15`	（任意）インターフェイスに IS-IS CSNP を設定します。指定できる範囲は 0 ～ 65535 です。デフォルトは 10 秒です。
ステップ 8	**isis retransmit-interval** *seconds* 例： `Device(config-if)#isis retransmit-interval 7`	（任意）ポイントツーポイントリンクの IS-IS LSP の再送信間隔（秒）を設定します。整数で、ネットワーク上の 2 つのルータ間で予測されるラウンドトリップ遅延よりも大きい値を指定してください。指定できる範囲は 0 ～ 65535 です。デフォルトは 5 秒です。
ステップ 9	**isis retransmit-throttle-interval** *milliseconds* 例： `Device(config-if)#isis retransmit-throttle-interval 4000`	（任意）IS-IS LSP 再送信スロットルインターバルを設定します。これは、IS-IS LSP がポイントツーポイントリンク上で再送信される最大レート（パケット間のミリ秒数）です。指定できる範囲は 0 ー 65535 です。デフォルトは **isis lsp-interval** コマンドによって決定されます。
ステップ 10	**isis priority** *value* [**level-1** | **level-2**] 例： `Device(config-if)#isis priority 50`	（任意）指定ルータの優先順位を設定します。指定できる範囲は 0 ～ 127 です。デフォルトは 64 です。
ステップ 11	**isis circuit-type** {**level-1** | **level-1-2** | **level-2-only**} 例： `Device(config-if)#isis circuit-type level-1-2`	（任意）指定されたインターフェイス上のネイバーで必要な隣接タイプを設定します（インターフェイスの回線タイプを指定します）。 • **level-1**：このノードとネイバーの両方に共通のエリアアドレスが少なくとも 1 つある場合、レベル 1 隣接関係が確立されます。 • **level-1-2**：ネイバーもレベル 1 およびレベル 2 の両方として設定されていて、少なくとも 1 つの共通のエリアがある場合、レベル 1 およびレベル 2 隣接関係が確立されます。共通のエリアがない場合は、レベル 2 隣接関係が確立されます。これはデフォルト設定です。これがデフォルトのオプションです。

	コマンドまたはアクション	目的
		• **level 2**：レベル 2 隣接関係が確立されます。ネイバー ルータがレベル 1 ルータである場合、隣接関係は確立されません。
ステップ 12	**isis password** *password* [**level-1** | **level-2**] 例： `Device(config-if)#isis password secret`	（任意）インターフェイスの認証パスワードを設定します。デフォルトでは、認証はディセーブルに設定されています。レベル1またはレベル2を指定すると、それぞれレベル1またはレベル2ルーティング用のパスワードだけがイネーブルになります。レベルを指定しない場合、デフォルトはレベル1およびレベル 2 です。
ステップ 13	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。

IS-IS 認証の設定方法

ここでは、認証キーを生成する方法、インターフェイスの IS-IS 認証を設定する方法、およびインスタンスの IS-IS 認証を設定する方法について説明します。

認証キーの設定

複数のキーにライフタイムを設定できます。認証パケットを送信するために、最新の送信ライフタイム設定を持つキーが選択されます。複数のキーが同じ送信ライフタイム設定を持つ場合、キーはランダムに選択されます。受信した認証パケットを調べて受け入れるには、**accept-lifetime** コマンドを使用します。デバイスは、これらのライフタイムを認識している必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 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： `Device#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。

	コマンドまたはアクション	目的
ステップ 3	**key chain** *name-of-chain* 例： `Device(config)#key chain key10`	キーチェーンを識別し、キーチェーン 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	**key** *number* 例： `Device(config-keychain)#key 2000`	キー番号を識別します。範囲は 0 ～ 65535 です。
ステップ 5	**key-string** *text* 例： `Device(config-keychain-key)#Room 20, 10th floor`	キー ストリングを確認します。ストリングには 1 ～ 80 文字の大文字および小文字の英数字を指定できますが、最初の文字に数字を指定できません。
ステップ 6	**accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} 例： `Device(config-keychain-key)#accept-lifetime 12:30:00 Jan 25 1009 infinite`	（任意）キーを受信できる期間を指定します。 *start-time* および *end-time* 構文には、*hh:mm:ss month date year* または *hh:mm:ss date month year* のいずれかを使用できます。デフォルトは、デフォルトの *start-time* 以降、無制限です。指定できる最初の日付は 1993 年 1 月 1 日です。デフォルトの *end-time* および **duration** は **infinite** です。
ステップ 7	**send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} 例： `Device(config-keychain-key)#accept-lifetime 23:30:00 Jan 25 1019 infinite`	（任意）キーを送信できる期間を指定します。 *start-time* および *end-time* 構文には、*hh:mm:ss month date year* または *hh:mm:ss date month year* のいずれかを使用できます。デフォルトの *start-time* は **infinite** で、指定できる最初の日付は 1993 年 1 月 1 日です。デフォルトの *end-time* および **duration** は **infinite** です。
ステップ 8	**cryptographic-algorithm** {**hmac-sha-1** | **hmac-sha-256** | **hmac-sha-384** | **hmac-sha-512** | **md5** } 例： `Device(config-keychain-key)#cryptographic-algorithm hmac-sha1-256`	（任意）暗号化アルゴリズムを指定します。
ステップ 9	**end** 例： `Device(config-keychain-key)#end`	特権 EXEC モードに戻ります。
ステップ 10	**show key chain** 例：	認証キーの情報を表示します。

	コマンドまたはアクション	目的
	`Device#show key chain`	

IS-IS インスタンスの HMAC-MD5 またはクリアテキスト認証の設定

ある認証方法から別の認証方法へ円滑に移行を実現し、IS-IS PDU の継続的な認証を可能にするには、ネットワークで通信する各デバイスでこの手順を実行します。

始める前に

認証文字列キーが生成されている必要があります。ネットワーク内のすべてのデバイスで同じ認証文字列キーを設定する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 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： `Device#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 isis** [**area** *tag*] 例： `Device(config)#router isis 1`	IP ルーティング プロトコルとして IS-IS を有効化し、必要に応じてプロセスにタグを割り当てます。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	**authentication send-only** [**level-1** | **level-2**] 例： `Device(config-router)#authentication send-only`	指定した IS-IS インスタンスについて送信された（受信ではなく）PDU に対してのみ認証が実行されるように指定します。
ステップ 5	**authentication mode** {**md5** | **text**}[**level-1** | **level-2**] 例： `Device(config-router)#authentication mode md5`	指定された IS-IS インスタンスについて PDU で使用される認証のタイプを指定します。 • **md5**：MD5 認証。 • **text**：クリアテキスト認証。
ステップ 6	**authentication key-chain** *name-of-chain* [**level-1** | **level-2**] 例：	指定された IS-IS インスタンスについて認証が有効になります。

	コマンドまたはアクション	目的
	`Device(config-router)#authentication key-chain remote3754`	
ステップ 7	**no authentication send-only** 例： `Device(config-router)#no authentication send-only`	指定した IS-IS インスタンスについて送信および受信された PDU に対してのみ認証が実行されるように指定します。

IS-IS インターフェイスの HMAC-MD5 またはクリア テキスト認証の設定

ある認証方法から別の認証方法へ円滑に移行を実現し、IS-IS PDU の継続的な認証を可能にするには、ネットワークで通信する各デバイスでこの手順を実行します。

始める前に

認証文字列キーが生成されている必要があります。ネットワーク内のすべてのデバイスで同じ認証文字列キーを設定する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 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： `Device#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	**interface** *type number* 例： `Device(config)#interface ethernet 0`	インターフェイスを設定します。
ステップ 4	**isis authentication send-only** [**level-1** | **level-2**] 例： `Device(config-if)#isis authentication send-only`	指定した IS-IS インターフェイスについて送信された（受信ではなく）PDUに対してのみ認証が実行されるように指定します。

	コマンドまたはアクション	目的
ステップ 5	**isis authentication mode** {**md5** | **text**}[**level-1** | **level-2**] 例： `Device(config-if)#isis authentication mode md5`	指定された IS-IS インスタンスについて PDU で使用される認証のタイプを指定します。 • **md5**：MD5 認証。 • **text**：クリアテキスト認証。
ステップ 6	**isis authentication key-chain** *name-of-chain* [**level-1** | **level-2**] 例： `Device(config-if)#isis authentication key-chain multistate87723`	指定された IS-IS インスタンスについて MD5 認証が有効になります。
ステップ 7	**no isis authentication send-only** 例： `Device(config-if)#no isis authentication send-only`	IS-IS インスタンスについて送信および受信された PDU に対してのみ認証が実行されるように指定します。

IS-IS のモニタリングおよびメンテナンス

ルーティング テーブル、キャッシュ、およびデータベースの内容など、特定の IS-IS の統計情報を表示できます。また、特定のインターフェイス、フィルタ、またはネイバーに関する情報も表示できます。

次の表に、IS-IS ルーティングを消去および表示するために使用する特権 EXEC コマンドを示します。

表 *35 : IS-IS show* コマンド

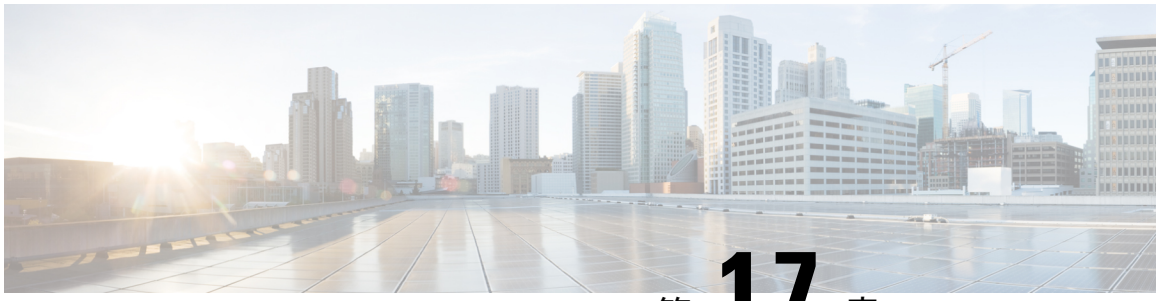
コマンド
show ip route isis
show isis database
show isis routes
show isis spf-log
show isis topology
show route-map

コマンド
trace clns *[接続先（Destination）]*

IS-IS の機能情報

表 *36 : IS-IS* の機能情報

機能名	リリース	機能情報
Intermediate System-to-Intermediate System（IS-IS）	Cisco IOS XE Everest 16.5.1a	この機能が導入されました。
	Cisco IOS XE Gibraltar 16.10.1	IS-IS は、セキュア ハッシュ アルゴリズム（SHA）認証（SHA-1、SHA-256、SHA-384、およびSHA-512）をサポートするようになりました。



第 17 章

プロトコル独立機能

プロトコル独立機能

分散型シスコ エクスプレス フォワーディング

シスコ エクスプレス フォワーディングに関する情報

シスコ エクスプレス フォワーディング（CEF）は、ネットワーク パフォーマンスを最適化するために使用されるレイヤ 3 IP スイッチング技術です。CEF には高度な IP 検索および転送アルゴリズムが実装されているため、レイヤ 3 スイッチングのパフォーマンスを最大化できます。高速スイッチング ルート キャッシュよりも CPU にかかる負担が少ないため、CEF はより多くの CPU 処理能力をパケット転送に割り当てることができます。スイッチ スタックでは、ハードウェアによって distributed CEF（dCEF）が使用されます。動的なネットワークでは、ルーティングの変更によって、高速スイッチング キャッシュ エントリが頻繁に無効になります。高速スイッチングキャッシュ エントリが無効になると、トラフィックがルートキャッシュによって高速スイッチングされずに、ルーティング テーブルによってプロセス スイッチングされることがあります。CEF および dCEF は転送情報ベース（FIB）検索テーブルを使用して、宛先ベースの IP パケット スイッチングを実行します。

CEF および dCEF での 2 つの主要なコンポーネントは、分散 FIB と分散隣接テーブルです。

- FIB はルーティング テーブルや情報ベースと同様、IP ルーティング テーブルに転送情報のミラーイメージが保持されます。ネットワーク内でルーティングまたはトポロジが変更されると、IP ルーティング テーブルがアップデートされ、これらの変更が FIB に反映されます。FIB には、IP ルーティング テーブル内の情報に基づいて、ネクストホップのアドレス情報が保持されます。FIB にはルーティング テーブル内の既知のルートがすべて格納されているため、CEF はルート キャッシュをメンテナンスする必要がなく、トラフィックのスイッチングがより効率化され、トラフィック パターンの影響も受けません。
- リンク層上でネットワーク内のノードが 1 ホップで相互に到達可能な場合、これらのノードは隣接関係にあると見なされます。CEF は隣接テーブルを使用し、レイヤ 2 アドレッシ

ング情報を付加します。隣接テーブルには、すべての FIB エントリに対する、レイヤ 2 のネクストホップのアドレスが保持されます。

スイッチまたはスイッチスタックは、ギガビット速度の回線レート IP トラフィックを達成するため特定用途向け集積回路（ASIC）を使用しているので、CEF または dCEF 転送はソフトウェア転送パス（CPU により転送されるトラフィック）にだけ適用されます。

シスコ エクスプレス フォワーディングの設定方法

デフォルトで、CEF または dCEF はグローバルにイネーブルに設定されています。何らかの理由でこれが無効になった場合は、**ip cef**または **ip cef distributed** グローバル コンフィギュレーション コマンドを使用し、再度有効に設定できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# 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 2	**ip cef** 例： `Device(config)# ip cef`	非スタッキング スイッチで CEF の動作をイネーブルにします。 ステップ 4 に進みます。
ステップ 3	**ip cef distributed** 例： `Device(config)# ip cef distributed`	アクティブ スイッチで CEF の動作をイネーブルにします。
ステップ 4	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 5	**ip route-cache cef** 例： `Device(config-if)# ip route-cache cef`	ソフトウェア転送トラフィック用のインターフェイスで CEF をイネーブルにします。 （注） **ip route-cache cef** コマンドはデフォルトで有効になっており、無効にはできません。
ステップ 6	**end** 例：	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
	`Device(config-if)# end`	
ステップ 7	**show ip cef** 例： `Device# show ip cef`	すべてのインターフェイスの CEF ステータスを表示します。
ステップ 8	**show cef linecard** [**detail**] 例： `Device# show cef linecard detail`	（任意）非スタッキング スイッチの CEF 関連インターフェイス情報を表示します。
ステップ 9	**show cef linecard** [*slot-number*] [**detail**] 例： `Device# show cef linecard 5 detail`	（任意）スタック内のすべてのスイッチ、または指定されたスイッチに対して、スイッチの CEF 関連インターフェイス情報をスタック メンバ別に表示します。 （任意）*slot-number* には、スタック メンバーのスイッチ番号を入力します。
ステップ 10	**show cef interface** [*interface-id*] 例： `Device# show cef interface gigabitethernet 1/0/1`	すべてのインターフェイスまたは指定されたインターフェイスの詳細な CEF 情報を表示します。
ステップ 11	**show adjacency** 例： `Device# show adjacency`	CEF の隣接テーブル情報を表示します。
ステップ 12	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

CEF トラフィック用のロードバランシングスキーム

CEFトラフィック用のロードバランシングスキームの設定に関する制約事項

- デバイスまたはデバイススタックメンバのロードバランシングを同じように、グローバルに設定する必要があります。
- CEF トラフィックのパケットごとのロードバランシングはサポートされていません。

CEF ロード バランシングの概要

CEF のロードバランシングを行うと、トラフィックを複数のパスに分散することにより、リソースを最適化することができます。CEFのロードバランシングは、送信元と宛先のパケット情報の組み合わせに基づいて動作します。

ロードバランシングは宛先単位で設定できます。ロードバランシングの判断はアウトバウンドインターフェイス上で行われるため、ロード バランシングは、アウトバウンド インターフェイスで設定する必要があります。

CEF トラフィックに対する宛先別ロードバランシング

宛先単位のロードバランシングにより、デバイスは、複数のパスを使用して、複数の発信元と宛先ホストのペアにわたって負荷を共有することができます。指定された発信元と宛先ホストのペアは、複数のパスを使用可能な場合であっても、同じパスを使用することが保証されています。異なるペアを宛先とするトラフィック ストリームは、異なるパスを使用します。

CEF がイネーブルの場合、宛先別ロード バランシングはデフォルトでイネーブルです。CEF をイネーブルにした場合、宛先単位のロードバランシングを使用するための追加タスクはありません。多くの状況では、ロード バランシングの方法として宛先単位を使用します。

宛先単位のロードバランシングはトラフィックの統計的な分散に依存しているため、発信元と宛先ホストのペア数が増大すると、ロード シェアリングがさらに有効になります。

宛先単位のロード バランシングを使用することにより、個々のホスト ペアのパケットが順に到達することが保証されます。特定のホストペアに宛てられたすべてのパケットは、（複数の場合も）同じリンクを介して転送されます。

CEF トラフィックに対するロード バランシング アルゴリズム

CEF トラフィックで使用するために、次のロード バランシング アルゴリズムが用意されています。ロードバランシング アルゴリズムは、**ip cef load-sharing algorithm** コマンドで選択します。

- オリジナルアルゴリズム：オリジナルのロード バランシング アルゴリズムでは、すべてのデバイスで同じアルゴリズムが使用されるため、複数のデバイスにわたるロードシェアリングで歪みが発生します。ネットワーク環境に応じて、アルゴリズムを選択する必要があります。
- ユニバーサルアルゴリズム：ユニバーサル ロード バランシング アルゴリズムでは、ネットワーク上の各デバイスは、発信元と宛先の各アドレスペアに対して異なるロードシェアリングの判断を行うことができます。これにより、ロードシェアリングの不均衡が解決されます。デバイスは、デフォルトではユニバーサル ロード シェアリングを実行するよう設定されています。

CEF トラフィックに対するロードバランシングの設定方法

ここでは、CEF トラフィックに対するロードバランシングの設定について説明します。

CEF の宛先別ロードバランシングの有効化または無効化

CEF の宛先単位のロードバランシングを有効または無効にするには、次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. [**no**] **ip load-sharing per-destination**
5. **end**

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： `Device# enable`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**configure terminal** 例： `Device# 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	**interface** *interface-id* 例： `Device(config-if)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 4	[**no**] **ip load-sharing per-destination** 例： `Device(config-if)# ip load-sharing per-destination`	インターフェイスで CEF の宛先別ロードバランシングを有効にします。 **no ip load-sharing per-destination** コマンドを使用すると、インターフェイスで CEF の宛先別ロードバランシングが無効になります。
ステップ 5	**end** 例： `Device(config-if)# end`	インターフェイス 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。

CEF トラフィックに対するトンネル ロードバランシング アルゴリズムの選択

ネットワーク環境に少数の発信元と宛先のペアしか存在しない場合には、トンネルアルゴリズムを選択します。デバイスは、デフォルトではユニバーサル ロード シェアリングを実行するよう設定されています。

CEF トラフィック用にトンネル ロード バランシング アルゴリズムを選択するには、次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **ip cef load-sharing algorithm** {**original** | **universal** [*id*] }
4. **end**

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： `Device# enable`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**configure terminal** 例： `Device# 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 cef load-sharing algorithm** {**original** | **universal** [*id*] } 例： `Device(config)# ip cef load-sharing algorithm universal`	CEF のロードバランシング アルゴリズムを選択します。 • **original** キーワードは、送信元 IP と宛先 IP のハッシュに基づいて、ロード バランシング アルゴリズムとしてオリジナルアルゴリズムを設定します。 • **universal** キーワードは、送信元 IP、宛先 IP、レイヤ 3 プロトコル、レイヤ 4 送信元ポート、レイヤ 4 宛先ポート、および IPv6 トラフィック ラベル（IPv6 トラフィック用）を使用するロード バランシング アルゴリズムを設定します。 • *id* 引数は、固定 ID です。
ステップ 4	**end** 例：	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
	`Device(config)# end`	

CEF トラフィックのロードバランシングの設定例

ここでは、CEF トラフィックのロードバランシングの設定例を示します。

例：CEF の宛先別ロードバランシングの有効化または無効化

CEF がイネーブルの場合、宛先別ロード バランシングはデフォルトでイネーブルです。次の例は、宛先単位のロード バランシングをディセーブルにする方法を示しています。

```
Device> enable
Device# configure terminal
Device(config)# interface Ethernet1/0/1
Device(config-if)# no ip load-sharing per-destination
Device(config-if)# end
```

等コスト ルーティング パスの個数

等コスト ルーティング パスに関する情報

同じネットワークへ向かう同じメトリックのルートが複数ルータに格納されている場合、これらのルートは等価コストを保有していると見なされます。ルーティング テーブルに複数の等コスト ルートが含まれる場合は、これらをパラレル パスと呼ぶこともあります。ネットワークへの等コスト パスがルータに複数格納されている場合、ルータはこれらを同時に使用できます。パラレル パスを使用すると、パスに障害が発生した場合に冗長性を確保できます。また、使用可能なパスにパケットの負荷を分散し、使用可能な帯域幅を有効利用することもできます。等コスト ルートは、スタック内の各スイッチでサポートされます。

等コスト ルートはルータによって自動的に取得、設定されますが、ルーティング テーブルの IP ルーティング プロトコルでサポートされるパラレル パスの最大数は制御可能です。スイッチ ソフトウェア では最大 32 の等コスト ルーティングが許可されていますが、スイッチ ハードウェアはルートあたり 17 パス以上は使用しません。

等コスト ルーティング パスの設定方法

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	**configure terminal** 例： `Device# 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** {**rip** | **ospf** | **eigrp**} 例： `Device(config)# router eigrp`	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	**maximum-paths** *maximum* 例： `Device(config-router)# maximum-paths 2`	プロトコル ルーティング テーブルのパラレル パスの最大数を設定します。指定できる範囲は 1 ～ 16 です。ほとんどの IP ルーティング プロトコルでデフォルトは 4 ですが、BGP の場合だけ 1 です。
ステップ 5	**end** 例： `Device(config-router)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show ip protocols** 例： `Device# show ip protocols`	*Maximum path* フィールドの設定を確認します。
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

スタティック ユニキャスト ルート

スタティック ユニキャスト ルートに関する情報

スタティック ユニキャスト ルートは、特定のパスを通過して送信元と宛先間でパケットを送受信するユーザー定義のルートです。ルータが特定の宛先へのルートを構築できない場合、スタティック ルートは重要で、到達不能なすべてのパケットが送信される最終ゲートウェイを指定する場合に有効です。

ユーザーによって削除されるまで、スタティック ルートはスイッチに保持されます。ただし、アドミニストレーティブ ディスタンスの値を割り当て、スタティック ルートをダイナミック ルーティング情報で上書きできます。各ダイナミック ルーティング プロトコルには、デフォルトのアドミニストレーティブディスタンスが設定されています（表 10 を参照）。ダイナミック ルーティング プロトコルの情報でスタティック ルートを上書きする場合は、スタティック

ルートのアドミニストレーティブディスタンスがダイナミックプロトコルのアドミニストレーティブ ディスタンスよりも大きな値になるように設定します。

表 *37*: ダイナミック ルーティング プロトコルのデフォルトのアドミニストレーティブ ディスタンス

ルートの送信元	デフォルト距離
接続されているインターフェイス	0
スタティック ルート	1
EIGRP サマリー ルート	5
内部 EIGRP	90
IGRP	100
OSPF	110
内部 BGP	200
不明	225

インターフェイスを指し示すスタティックルートは、RIP、IGRP、およびその他のダイナミック ルーティング プロトコルを通してアドバタイズされます。**redistribute** スタティック ルータ コンフィギュレーション コマンドが、これらのルーティングプロトコルに対して指定されているかどうかは関係ありません。これらのスタティックルートがアドバタイズされるのは、インターフェイスを指し示すスタティック ルートが接続された結果、静的な性質を失ったとルーティング テーブルで見なされるためです。ただし、network コマンドで定義されたネットワーク以外のインターフェイスに対してスタティックルートを定義する場合は、ダイナミック ルーティング プロトコルに **redistribute** スタティックコマンドを指定しない限り、ルートはアドバタイズされません。

インターフェイスがダウンすると、ダウンしたインターフェイスを経由するすべてのスタティック ルートが IP ルーティング テーブルから削除されます。転送ルータのアドレスとして指定されたアドレスへ向かう有効なネクストホップがスタティック ルート内に見つからない場合は、IP ルーティング テーブルからそのスタティック ルートも削除されます。

スタティック ユニキャスト ルートの設定

スタティック ユニキャスト ルートは、特定のパスを通過して送信元と宛先間でパケットを送受信するユーザー定義のルートです。ルータが特定の宛先へのルートを構築できない場合、スタティック ルートは重要で、到達不能なすべてのパケットが送信される最終ゲートウェイを指定する場合に有効です。

スタティック ルートを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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 route prefix mask** {*address* | *interface*} [*distance*] 例： `Device(config)# ip route prefix mask gigabitethernet 1/0/4`	スタティック ルートを確立します。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show ip route** 例： `Device# show ip route`	設定を確認するため、ルーティング テーブルの現在の状態を表示します。
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

次のタスク

スタティックルートを削除するには、**no ip route** *prefix mask* {*address*| *interface*} グローバル コンフィギュレーション コマンドを使用します。ユーザーによって削除されるまで、スタティックルートはデバイスに保持されます。

デフォルトのルートおよびネットワーク

デフォルトのルートおよびネットワークに関する情報

ルータは、他のすべてのネットワークへのルートを学習できません。完全なルーティング機能を実現するには、一部のルータをスマートルータとして使用し、それ以外のルータのデフォルト ルートをスマート ルータ宛てに指定します（スマート ルータにはインターネットワーク全体のルーティング テーブルに関する情報が格納されます）。これらのデフォルト ルートは動的に学習できますが、ルータごとに設定することもできます。ほとんどのダイナミックな内部ルーティング プロトコルには、スマート ルータを使用してデフォルト情報を動的に生成し、他のルータに転送するメカニズムがあります。

指定されたデフォルト ネットワークに直接接続されたインターフェイスがルータに存在する場合は、そのデバイス上で動作するダイナミック ルーティング プロトコルによってデフォルト ルートが生成されます。RIP の場合は、疑似ネットワーク 0.0.0.0 がアドバタイズされます。

ネットワークのデフォルトを生成しているルータには、そのルータ自身のデフォルトルートも指定する必要があります。ルータが自身のデフォルトルートを生成する方法の 1 つは、適切なデバイスを経由してネットワーク 0.0.0.0 に至るスタティック ルートを指定することです。

ダイナミック ルーティング プロトコルによってデフォルト情報を送信するときは、特に設定する必要はありません。ルーティング テーブルは定期的にスキャンされ、デフォルト ルートとして最適なデフォルト ネットワークが選択されます。IGRP ネットワークでは、システムのデフォルト ネットワークの候補が複数存在する場合もあります。Cisco ルータでは、デフォルト ルートまたは最終ゲートウェイを設定するため、アドミニストレーティブ ディスタンスおよびメトリック情報を使用します。

ダイナミックなデフォルト情報がシステムに送信されない場合は、**ip default-network** グローバル コンフィギュレーション コマンドを使用し、デフォルトルートの候補を指定します。このネットワークが任意の送信元のルーティング テーブルに格納されている場合は、デフォルト ルートの候補としてフラグ付けされます。ルータにデフォルト ネットワークのインターフェイスが存在しなくても、そこへのパスが格納されている場合、そのネットワークは 1 つの候補と見なされ、最適なデフォルト パスへのゲートウェイが最終ゲートウェイになります。

デフォルトのルートおよびネットワークの設定方法

デフォルトルートおよびネットワークを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# 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。

	コマンドまたはアクション	目的
ステップ 2	**ip default-network** *network number* 例： `Device(config)# ip default-network 1`	デフォルト ネットワークを指定します。
ステップ 3	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 4	**show ip route** 例： `Device# show ip route`	最終ゲートウェイで選択されたデフォルトルートを表示します。
ステップ 5	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ルーティング情報を再配信するためのルート マップ

ルート マップの概要

スイッチでは複数のルーティング プロトコルを同時に実行し、ルーティング プロトコル間で情報を再配信できます。ルーティングプロトコル間での情報の再配信は、サポートされているすべての IP ベース ルーティング プロトコルに適用されます。

2 つのドメイン間で拡張パケット フィルタまたはルート マップを定義することにより、ルーティング ドメイン間でルートの再配信を条件付きで制御することもできます。**match** および **set** ルート マップ コンフィギュレーション コマンドは、ルートマップの条件部を定義します。**match** コマンドは、条件が一致する必要があることを指定しています。**set** コマンドは、ルーティングアップデートが match コマンドで定義した条件を満たす場合に行われる処理を指定します。再配布はプロトコルに依存しない機能ですが、**match** および **set** ルート マップ コンフィギュレーション コマンドの一部は特定のプロトコル固有のものです。

route-map コマンドのあとに、**match** コマンドおよび **set** コマンドをそれぞれ 1 つまたは複数指定します。**match** コマンドを指定しない場合は、すべて一致すると見なされます。**set** コマンドを指定しない場合、一致以外の処理はすべて実行されません。このため、少なくとも 1 つの **match** または **set** コマンドを指定する必要があります。

（注） **set** ルート マップ コンフィギュレーション コマンドを使用しないルートマップは、CPU に送信されるので、CPU の使用率が高くなります。

ルートマップステートメントは、**permit** または **deny** として識別することもできます。ステートメントが拒否としてマークされている場合、一致基準を満たすパケットは通常の転送チャネルを通じて送り返されます（宛先ベースルーティング）、ステートメントが許可としてマークされている場合は、一致基準を満たすパケットに set コマンドが適用されます。一致基準を満たさないパケットは、通常のルーティング チャネルを通じて転送されます。

ルート マップの設定方法

次に示すステップ 3 ～ 14 はそれぞれ任意ですが、少なくとも 1 つの **match** ルート マップ コンフィギュレーション コマンド、および 1 つの **set** ルート マップ コンフィギュレーション コマンドを入力する必要があります。

（注） キーワードは、ルート配信を制御する手順で定義されているものと同じです。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# 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 2	**route-map***map-tag* [**permit** | **deny**] [*sequence number*] 例： `Device(config)# route-map rip-to-ospf permit 4`	再配信を制御するために使用するルート マップを定義し、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。 *map-tag*：ルート マップ用のわかりやすい名前を指定します。**redistribute**ルータ コンフィギュレーション コマンドはこの名前を使用して、このルートマップを参照します。複数のルート マップで同じマップ タグ名を共有できます。 （任意）**permit** が指定され、このルートマップの一致条件が満たされている場合は、set アクションの制御に従ってルートが再配信されます。**deny**が指定が指定されている場合、ルートは再配信されません。 *sequence number*（任意）：同じ名前によってすでに設定されているルート マップのリスト内で、新しいルート マップの位置を指定する番号です。
ステップ 3	**match as-path** *path-list-number* 例：	BGP AS パス アクセス リストと照合します。

	コマンドまたはアクション	目的
	`Device(config-route-map)# match as-path 10`	
ステップ 4	**match community-list** *community-list-number* [**exact**] 例： `Device(config-route-map)# match community-list 150`	BGP コミュニティ リストのマッチングを行います。
ステップ 5	**match ip address** {*access-list-number* | *access-list-name*} [*...access-list-number* | *...access-list-name*] 例： `Device(config-route-map)# match ip address 5 80`	名前または番号を指定し、標準アクセス リストと照合します。1 ～ 199 の整数を指定できます。
ステップ 6	**match metric** metric-value 例： `Device(config-route-map)# match metric 2000`	指定されたルート メトリックと一致させます。*metric-value* には、0 ～ 4294967295 の値が指定された、EIGRP のメトリックを指定できます。
ステップ 7	**match ip next-hop** {*access-list-number* | *access-list-name*} [*...access-list-number* | *...access-list-name*] 例： `Device(config-route-map)# match ip next-hop 8 45`	指定されたアクセスリスト（番号1～199）のいずれかで送信される、ネクストホップのルータ アドレスと一致させます。
ステップ 8	**match tag** *tag value* [*...tag-value*] 例： `Device(config-route-map)# match tag 3500`	1つまたは複数のルートタグ値からなるリスト内の指定されたタグ値と一致させます。0～4294967295 の整数を指定できます。
ステップ 9	**match interface***type number* [*...type-number*] 例： `Device(config-route-map)# match interface gigabitethernet 1/0/1`	指定されたインターフェイスの1つから、指定されたネクスト ホップへのルートと一致させます。
ステップ 10	**match ip route-source** {*access-list-number* | *access-list-name*} [*...access-list-number* | *...access-list-name*] 例： `Device(config-route-map)# match ip route-source 10 30`	アドバタイズされた指定のアクセス リストによって指定したアドレスに一致します。
ステップ 11	**match route-type** {**local** | **internal** | **external** [**type-1** | **type-2**]}	指定された **route-type** と一致させます。

	コマンドまたはアクション	目的
	例： `Device(config-route-map)# match route-type local`	• **local**：ローカルに生成された BGP ルート。 • **internal**：OSPF エリア内およびエリア間ルート、または EIGRP 内部ルート。 • **external**：OSPF 外部ルート（タイプ 1 またはタイプ 2）または EIGRP 外部ルート。
ステップ 12	**set dampening** *halflife reuse suppress max-suppress-time* 例： `Device(config-route-map)# set dampening 30 1500 10000 120`	BGP ルート ダンプニング係数を設定します。
ステップ 13	**set local-preference** *value* 例： `Device(config-route-map)# set local-preference 100`	ローカル BGP パスに値を割り当てます。
ステップ 14	**set origin** {**igp** | **egp** *as* | **incomplete**} 例： `Device(config-route-map)# set origin igp`	BGP 送信元コードを設定します。
ステップ 15	**set as-path** {**tag** | **prepend** *as-path-string*} 例： `Device(config-route-map)# set as-path tag`	BGP の自律システム パスを変更します。
ステップ 16	**set level** {**level-1** | **level-2** | **level-1-2** | **stub-area** | **backbone**} 例： `Device(config-route-map)# set level level-1-2`	ルーティング ドメインの指定エリアにアドバタイズされるルートのレベルを設定します。**stub-area** および **backbone** は、OSPF NSSA およびバックボーンエリアです。
ステップ 17	**set metric** *metric value* 例： `Device(config-route-map)# set metric 100`	再配布されるルートを指定するためのメトリック値を設定します（EIGRP のみ）。*metric value* は -294967295 ～ 294967295 の整数です。
ステップ 18	**set metric***bandwidth delay reliability loading mtu* 例： `Device(config-route-map)# set metric 10000 10 255 1 1500`	再配布されるルートを指定するためのメトリック値を設定します（EIGRP のみ）。 • *bandwidth*：0 ～ 4294967295 の範囲のルートのメトリック値または IGRP 帯域幅（キロビット/秒単位）。

	コマンドまたはアクション	目的
		• *delay*：0 ～ 4294967295 の範囲のルート遅延（10 マイクロ秒単位）。 • *reliability*：0 ～ 255 の数値で表されるパケット伝送の成功可能性。255 は信頼性が 100% であること、0 は信頼性がないことを意味します。 • *loading*：0 ～ 255 の数値で表されるルートの有効帯域幅（255 は 100% の負荷）。 • *mtu*：ルートの MTU の最小サイズ（バイト単位）。範囲は 0 ～ 4294967295 です。
ステップ 19	**set metric-type** {**type-1** | **type-2**} 例： `Device(config-route-map)# set metric-type type-2`	再配信されるルートに OSPF 外部メトリック タイプを設定します。
ステップ 20	**set metric-type internal** 例： `Device(config-route-map)# set metric-type internal`	ネクスト ホップの IGP メトリックと一致するように、EBGP ネイバーにアドバタイズされるプレフィックスの Multi-Exit 識別子（MED）値を設定します。
ステップ 21	**set weight** *number* 例： `Device(config-route-map)# set weight 100`	ルーティング テーブルの BGP 重みを設定します。指定できる値は 1 ～ 65535 です。
ステップ 22	**end** 例： `Device(config-route-map)# end`	特権 EXEC モードに戻ります。
ステップ 23	**show route-map** 例： `Device# show route-map`	設定を確認するため、設定されたすべてのルート マップ、または指定されたルート マップだけを表示します。
ステップ 24	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ルート配信の制御方法

次に示すステップ 3 ～ 14 はそれぞれ任意ですが、少なくとも 1 つの **match** ルート マップ コンフィギュレーション コマンド、および 1 つの **set** ルート マップ コンフィギュレーション コマンドを入力する必要があります。

（注） キーワードは、再配信用にルート マップを設定する手順で定義されているものと同じです。

ルーティング プロトコルのメトリックを、必ずしも別のルーティング プロトコルのメトリックに変換する必要はありません。たとえば、RIP メトリックはホップ カウントで、IGRP メトリックは5つの特性の組み合わせです。このような場合は、メトリックを独自に設定し、再配信されたルートに割り当てます。ルーティング情報を制御せずにさまざまなルーティング プロトコル間で交換するとルーティングループが発生し、ネットワーク動作が著しく低下することがあります。

メトリック変換の代わりに使用されるデフォルトの再配信メトリックが定義されていない場合は、ルーティング プロトコル間で自動的にメトリック変換が発生することがあります。

- RIP はスタティック ルートを自動的に再配信できます。スタティック ルートにはメトリック 1（直接接続）が割り当てられます。
- デフォルト モードになっている場合、どのプロトコルも他のルーティング プロトコルを再配信できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** **例**： `Device# 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 2	**router** {**rip** | **ospf** | **eigrp**} **例**： `Device(config)# router eigrp 10`	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 3	**redistribute** *protocol* [*process-id*] {**level-1** | **level-1-2** | **level-2**} [**metric** *metric-value*] [**metric-type** *type-value*] [**match internal** | **external** *type-value*] [**tag** *tag-value*] [**route-map** *map-tag*] [**weight** *weight*] [**subnets**] **例**： `Device(config-router)# redistribute eigrp 1`	ルーティング プロトコル間でルートを再配信します。route-map を指定しないと、すべてのルートが再配信されます。キーワード **route-map** に *map-tag* を指定しないと、ルートは配信されません。

	コマンドまたはアクション	目的
ステップ 4	**default-metric** *number* 例： `Device(config-router)# default-metric 1024`	現在のルーティングプロトコルが、再配信されたすべてのルートに対して同じメトリック値を使用するように設定します（RIP と OSPF）。
ステップ 5	**default-metric** bandwidth delay reliability loading mtu 例： `Device(config-router)# default-metric 1000 100 250 100 1500`	EIGRP ルーティング プロトコルが、EIGRP 以外で再配信されたすべてのルートに対して同じメトリック値を使用するように設定します。
ステップ 6	**end** 例： `Device(config-router)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show route-map** 例： `Device# show route-map`	設定を確認するため、設定されたすべてのルートマップ、または指定されたルートマップだけを表示します。
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ポリシーベース ルーティング

PBR の設定に関する制約事項

- ポリシーベースルーティング（PBR）は、トラフィックの GRE トンネルへの転送ではサポートされません。これは、任意のインターフェイスに適用される PBR と、トラフィックの GRE トンネルへの転送（PBR ネクストホップもしくはデフォルトのトネクストホップまたは設定済みのインターフェイスを使用）に適用される PBR に適用されます。
- PBR は、GRE トンネル自体ではサポートされていません（GRE トンネル自体のもとで適用されます）。
- Cisco IOS XE Amsterdam 17.3.5 リリース以降、PBR は断片化されたトラフィックには適用されません。断片化されたトラフィックは、通常のルーティングパスに従います。
- PBR とネットワークアドレス変換（NAT）は、同じインターフェイスではサポートされません。PBR と NAT は、異なるインターフェイス上に設定されている場合にのみ連携します。

ポリシーベース ルーティングの概要

PBR を使用すると、トラフィック フローに定義済みポリシーを設定できます。PBR を使用してルーティングをより細かく制御するには、ルーティングプロトコルから取得したルートの信頼度を小さくします。PBR は、次の基準に基づいて、パスを許可または拒否するルーティングポリシーを設定したり、実装したりできます。

- 特定のエンド システムの ID
- アプリケーション
- プロトコル

PBR を使用すると、等価アクセスや送信元依存ルーティング、インタラクティブ対バッチ トラフィックに基づくルーティング、専用リンクに基づくルーティングを実現できます。たとえば、在庫記録を本社に送信する場合は高帯域で高コストのリンクを短時間使用し、電子メールなど日常的に使用するアプリケーションデータは低帯域で低コストのリンクで送信できます。

PBR が有効な場合は、アクセス コントロール リスト（ACL）を使用してトラフィックを分類し、各トラフィックがそれぞれ異なるパスを経由するようにします。PBR は着信パケットに適用されます。PBR が有効なインターフェイスで受信されたすべてのパケットは、ルート マップを通過します。ルートマップで定義された基準に基づいて、パケットは適切なネクストホップに転送（ルーティング）されます。

- 許可とマークされているルート マップ文は次のように処理されます。
  - match コマンドは長さまたは複数の ACL で照合できます。ルート マップ文には複数の match コマンドを含めることができます。論理関数またはアルゴリズム関数は、許可または拒否の決定がされるまで、すべての match コマンドで実行されます。

    次に例を示します。

    match length A B

    match ip address acl1 acl2

    match ip address acl3

  パケットは、match length A B または acl1 または acl2 または acl3 により許可される場合に許可されます。

  - 決定が許可の場合は、set コマンドで指定されたアクションがパケットで適用されます。
  - 下された決定が拒否の場合は、PBR アクション（set コマンドで指定された）が適用されません。代わりに、処理ロジックが、シーケンス内の次のルートマップ文（シーケンス番号が次に高い文）に移動します。次の文が存在しない場合は、PBR 処理が終了し、パケットがデフォルトの IP ルーティング テーブルを使用してルーティングされます。
- PBR では、拒否としてマークされているルートマップ ステートメントはサポートされません。

標準 IP ACL を使用すると、アプリケーション、プロトコル タイプ、またはエンド ステーションに基づいて一致基準を指定するように、送信元アドレスまたは拡張 IP ACL の一致基準を指定できます。一致が見つかるまで、ルートマップにこのプロセスが行われます。一致が見つからない場合、通常の宛先ベースルーティングが行われます。match ステートメント リストの末尾には、暗黙の拒否ステートメントがあります。

match 句が満たされた場合は、set 句を使用して、パス内のネクスト ホップ ルータを識別する IP アドレスを指定できます。

ローカル PBR 設定は、デバイス管理目的で生成される RADIUS パケットの DSCP マーキングの設定をサポートします。

PBR の設定方法

- PBR を使用するには、スタンドアロンスイッチまたはアクティブスイッチ上で Network Essentials ライセンスをイネーブルにしておく必要があります。
- マルチキャスト トラフィックには、ポリシーによるルーティングが行われません。PBR が適用されるのはユニキャスト トラフィックだけです。
- ルーテッド ポートまたは SVI 上で、PBR を有効にできます。
- スイッチは一致長に基づき PBR をサポートします。
- レイヤ 3 モードの EtherChannel ポート チャネルにはポリシー ルート マップを適用できますが、EtherChannel のメンバーである物理インターフェイスには適用できません。適用しようとすると、コマンドが拒否されます。ポリシー ルート マップが適用されている物理インターフェイスは、EtherChannel のメンバーになることができません。
- スイッチまたはスイッチ スタックには最大 128 個の IP ポリシー ルート マップを定義できます。
- スイッチまたはスイッチ スタックには、PBR 用として最大 512 個のアクセス コントロール エントリ（ACE）を定義できます。
- ルート マップに一致基準を設定する場合は、次の注意事項に従ってください。
  - ローカル アドレス宛てのパケットを許可する ACL と照合させないでください。
- WCCP と PBR は、スイッチ インターフェイスで相互に排他的です。PBR がインターフェイスで有効になっているときは、WCCP を有効にできません。その反対の場合も同じで、WCCP がインターフェイスで有効になっているときは、PBR を有効にできません。
- PBR で使用されるハードウェア エントリ数は、ルート マップ自体、使用される ACL、ACL およびルート マップ エントリの順序によって異なります。
- TOS、DSCP、および IP Precedence に基づく PBR はサポートされません。
- set interface、set default next-hop、および set default interface はサポートされません。
- **ip next-hop recursive** および **ip next-hop verify availability** 機能は使用できません。next-hop は、直接接続される必要があります。

- set アクションのないポリシー マップはサポートされます。一致パケットは通常どおりにルーティングされます。
- match 句のないポリシー マップはサポートされます。set アクションはすべてのパケットに適用されます。

デフォルトでは、PBR はスイッチ上で無効です。PBR を有効にするには、一致基準および結果アクションを指定するルートマップを作成する必要があります。次に、特定のインターフェイスでそのルート マップ用の PBR を有効にします。指定したインターフェイスに着信したパケットのうち、match 句と一致したものはすべて PBR の対象になります。

スイッチ（CPU）で生成されたパケットまたはローカルパケットは、通常どおりにポリシールーティングされません。スイッチ上でローカル PBR をグローバルに有効にすると、そのスイッチから送信されたすべてのユニキャストパケットがローカル PBR の影響を受けます。ローカル PBR に関してサポートされているプロトコルは、NTP、DNS、MSDP、SYSLOG、および TFTP です。ローカル PBR は、デフォルトで無効に設定されています。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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	**route-map** *map-tag* [**permit**] [*sequence number*] 例： `Device(config)# route-map pbr-map permit`	パケットの出力場所を制御するために使用するルート マップを定義し、ルート マップのコンフィギュレーション モードを開始します。 • *map-tag* － ：ルートマップ用のわかりやすい名前を指定します。 **ip policy route-map** インターフェイス コンフィギュレーション コマンドは、この名前を使用して、このルートマップを参照します。同じ map-tag がある複数の route-map 文は、1 つの route-map を定義します。 •（任意） **permit** － ： **permit** が指定され、このルートマップの一致条件が満たされている場合は、set アクションの制御に従ってルートがポリシールーティングされます。

	コマンドまたはアクション	目的
		• （任意） *sequence number* — ：シーケンス番号は、特定のルートマップ内の route-map ステートメントの位置を示します。
ステップ 4	**match ip address** {*access-list-number* | *access-list-name*} [*access-list-number* |*...access-list-name*] 例： `Device(config-route-map)# match ip address 110 140`	1 つ以上の標準または拡張アクセス リストで許可されている送信元および宛先 IP アドレスを照合します。ACL は、複数の送信元および宛先 IP アドレスでも照合できます。 **match** コマンドを指定しない場合、ルートマップはすべてのパケットに適用されます。
ステップ 5	**match length min max** 例： `Device(config-route-map)# match length 64 1500`	パケット長と照合します。
ステップ 6	**set ip next-hop** *ip-address* [*...ip-address*] 例： `Device(config-route-map)# set ip next-hop 10.1.6.2`	基準と一致するパケットの動作を指定します。パケットのルーティング先となるネクスト ホップを設定します（ネクスト ホップは隣接している必要があります）。
ステップ 7	**exit** 例： `Device(config-route-map)# exit`	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 8	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するインタフェースを指定します。
ステップ 9	**ip policy route-map** *map-tag* 例： `Device(config-if)# ip policy route-map pbr-map`	レイヤ 3 インターフェイス上で PBR を有効にし、使用するルート マップを識別します。1 つのインターフェイスに設定できるルートマップは、1つだけです。ただし、異なるシーケンス番号を持つ複数のルート マップ エントリを設定できます。これらのエントリは、最初の一致が見つかるまで、シーケンス番号順に評価されます。一致が見つからない場合、パケットは通常どおりにルーティングされます。
ステップ 10	**ip route-cache policy** 例： `Device(config-if)# ip route-cache policy`	（任意）PBR の高速スイッチングを有効にします。PBR の高速スイッチングを有効にするには、PBR を有効にする必要があります。

	コマンドまたはアクション	目的
ステップ 11	**exit** 例： `Device(config-if)# exit`	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 12	**ip local policy route-map** *map-tag* 例： `Device(config)# ip local policy route-map local-pbr`	（任意）ローカル PBR を有効にして、スイッチから送信されるパケットに PBR を実行します。ローカル PBR は、スイッチによって生成されるパケットに適用されます。着信パケットには適用されません。
ステップ 13	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 14	**show route-map** [*map-name*] 例： `Device# show route-map`	（任意）設定を確認するため、設定されたすべてのルート マップ、または指定されたルート マップだけを表示します。
ステップ 15	**show ip policy** 例： `Device# show ip policy`	（任意）インターフェイスに付加されたポリシー ルート マップを表示します。
ステップ 16	**show ip local policy** 例： `Device# show ip local policy`	（任意）ローカル PBR が有効であるかどうか、および有効である場合は使用されているルート マップを表示します。

ルーティング情報のフィルタリング

ルーティング プロトコル情報をフィルタリングする場合は、以下の作業を実行します。

（注） OSPF プロセス間でルートが再配信される場合、OSPF メトリックは保持されません。

受動インターフェイスの設定

ローカルネットワーク上の他のルータが動的にルートを取得しないようにするには、**passive-interface** ルータ コンフィギュレーション コマンドを使用し、ルーティング アップデート メッセージがルータインターフェイスから送信されないようにします。OSPF プロトコルでこのコマンドを使用すると、パッシブに指定したインターフェイス アドレスが OSPF ドメインのスタブ ネットワークとして表示されます。OSPF ルーティング情報は、指定されたルータ インターフェイスから送受信されません。

多数のインターフェイスが存在するネットワークで、インターフェイスを手動でパッシブに設定する作業を回避するには、**passive-interface default** ルータ コンフィギュレーション コマンドを使用し、すべてのインターフェイスをデフォルトでパッシブになるように設定します。このあとで、隣接関係が必要なインターフェイスを手動で設定します。

パッシブとして有効にしたインターフェイスを確認するには、**show ip ospf interface** などのネットワークモニタリング用特権EXECコマンドを使用します。アクティブとして有効にしたインターフェイスを確認するには、**show ip interface** 特権 EXEC コマンドを使用します。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# 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 2	**router** {**rip** | **ospf** | **eigrp**} 例： `Device(config)# router ospf`	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 3	**passive-interface** *interface-id* 例： `Device(config-router)# passive-interface gigabitethernet 1/0/1`	指定されたレイヤ 3 インターフェイス経由のルーティング アップデートの送信を抑制します。
ステップ 4	**passive-interface default** 例： `Device(config-router)# passive-interface default`	（任意）すべてのインターフェイスを、デフォルトでパッシブとなるように設定します。
ステップ 5	**no passive-interface** *interface type* 例： `Device(config-router)# no passive-interface gigabitethernet1/0/3 gigabitethernet 1/0/5`	（任意）隣接関係を送信する必要があるインターフェイスだけをアクティブにします。
ステップ 6	**network** *network-address* 例： `Device(config-router)# network 10.1.1.1`	（任意）ルーティング プロセス用のネットワークリストを指定します。*network-address* はIPアドレスです。
ステップ 7	**end** 例：	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
	`Device(config-router)# end`	
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ルーティング アップデートのアドバタイズおよび処理の制御

アクセス制御リストと **distribute-list** ルータ コンフィギュレーション コマンドを組み合わせて使用すると、ルーティングアップデート中にルートのアドバタイズを抑制し、他のルータが 1 つまたは複数のルートを取得しないようにできます。この機能を OSPF で使用した場合は外部ルートにだけ適用されるため、インターフェイス名を指定できません。

distribute-list ルータ コンフィギュレーション コマンドを使用し、着信したアップデートのリストのうち特定のルートを処理しないようにすることもできます。（OSPF にこの機能は適用されません）。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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** {**rip** | **eigrp**} 例： `Device(config)# router eigrp 10`	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	**distribute-list** {*access-list-number* | *access-list-name*} **out** [*interface-name* | *routing process* | *autonomous-system-number*] 例： `Device(config-router)# distribute 120 out gigabitethernet 1/0/7`	アクセス リスト内のアクションに応じて、ルーティングアップデート内のルートのアドバタイズを許可または拒否します。

	コマンドまたはアクション	目的
ステップ 5	**distribute-list** {*access-list-number* | *access-list-name*} **in** [*type-number*] 例： `Device(config-router)# distribute-list 125 in`	アップデートにリストされたルートの処理を抑制します。
ステップ 6	**end** 例： `Device(config-router)# end`	特権 EXEC モードに戻ります。
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーションファイルに設定を保存します。

ルーティング情報の送信元のフィルタリング

一部のルーティング情報が他の情報よりも正確な場合があるため、フィルタリングを使用して、さまざまな送信元から送られる情報にプライオリティを設定できます。「アドミニストレーティブディスタンス」は、ルータやルータのグループなど、ルーティング情報の送信元の信頼性を示す数値です。大規模ネットワークでは、他のルーティングプロトコルよりも信頼できるルーティング プロトコルが存在する場合があります。アドミニストレーティブ ディスタンスの値を指定すると、ルータはルーティング情報の送信元をインテリジェントに区別できるようになります。常にルーティング プロトコルのアドミニストレーティブ ディスタンスが最短（値が最小）であるルートが選択されます。

各ネットワークには独自の要件があるため、アドミニストレーティブディスタンスを割り当てる一般的な注意事項はありません。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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。

	コマンドまたはアクション	目的
ステップ 3	**router** {**rip** | **ospf** | **eigrp**} 例： `Device(config)# router eigrp 10`	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	**distance** *weight* {*ip-address* {*ip-address mask*}} [*ip access list*] 例： `Device(config-router)# distance 50 10.1.5.1`	アドミニストレーティブ ディスタンスを定義します。 *weight*：アドミニストレーティブディスタンスは10 ～ 255 の整数です。単独で使用した場合、*weight* はデフォルトのアドミニストレーティブディスタンスを指定します。ルーティング情報の送信元に他の指定がない場合に使用されます。アドミニストレーティブディスタンスが 255 のルートはルーティング テーブルに格納されません。 （任意）*ip access list*：着信ルーティング アップデートに適用される IP 標準または IP 拡張アクセス リストです。
ステップ 5	**end** 例： `Device(config-router)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show ip protocols** 例： `Device# show ip protocols`	指定されたルーティング プロセス用のデフォルトのアドミニストレーティブ ディスタンスを表示します。
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

認証キーの管理

キー管理を使用すると、ルーティング プロトコルで使用される認証キーを制御できます。一部のプロトコルでは、キー管理を使用できません。認証キーは EIGRP および RIP バージョン 2 で使用できます。

前提条件

認証キーを管理する前に、認証をイネーブルにする必要があります。プロトコルに対して認証をイネーブルにする方法については、該当するプロトコルについての説明を参照してくださ

い。認証キーを管理するには、キー チェーンを定義してそのキー チェーンに属するキーを識別し、各キーの有効期間を指定します。各キーは、独自のキー識別子（**key** *number* キーチェーン コンフィギュレーション コマンドで指定されたもの）を保持し、ローカルに格納されています。キー ID、およびメッセージに関連付けられたインターフェイスの組み合わせにより、使用中の認証アルゴリズムおよび Message Digest 5（MD5）認証キーが一意に識別されます。

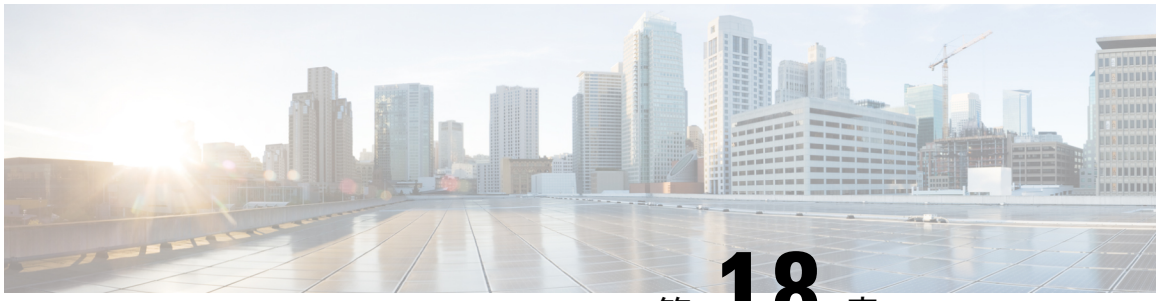
認証キーの設定方法

有効期間が指定された複数のキーを設定できます。存在する有効なキーの数にかかわらず、送信される認証パケットは1つだけです。最小の番号から順にキー番号が調べられ、最初に見つかった有効なキーが使用されます。キー変更中は、有効期間が重なっても問題ありません。これらの有効期間は、ルータに通知する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# 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 2	**key chain** *name-of-chain* 例： `Device(config)# key chain key10`	キーチェーンを識別し、キーチェーン 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	**key** *number* 例： `Device(config-keychain)# key 2000`	キー番号を識別します。有効値は 0 ～ 2147483647 です。
ステップ 4	**key-string** *text* 例： `Device(config-keychain)# Room 20, 10th floor`	キー ストリングを確認します。ストリングには 1 ～ 80 文字の大文字および小文字の英数字を指定できますが、最初の文字に数字を指定できません。
ステップ 5	**accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} 例： `Device(config-keychain)# accept-lifetime 12:30:00 Jan 25 1009 infinite`	（任意）キーを受信できる期間を指定します。 *start-time* および *end-time* 構文には、*hh:mm:ss Month date year* または *hh:mm:ss date Month year* のいずれかを使用できます。デフォルトは、デフォルトの *start-time* 以降、無制限です。指定できる最初の日付は 1993 年 1 月 1 日です。デフォルトの *end-time* および **duration** は **infinite** です。
ステップ 6	**send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*}	（任意）キーを送信できる期間を指定します。

	コマンドまたはアクション	目的
	例： `Device(config-keychain)# accept-lifetime 23:30:00 Jan 25 1019 infinite`	*start-time* および *end-time* 構文には、*hh:mm:ss Month date year* または *hh:mm:ss date Month year* のいずれかを使用できます。デフォルトは、デフォルトの *start-time* 以降、無制限です。指定できる最初の日付は 1993 年 1 月 1 日です。デフォルトの *end-time* および **duration** は **infinite** です。
ステップ 7	**end** 例： `Device(config-keychain)# end`	特権 EXEC モードに戻ります。
ステップ 8	**show key chain** 例： `Device# show key chain`	認証キーの情報を表示します。
ステップ 9	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーションファイルに設定を保存します。



第 18 章

VRF-Lite の設定

VRF-Lite について

VRF-Lite の機能によって、サービス プロバイダーは、VPN 間で重複した IP アドレスを使用できる複数の VPN をサポートできます。VRF-Lite は入力インターフェイスを使用して異なる VPN のルートを区別し、各 VRF に 1 つまたは複数のレイヤ 3 インターフェイスを対応付けて仮想パケット転送テーブルを形成します。VRF のインターフェイスは、イーサネット ポートなどの物理インターフェイス、または VLAN SVI などの論理インターフェイスにすることができますが、レイヤ 3 インターフェイスは、一度に複数の VRF に属することはできません。

（注） VRF-Lite インターフェイスは、レイヤ 3 インターフェイスである必要があります。

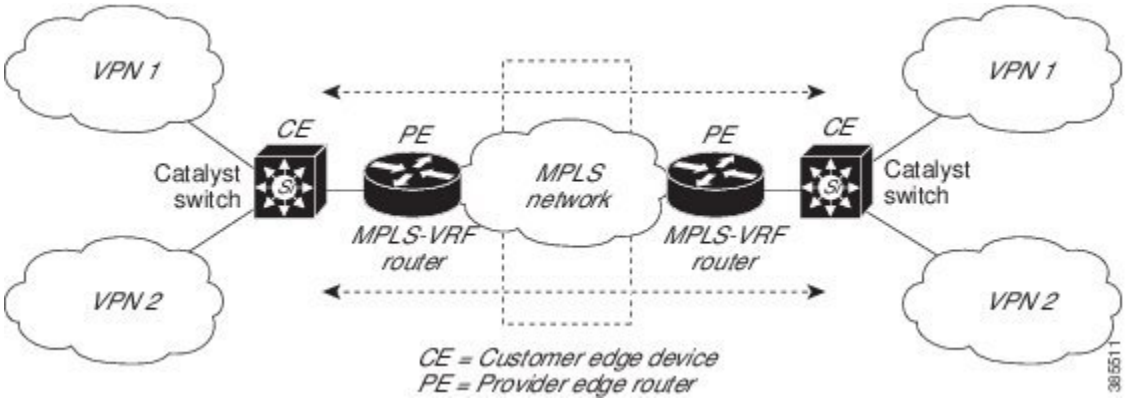
VRF-Lite には次のデバイスが含まれます。

- カスタマーエッジ（CE）デバイスにおいて、カスタマーは、1 つまたは複数の プロバイダー エッジ（PE）ルータへのデータ リンクを介してサービス プロバイダー ネットワークにアクセスできます。CE デバイスは、サイトのローカル ルートをプロバイダー エッジ ルータにアドバタイズし、そこからリモート VPN ルートを学習します。Cisco Catalyst スイッチは、CE にすることができます。
- プロバイダー ルータ（またはコア ルータ）とは、サービス プロバイダー ネットワーク内にあり、CE デバイスに接続していないすべてのルータです。

VRF-lite を使用すると、複数のカスタマーが 1 つの CE を共有できます。また、1 つの物理リンクのみが CE と PE 間に使用されます。共有 CE は、お客様ごとに別々の VRF テーブルを維持し、独自のルーティングテーブルに基づいて、カスタマーごとにパケットをスイッチングまたはルーティングします。VRF-lite は限定された PE の機能を CE デバイスに拡張して、個別の VRF テーブルを保守する機能を付与し、VPN のプライバシーおよびセキュリティをブランチオフィスまで拡張します。

次の図に、各 Cisco Catalyst スイッチが複数の仮想 CE として機能する設定を示します。VRF-Lite はレイヤ 3 機能であるため、VRF の各インターフェイスはレイヤ 3 インターフェイスである必要があります。

図 *11* : 複数の仮想 *CE* として機能する *Cisco Catalyst* スイッチ



次の図に、VRF-Lite の CE 対応ネットワークでのパケット転送プロセスを示します。

- CE が VPN からパケットを受信すると、CE は入力インターフェイスに基づいたルーティング テーブルを検索します。ルートが見つかると、CE はパケットを PE に転送します。
- 入力 PE は、CE からパケットを受信すると、VRF 検索を実行します。ルートが見つかると、ルータは対応する MPLS ラベルをパケットに追加し、MPLS ネットワークに送信します。
- 出力 PE は、ネットワークからパケットを受信すると、ラベルを除去してそのラベルを使用し、正しい VPN ルーティング テーブルを識別します。次に、出力 PE が通常のルート検索を行います。ルートが見つかると、パケットを正しい隣接デバイスに転送します。
- CE が出力 PE からパケットを受信すると、CE は入力インターフェイスを使用して正しい VPN ルーティング テーブルを検索します。ルートが見つかると、CE はパケットを VPN 内に転送します。

VRF を設定するには、VRF テーブルを作成し、VRF に対応付けられたレイヤ 3 インターフェイスを指定します。次に、VPN および CE と PE 間でルーティング プロトコルを設定します。プロバイダーのバックボーンで VPN ルーティング情報を配信する場合は、BGP が優先ルーティング プロトコルです。VRF-Lite ネットワークには、次の 3 つの主要なコンポーネントがあります。

- VPN ルート ターゲット コミュニティ：VPN コミュニティの他のすべてのメンバをリストします。VPN コミュニティ メンバーごとに VPN ルート ターゲットを設定する必要があります。
- VPN コミュニティ PE ルータのマルチプロトコル BGP ピアリング：VPN コミュニティのすべてのメンバーに VRF 到達可能性情報を伝播します。VPN コミュニティのすべての PE ルータで BGP ピアリングを設定する必要があります。
- VPN 転送：VPN サービスプロバイダー ネットワークのすべての VPN コミュニティ メンバ間のすべてのトラフィックを転送します。

VRF-Lite の設定に関するガイドライン

IPv4 と IPv6

- VRF-Lite が設定されたスイッチは複数のカスタマーで共有され、すべてのカスタマーが独自のルーティング テーブルを持ちます。
- カスタマーは別々の VRF テーブルを使用するので、同じ IP アドレスを再利用できます。別々の VPN では IP アドレスの重複が許可されます。
- VRF-Lite では、複数のカスタマーが PE と CE の間で同一の物理リンクを共有できます。複数の VLAN を持つトランク ポートでは、パケットがお客様間で分離されます。すべてのカスタマーが独自の VLAN を持ちます。
- PE ルータでは、VRF-Lite の使用と複数の CE の使用には違いがありません。VRF-Lite について （337 ページ） では、複数の仮想レイヤ 3 インターフェイスが VRF-Lite デバイスに接続されています。
- Cisco Catalyst スイッチでは、物理ポートか VLAN SVI、またはその両方の組み合わせを使用して、VRF を設定できます。アクセス ポートまたはトランク ポート経由で SVI を接続できます。
- カスタマーは、別のカスタマーと重複しないかぎり、複数の VLAN を使用できます。カスタマーの VLAN は、スイッチに保存されている適切なルーティング テーブルの識別に使用される特定のルーティング テーブル ID にマッピングされます。
- レイヤ 3 TCAM リソースは、すべての VRF 間で共有されます。各 VRF が十分な CAM 領域を持つようにするには、**maximum routes** コマンドを使用します。
- VRF を使用した Cisco Catalyst スイッチは、1 つのグローバル ネットワークと複数の VRF をサポートできます。サポートされるルートの総数は、TCAM のサイズに制限されます。
- 1 つの VRF を IPv4 と IPv6 の両方に設定できます。
- 着信パケットの宛先アドレスが VRF テーブルにない場合、そのパケットはドロップされます。また、VRF ルートに TCAM 領域が十分にない場合、その VRF のハードウェアス

イッチングは無効になり、対応するデータ パケットがソフトウェアに送信されて処理されます。

IPv4 固有

- CE と PE 間のほとんどのルーティング プロトコル（BGP、OSPF、EIGRP、RIP、およびスタティック ルーティング）を使用できます。ただし、次の理由から External BGP（EBGP）を使用することを推奨します。
  - BGP では、複数の CE とのやり取りに複数のアルゴリズムを必要としません。
  - BGP は、さまざまな管理者によって稼働するシステム間でルーティング情報を渡すように設計されています。
  - BGP は、ルートの属性の CE への引き渡しを単純化します。
- Cisco Catalyst スイッチでは、PIM-SM プロトコル と PIM-SSM プロトコルがサポートされます。
- **router ospf** の **capability vrf-lite** サブコマンドは、PE と CE 間のルーティング プロトコルとして OSPF が設定されている場合に使用する必要があります。

IPv6 固有

- VRF 認識 OSPFv3、BGPv6、EIGRPv6、および IPv6 スタティックルーティングがサポートされます。
- VRF 認識 IPv6 ルート アプリケーションには、ping、telnet、ssh、tftp、ftp、およびトレースルートが含まれています（このリストには管理インターフェイスは含まれていません。これは、その下に IPv4 も IPv6 も設定できますが、別々に処理されます）。

VRF-Lite の設定方法

ここでは、VRF-Lite の設定について説明します。

IPv4 用の VRF-Lite の設定

ここでは、IPv4 用の VRF-Lite の設定について説明します。

VRF 認識サービスの設定

IP サービスは、グローバルなインターフェイス上と、グローバルなルーティング インスタンス内で設定できます。IP サービスは複数のルーティング インスタンス上で稼働するように拡張されます。これが、VRF 認識です。システム内の任意の設定済み VRF であればいずれも、VRF 認識サービス用に指定できます。

VRF 認識サービスは、プラットフォームから独立したモジュールに実装されています。VRF は、Cisco IOS 内の複数のルーティング インスタンスを提供します。各プラットフォームには、サポートする VRF 数に関して独自の制限があります。

VRF 認識サービスには、次の特性があります。

- ユーザーは、ユーザー指定の VRF 内のホストに ping を実行できます。
- ARP エントリは、個別の VRF で学習されます。ユーザーは、特定の VRF の ARP エントリを表示できます。

ARP のユーザ インターフェイスの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**show ip arp vrf** *vrf-name* 例： `Device# show ip arp vrf vrf-name`	指定された VRF で、ARP テーブル（スタティック エントリおよびダイナミック エントリ）を表示します。
ステップ 2	**arp vrf** *vrf-name ip-address mac-address ARPA* 例： `Device(config)# arp vrf vrf-name ip-address mac-address ARPA`	指定された VRF でスタティック ARP エントリを作成します。

TACACS+ サーバ用の Per-VRF の設定

TACACS+ サーバ機能の per-VRF は TACACS+ サーバの per- 仮想単位ルート転送（per-VRF）の認証、認可、アカウンティング（AAA）を設定することができます。

VRF ルーティング テーブル（ステップ 3 および 4 で示すように）を作成し、インターフェイスを設定する（ステップ 6、7、および 8）ことができます。TACACS+ サーバの per-VRF 単位の実際の設定は、ステップ 10～13 で行われます。

始める前に

TACACS+ サーバの per-VRF を設定する前に、AAA およびサーバ グループを設定しておく必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	**configure terminal** 例： `Device# 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	**vrf definition** *vrf-name* 例： `Device(config)# vrf definition vrf-name`	VRFテーブルを設定し、VRFコンフィギュレーション モードを開始します。
ステップ 4	**rd** *route-distinguisher* 例： `Device(config-vrf)# rd route-distinguisher`	VRFインスタンスに対するルーティングおよびフォワーディング テーブルを作成します。
ステップ 5	**exit** 例： `Device(config-vrf)# exit`	VRFコンフィギュレーションモードを終了します。
ステップ 6	**interface** *interface-name* 例： `Device(config)# interface interface-name`	インターフェイスを設定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 7	**vrf forwarding** *vrf-name* 例： `Device(config-if)# vrf forwarding vrf-name`	インターフェイスに VRF を設定します。
ステップ 8	**ip address** *ip-address mask* [**secondary**] 例： `Device(config-if)# ip address ip-address mask [secondary]`	インターフェイスに対するプライマリ IP アドレスまたはセカンダリ IP アドレスを設定します。
ステップ 9	**exit** 例： `Device(config-vrf)# exit`	インターフェイス コンフィギュレーション モードを終了します。
ステップ 10	**aaa group server tacacs+** *group-name* 例： `Device(config)# aaa group server tacacs+ tacacs1`	異なる TACACS+ サーバ ホストを別々のリストと方式にグループ化し、server-group 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	**server-private** {*ip-address* | *name*} [**nat**] [**single-connection**] [**port** *port-number*] [**timeout** *seconds*] [**key** [**0** | **7**] *string*] 例： `Device(config-sg-tacacs+)# server-private 10.1.1.1 port 19 key cisco`	グループ サーバに対するプライベート TACACS+ サーバの IP アドレスを設定します。

	コマンドまたはアクション	目的
ステップ 12	**vrf forwarding** *vrf-name* 例： `Device(config-sg-tacacs+)# vrf forwarding vrf-name`	AAA TACACS+ サーバ グループの VRF リファレンスを設定します。
ステップ 13	**ip tacacs source-interface** *subinterface-name* 例： `Device(config-sg-tacacs+)# ip tacacs source-interface subinterface-name`	すべての発信 TACACS+ パケットに対して、指定されたインターフェイスの IP アドレスを使用します。
ステップ 14	**exit** 例： `Device(config-sg-tacacs)# exit`	server-group コンフィギュレーション モードを終了します。

例

次の例で、per-VRF TACACS+ の設定に必要なすべての手順をリストします。

```
Device> enable
Device# configure terminal
Device(config)# vrf definition cisco
Device(config-vrf)# rd 100:1
Device(config-vrf)# exit
Device(config)# interface Loopback0
Device(config-if)# vrf forwarding cisco
Device(config-if)# ip address 10.0.0.2 255.0.0.0
Device(config-if)# exit
Device(config-sg-tacacs+)# vrf forwarding cisco
Device(config-sg-tacacs+)# ip tacacs source-interface Loopback0
Device(config-sg-tacacs)# exit
```

マルチキャスト VRF の設定

手順の概要

1. **configure terminal**
2. **ip routing**
3. **vrf definition** *vrf-name*
4. **ip multicast-routing vrf** *vrf-name*
5. **rd** *route-distinguisher*
6. **route-target** {**export** | **import** | **both**} *route-target-ext-community*
7. **import map** ルート マップ
8. **interface** *interface-id*
9. **vrf forwarding** *vrf-name*
10. **ip address** *ip-address***mask**
11. **ip pim sparse-mode**
12. **end**

13. **show vrf definition** [**brief** | **detail** | **interfaces**] [*vrf-name*]
14. **copy running-config startup-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	**configure terminal** 例： `Device# configure terminal`	グローバル設定モードを開始します。
ステップ 2	**ip routing** 例： `Device(config)# ip routing`	IP ルーティングを有効にします。
ステップ 3	**vrf definition** *vrf-name* 例： `Device(config)# vrf definition vrf-name`	VRF テーブルを設定し、VRF コンフィギュレーション モードを開始します。
ステップ 4	**ip multicast-routing vrf** *vrf-name* 例： `Device(config-vrf)# ip multicast-routing vrf vrf-name`	（任意）VRF テーブルでグローバル マルチキャスト ルーティングをイネーブルにします。
ステップ 5	**rd** *route-distinguisher* 例： `Device(config-vrf)# rd route-distinguisher`	ルート識別子を指定して VRF テーブルを作成します。自律システム（AS）番号および任意の数（xxx:y）または IP アドレスおよび任意の数（A.B.C.D:y）のどちらかを入力します。
ステップ 6	**route-target** {**export** | **import** | **both**} *route-target-ext-community* 例： `Device(config-vrf)# route-target {export | import | both} route-target-ext-community`	指定された VRF のインポート、エクスポート、またはインポートおよびエクスポート ルート ターゲット コミュニティのリストを作成します。AS システム番号と任意の番号（xxx:y）または IP アドレスと任意の番号（A.B.C.D:y）を入力します。 ルート ターゲット ext コミュニティ値は、ステップ 4 で入力した route-distinguisher 値と同じです。
ステップ 7	**import map** ルート マップ 例： `Device(config-vrf)# import map route-map`	（任意）VRF にルート マップを対応付けます。
ステップ 8	**interface** *interface-id* 例： `Device(config)# interface interface-id`	インターフェイス コンフィギュレーション モードを開始して、VRF に対応付けるレイヤ 3 インターフェイスを指定します。有効なインターフェイスは、ルーテッド ポートまたは SVI です。

	コマンドまたはアクション	目的
ステップ 9	**vrf forwarding** *vrf-name* 例： `Device(config-if)# vrf forwarding vrf-name`	VRF をレイヤ3インターフェイスに対応付けます。
ステップ 10	**ip address** *ip-address***mask** 例： `Device(config-if)# ip address ip-address mask`	レイヤ 3 インターフェイスの IP アドレスを設定します。
ステップ 11	**ip pim sparse-mode** 例： `Device(config-if)# ip pim sparse-mode`	VRF に関連付けられているレイヤ 3 インターフェイス上で、PIM をイネーブルにします。
ステップ 12	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。
ステップ 13	**show vrf definition** [**brief** | **detail** | **interfaces**] [*vrf-name*] 例： `Device# show vrf definition brief`	設定を確認します。設定した VRF に関する情報を表示します。
ステップ 14	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

例

次に、VRF テーブル内にマルチキャストを設定する例を示します。

```
Device(config)# ip routing
Device(config)# vrf definition multiVrfA
Device(config-vrf)# ip multicast-routing vrf multiVrfA
Device(config-vrf)# interface GigabitEthernet3/1/0
Device(config-if)# vrf forwarding multiVrfA
Device(config-if)# ip address 172.21.200.203 255.255.255.0
Device(config-if)# ip pim sparse-mode
```

VPN ルーティング セッションの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例：	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
	`Device# configure terminal`	
ステップ 2	**router ospf** *process-id* **vrf vrf-name** 例： `Device(config)# router ospf process-id vrf vrf-name`	OSPF ルーティングを有効にし、VPN 転送テーブルを指定して、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 3	**log-adjacency-changes** 例： `Device(config-router)# log-adjacency-changes`	（任意）隣接状態（デフォルト）の変更を記録します。
ステップ 4	**redistribute bgp** *autonomous-system-number subnets* 例： `Device(config-router)# redistribute bgp autonomous-system-number subnets`	BGP ネットワークから OSPF ネットワークに情報を再配布するようにスイッチを設定します。
ステップ 5	**network** *network-number* **area** *area-id* 例： `Device(config-router)# network network-number area area-id`	OSPF が動作するネットワーク アドレスとマスク、およびそのネットワーク アドレスのエリア ID を定義します。
ステップ 6	**end** 例： `Device(config-router)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show ip ospf** *process-id* 例： `Device# show ip ospf process-id`	OSPF ネットワークの設定を確認します。
ステップ 8	copy running-config startup-config 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。 VPN 転送テーブルと OSPF ルーティング プロセスの関連付けを解除するには、**no router ospf** *process-id* **vrf** *vrf-name* グローバル コンフィギュレーション コマンドを使用します。

例

```
Device(config)# vrf definition VRF-RED
Device(config-vrf)# rd 1:1
Device(config-vrf)# exit
Device(config)# router eigrp virtual-name
Device(config-router)# address-family ipv4 vrf VRF-RED autonomous-system 1
Device(config-router-af)# network 10.0.0.0 0.0.0.255
Device(config-router-af)# topology base
Device(config-router-topology)# default-metric 10000 100 255 1 1500
```

```
Device(config-router-topology)# exit-af-topology
Device(config-router-af)# exit-address-family
```

BGP PE/CE ルーティング セッションの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# 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 2	**router bgp** *autonomous-system-number* 例： `Device(config)# router bgp autonomous-system-number`	その他の BGP ルータに渡された AS 番号で BGP ルーティング プロセスを設定し、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 3	**network** *network-number* **mask** *network-mask* 例： `Device(config-router)# network network-number mask network-mask`	BGP を使用してアナウンスするネットワークおよびマスクを指定します。
ステップ 4	**redistribute ospf** *process-id match internal* 例： `Device(config-router)# redistribute ospf process-id match internal`	OSPF 内部ルートを再配布するようにスイッチを設定します。
ステップ 5	**network** *network-number* **area** *area-id* 例： `Device(config-router)# network network-number area area-id`	OSPF が動作するネットワーク アドレスとマスク、およびそのネットワーク アドレスのエリア ID を定義します。
ステップ 6	**address-family ipv4 vrf** *vrf-name* 例： `Device(config-router-af)# address-family ipv4 vrf vrf-name`	PE から CE のルーティング セッションの BGP パラメータを定義し、VRF アドレス ファミリ モードを開始します。
ステップ 7	**neighbor** *address* **remote-as** *as-number* 例： `Device(config-router-af)# neighbor address remote-as as-number`	PE と CE ルータの間の BGP セッションを定義します。
ステップ 8	**neighbor** *address* **activate** 例： `Device(config-router-af)# neighbor address activate`	IPv4 アドレス ファミリのアドバタイズメントをアクティブ化します。

	コマンドまたはアクション	目的
ステップ 9	**end** 例： `Device(config-router-af)# end`	特権 EXEC モードに戻ります。
ステップ 10	**show ip bgp** [**ipv4**] [**neighbors**] 例： `Device# show ip bgp [ipv4] [neighbors]`	BGP 設定を確認します。 BGP ルーティングプロセスを削除するには、**no router bgp** *autonomous-system-number* グローバル コンフィギュレーション コマンドを使用します。ルーティング特性を削除するには、コマンドにキーワードを指定してこのコマンドを使用します。

IPv4 VRF の設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# 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 2	**ip routing** 例： `Device# 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	**vrf definition** *vrf-name* 例： `Device(config)# vrf definition vrf-name`	VRF 名を指定し、VRF コンフィギュレーション モードを開始します。
ステップ 4	**rd** *route-distinguisher* 例： `Device(config-vrf)# rd route-distinguisher`	ルート識別子を指定して VRF テーブルを作成します。自律システム番号と任意の数値（xxx:y）、またはIPアドレスと任意の数値（A.B.C.D:y）のいずれかを入力します。
ステップ 5	**route-target** {**export** | **import** | **both**} *route-target-ext-community* 例： `Device(config-vrf)# route-target {export | import | both} route-target-ext-community`	指定された VRF のインポート、エクスポート、またはインポートおよびエクスポートルートターゲットコミュニティのリストを作成します。AS システム番号と任意の番号（xxx:y）またはIP アドレスと任意の番号（A.B.C.D:y）を入力します。 （注） このコマンドは、BGP が動作している場合にのみ有効です。

	コマンドまたはアクション	目的
ステップ 6	**import map** ルート マップ 例： `Device(config-vrf)# import map route-map`	（任意）VRF にルート マップを対応付けます。
ステップ 7	**interface** *interface-id* 例： `Device(config-vrf)# interface interface-id`	インターフェイス コンフィギュレーション モードを開始して、VRF に対応付けるレイヤ 3 インターフェイスを指定します。インターフェイスにはルーテッド ポートまたは SVI を設定できます。
ステップ 8	**vrf forwarding** *vrf-name* 例： `Device(config-if)# vrf forwarding vrf-name`	VRF をレイヤ3インターフェイスに対応付けます。
ステップ 9	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。
ステップ 10	**show vrf definition** [**brief** | **detail** | **interfaces**] [*vrf-name*] 例： `Device# show vfr definition [brief | detail | interfaces] [vrf-name]`	設定を確認します。設定した VRF に関する情報を表示します。
ステップ 11	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。 VRF とそのすべてのインターフェイスを削除するには、**no vrf definition***vrf-name* グローバル コンフィギュレーション コマンドを使用します。VRF からインターフェイスを削除するには、**no vrf forwarding** インターフェイス コンフィギュレーション コマンドを使用します。

IPv6 用の VRF-Lite の設定

ここでは、IPv6 用の VRF-Lite の設定について説明します。

VRF 認識サービスの設定

IPv6 サービスは、グローバルなインターフェイス上と、グローバルなルーティング インスタンス内で設定できます。IPv6 サービスは複数のルーティング インスタンス上で稼働するように拡張されます。これが、VRF 認識です。システム内の任意の設定済み VRF であればいずれも、VRF 認識サービス用に指定できます。

VRF 認識サービスは、プラットフォームから独立したモジュールに実装されています。VRF は、Cisco IOS 内の複数のルーティング インスタンスを提供します。各プラットフォームには、サポートする VRF 数に関して独自の制限があります。

VRF 認識サービスには、次の特性があります。

- ユーザーは、ユーザー指定の VRF 内のホストに ping を実行できます。
- ネイバー探索エントリは、個別の VRF で学習されます。ユーザは、特定の VRF のネイバー探索（ND）エントリを表示できます。

次のサービスは VRF 認識です。

- Ping
- ユニキャスト RPF（uRPF）
- traceroute
- FTP および TFTP
- [Telnet および SSH（Telnet and SSH）]
- NTP

PING のユーザ インターフェイスの設定

VRF 認識 ping を設定するには、次の作業を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**ping vrf** *vrf-name* **ipv6-host** 例： `Device# ping vrf vrf-name ipv6-host`	指定された VRF で、IPv6 ホストまたはアドレスに対して ping を実行します。

uRPF のユーザ インターフェイスの設定

VRF に割り当てられているインターフェイス上で、uRPF を設定できます。送信元の検索が VRF テーブルで実行されます。

手順の概要

1. **configure terminal**
2. **interface** *interface-id*
3. **no switchport**
4. **vrf forwarding** *vrf-name*
5. **ipv6 address** *ip-address***subnet-mask**
6. **ipv6 verify unicast source reachable-via rx allow-default**
7. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# 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 2	**interface** *interface-id* 例： `Device(config)# interface interface-id`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 3	**no switchport** 例： `Device(config-if)# no switchport`	レイヤ 2 コンフィギュレーション モードからインターフェイスを削除します（物理インターフェイスの場合）。
ステップ 4	**vrf forwarding** *vrf-name* 例： `Device(config-if)# vrf forwarding vrf-name`	インターフェイス上で VRF を設定します。
ステップ 5	**ipv6 address** *ip-address***subnet-mask** 例： `Device(config-if)# ip address ip-address mask`	インターフェイスの IPv6 アドレスを入力します。
ステップ 6	**ipv6 verify unicast source reachable-via rx allow-default** 例： `Device(config-if)# ipv6 verify unicast source reachable-via rx allow-default`	インターフェイス上で uRPF を有効にします。
ステップ 7	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。

Traceroute のユーザ インターフェイスの設定

手順の概要

1. **traceroute vrf** *vrf-name ipv6address*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**traceroute vrf** *vrf-name ipv6address* 例： `Device# traceroute vrf vrf-name ipv6address`	宛先アドレスを取得する VPN VRF の名前を指定します。

Telnet および SSH のユーザ インターフェイスの設定

手順の概要

1. **telnet** *ipv6-address*/ **vrf** *vrf-name*
2. **ssh -l** *username* **-vrf** *vrf-name* **ipv6-host**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**telnet** *ipv6-address*/ **vrf** *vrf-name* 例： `Device# telnet ipv6-address/vrf vrf-name`	指定された VRF で、IPv6 ホストまたはアドレスに Telnet 経由で接続します。
ステップ 2	**ssh -l** *username* **-vrf** *vrf-name* **ipv6-host** 例： `Device# ssh -l username -vrf  vrf-name ipv6-host`	指定された VRF で、IPv6 ホストまたはアドレスに SSH 経由で接続します。

NTP のユーザ インターフェイスの設定

手順の概要

1. **configure terminal**
2. **ntp server vrf** *vrf-name* **ipv6-host**
3. **ntp peer vrf** *vrf-name* **ipv6-host**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# 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 2	**ntp server vrf** *vrf-name* **ipv6-host** 例： `Device(config)# ntp server vrf vrf-name ipv6-host`	指定された VRF で NTP サーバを設定します。
ステップ 3	**ntp peer vrf** *vrf-name* **ipv6-host** 例： `Device(config)# ntp peer vrf vrf-name ipv6-host`	指定された VRF で NTP ピアを設定します。

IPv6 VRF の設定

手順の概要

1. **configure terminal**
2. **vrf definition** *vrf-name*
3. **rd** *route-distinguisher*
4. **address-family** *ipv4* | *ipv6*
5. **route-target** {**export** | **import** | **both**} *route-target-ext-community*
6. **exit-address-family**
7. **vrf definition** *vrf-name*
8. **ipv6 multicast multitopology**
9. **address-family ipv6 multicast**
10. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# 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 2	**vrf definition** *vrf-name* 例： `Device(config)# vrf definition vrf-name`	VRF 名を指定し、VRF コンフィギュレーション モードを開始します。
ステップ 3	**rd** *route-distinguisher* 例： `Device(config-vrf)# rd route-distinguisher`	（任意）ルート識別子を指定して VRF テーブルを作成します。自律システム番号および任意の数（xxx:y）、または IP アドレスおよび任意の数（A.B.C.D:y）のいずれかを入力します。
ステップ 4	**address-family** *ipv4* | *ipv6* 例： `Device(config-vrf)# address-family ipv4 | ipv6`	（任意）デフォルトは IPv4 です。IPv6 の必須設定。
ステップ 5	**route-target** {**export** | **import** | **both**} *route-target-ext-community* 例： `Device(config-vrf)# route-target {export | import | both} route-target-ext-community`	指定された VRF のインポート、エクスポート、またはインポートおよびエクスポート ルート ターゲット コミュニティのリストを作成します。AS システム番号と任意の番号（xxx:y）または IP アドレスと任意の番号（A.B.C.D:y）を入力します。 （注） このコマンドは、BGP が動作している場合にのみ有効です。

	コマンドまたはアクション	目的
ステップ 6	**exit-address-family** 例： `Device(config-vrf)# exit-address-family`	VRF 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、VRF コンフィギュレーション モードに戻ります。
ステップ 7	**vrf definition** *vrf-name* 例： `Device(config)# vrf definition vrf-name`	VRF コンフィギュレーション モードを開始します。
ステップ 8	**ipv6 multicast multitopology** 例： `Device(config-vrf-af)# ipv6 multicast multitopology`	マルチキャスト固有の RPF トポロジを有効にします。
ステップ 9	**address-family ipv6 multicast** 例： `Device(config-vrf)# address-family ipv6 multicast`	マルチキャスト IPv6 アドレス ファミリを入力します。
ステップ 10	**end** 例： `Device(config-vrf-af)# end`	特権 EXEC モードに戻ります。

例

次に、VRF を設定する例を示します。

```
Device(config)#  vrf definition red
Device(config-vrf)# rd 100:1
Device(config-vrf)# address family ipv6
Device(config-vrf-af)# route-target both 200:1
Device(config-vrf)# exit-address-family
Device(config-vrf)# vrf definition red
Device(config-vrf)# ipv6 multicast multitopology
Device(config-vrf)# address-family ipv6 multicast
Device(config-vrf-af)# end
```

定義済み VRF へのインターフェイスの関連付け

手順の概要

1. **interface** *interface-id*
2. **no switchport**
3. **vrf forwarding** *vrf-name*
4. **ipv6 enable**
5. **ipv6 address** *ip-address subnet-mask*
6. **show ipv6 vrf** [**brief** | **detail** | **interfaces**] [*vrf-name*]

7. **copy running-config startup-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	**interface** *interface-id* 例： `Device(config-vrf)# interface interface-id`	インターフェイス コンフィギュレーション モードを開始して、VRF に対応付けるレイヤ 3 インターフェイスを指定します。インターフェイスにはルーテッド ポートまたは SVI を設定できます。
ステップ 2	**no switchport** 例： `Device(config-if)# no switchport`	コンフィギュレーションモードからインターフェイスを削除します（物理インターフェイスの場合）。
ステップ 3	**vrf forwarding** *vrf-name* 例： `Device(config-if)# vrf forwarding vrf-name`	VRF をレイヤ 3 インターフェイスに対応付けます。
ステップ 4	**ipv6 enable** 例： `Device(config-if)# ipv6 enable`	インターフェイスで IPv6 を有効にします。
ステップ 5	**ipv6 address** *ip-address subnet-mask* 例： `Device(config-if)# ipv6 address ip-address subnet-mask`	インターフェイスの IPv6 アドレスを入力します。
ステップ 6	**show ipv6 vrf** [**brief** | **detail** | **interfaces**] [*vrf-name*] 例： `Device# show ipv6 vrf [brief | detail | interfaces] [vrf-name]`	設定を確認します。設定した VRF に関する情報を表示します。
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーションファイルに設定を保存します。

例

次に、インターフェイスを VRF に関連付ける例を示します。

```
Switch(config-vrf)# interface ethernet0/1
Switch(config-if)# vrf forwarding red
Switch(config-if)# ipv6 enable
Switch(config-if)# ipv6 address 5000::72B/64
```

ルーティング プロトコル経由での VRF へのルートの入力

ここでは、ルーティングプロトコル経由での VRF へのルートの入力について説明します。

VRF スタティック ルートの設定

手順の概要

1. **configure terminal**
2. **ipv6 route** [**vrf** *vrf-name*] *ipv6-prefix*/*prefix-length* {*ipv6-address* | **interface-type** *interface-number* [*ipv6-address*]}

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# 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 2	**ipv6 route** [**vrf** *vrf-name*] *ipv6-prefix*/*prefix-length* {*ipv6-address*	**interface-type** *interface-number* [*ipv6-address*]} 例： `Device(config)# ipv6 route [vrf vrf-name] ipv6-prefix/prefix-length {ipv6-address	interface-type interface-number [ipv6-address]}`	VRF に固有のスタティック ルートを設定します。

例

```
Device(config)# ipv6 route vrf v6a 7000::/64 TenGigabitEthernet32 4000::2
```

OSPFv3 ルータ プロセスの設定

手順の概要

1. **configure terminal**
2. **router ospfv3** *process-id*
3. **area** *area-ID* [**default-cot** | **nssa** | **stub**]
4. **router-id** *router-id*
5. **address-family ipv6 unicast vrf** *vrf-name*
6. **redistribute source-protocol** [*process-id*] **options**
7. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# 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 2	**router ospfv3** *process-id* 例： `Device(config)# router ospfv3 process-id`	IPv6 アドレス ファミリの OSPFv3 ルータ コンフィギュレーション モードを有効にします。
ステップ 3	**area** *area-ID* [**default-cot** | **nssa** | **stub**] 例： `Device(config-router)# area area-ID [default-cot | nssa | stub]`	OSPFv3 エリアを設定します。
ステップ 4	**router-id** *router-id* 例： `Device(config-router)# router-id router-id`	固定ルータ ID を使用します。
ステップ 5	**address-family ipv6 unicast vrf** *vrf-name* 例： `Device(config-router)# address-family ipv6 unicast vrf vrf-name`	vrf vrf-name の OSPFv3 の IPv6 アドレス ファミリ コンフィギュレーション モードを開始します。
ステップ 6	**redistribute source-protocol** [*process-id*] **options** 例： `Device(config-router)# redistribute source-protocol [process-id] options`	あるルーティング ドメインから別のルーティング ドメインへ IPv6 ルートを再配布します。
ステップ 7	**end** 例： `Device(config-router)# end`	特権 EXEC モードに戻ります。

例

次に、OSPFv3 ルータ プロセスを設定する例を示します。

```
Device(config-router)# router ospfv3 1
Device(config-router)# router-id 1.1.1.1
Device(config-router)# address-family ipv6 unicast
Device(config-router-af)# exit-address-family
```

インターフェイス上での OSPFv3 の有効化

手順の概要

1. **configure terminal**
2. **interface** *type-number*
3. **ospfv3** *process-id* **area** *area-id* **ipv6** [**instance** *instance-id*]
4. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# 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 2	**interface** *type-number* 例： `Device(config-vrf)# interface type-number`	インターフェイスのタイプと番号を指定し、スイッチをインターフェイス コンフィギュレーション モードにします。
ステップ 3	**ospfv3** *process-id* **area** *area-id* **ipv6** [**instance** *instance-id*] 例： `Device(config-if)# ospfv3 process-id area area-ID ipv6 [instance instance-id]`	IPv6 AF を設定したインターフェイスで OSPFv3 を有効にします。
ステップ 4	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。

例

次に、インターフェイス上で OSPFv3 を有効にする例を示します。

```
Device(config)# interface GigabitEthernet2/1
Device(config-if)# no switchport
Device(config-if)# ipv6 address 4000::2/64
Device(config-if)# ipv6 enable
Device(config-if)# ipv6 ospf 1 area 0
Device(config-if)# end
```

EIGRPv6 ルーティング プロセスの設定

手順の概要

1. **configure terminal**
2. **router eigrp** *virtual-instance-name*
3. **address-family ipv6 vrf** *vrf-name* **autonomous-system** *autonomous-system-number*

4. **topology** {**base** | **topology-name tid number**
5. **exit-aftopology**
6. **eigrp router-id** *ip-address*
7. end

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# 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 2	**router eigrp** *virtual-instance-name* 例： `Device(config)# router eigrp virtual-instance-name`	EIGRP ルーティング プロセスを設定し、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 3	**address-family ipv6 vrf** *vrf-name* **autonomous-system** *autonomous-system-number* 例： `Device(config-router)# address-family ipv6 vrf vrf-name autonomous-system autonomous-system-number`	EIGRP IPv6 VRF-Lite を有効にし、アドレス ファミリ コンフィギュレーション モードを開始します。
ステップ 4	**topology** {**base**	**topology-name tid number** 例： `Device(config-router-af)# topology {base | topology-name tid number`	指定されたトポロジ インスタンスで IP トラフィックをルーティングするよう EIGRP プロセスを設定し、アドレス ファミリ トポロジ コンフィギュレーション モードを開始します。
ステップ 5	**exit-aftopology** 例： `Device(config-router-af-topology)# exit-aftopology`	アドレス ファミリ トポロジ コンフィギュレーション モードを終了します。
ステップ 6	**eigrp router-id** *ip-address* 例： `Device(config-router)# eigrp router-id ip-address`	固定ルータ ID の使用を有効にします。
ステップ 7	end 例： `Device(config-router)# end`	ルータ コンフィギュレーション モードを終了します。

例

次に、EIGRP ルーティング プロセスを設定する例を示します。

```
Device(config)# router eigrp test
```

```
Device(config-router)# address-family ipv6 unicast vrf b1 autonomous-system 10
Device(config-router-af)# topology base
Device(config-router-af-topology)# exit-af-topology
Device(config-router)# eigrp router-id 2.3.4.5
Device(config-router)# exit-address-family
```

EBGPv6 ルーティング プロセスの設定

手順の概要

1. **configure terminal**
2. **router bgp** *as-number*
3. **neighbor** *peer-group-name peer-group*
4. **neighbor** {*ip-address* | *ipv6-address*[*%*] | *peer-group-name*} **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number ...*]
5. **address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast** | **vpnv6**]
6. **neighbor** *ipv6-address* **peer-group** *peer-group-name*
7. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*[*%*]} **route-map** *map-name* {**in** | **out**}
8. exit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# 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 2	**router bgp** *as-number* 例： `Device(config)# router bgp as-number`	指定したルーティング プロセスのルータ コンフィギュレーション モードを開始します。
ステップ 3	**neighbor** *peer-group-name peer-group* 例： `Device(config-router)# neighbor peer-group-name peer-group`	マルチプロトコル BGP ピア グループを作成します。
ステップ 4	**neighbor** {*ip-address* | *ipv6-address*[*%*] | *peer-group-name*} **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number ...*] 例： `Device(config-router)# neighbor {ip-address | ipv6-address[%] | peer-group-name}remote-as autonomous-system-number [alternate-as autonomous-system-number ...]`	指定した自律システム内のネイバーの IPv6 アドレスを、ローカル ルータの IPv6 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 ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast** | **vpnv6**] 例：	IPv6 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。

	コマンドまたはアクション	目的
	`Device(config-router)# address-family ipv6 [vrf vrf-name] [unicast | multicast | vpnv6]`	• unicast キーワードは、IPv6 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 ipv6 コマンドでユニキャスト キーワードが指定されていない場合、スイッチは IPv6 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、IPv6 マルチキャスト アドレス プレフィックスを指定します。
ステップ 6	**neighbor** *ipv6-address* **peer-group** *peer-group-name* 例： `Device(config-router-af)# neighbor ipv6-address peer-group peer-group-name`	BGP ネイバーの IPv6 アドレスをピア グループに割り当てます。
ステップ 7	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*[%]} **route-map** *map-name* {**in** | **out**} 例： `Device(config-router-af)# neighbor {ip-address | peer-group-name | ipv6-address[%]}route-map map-name {in | out}`	着信ルートまたは発信ルートにルートマップを適用します。ルートマップへの変更は、ピアリングがリセットされるまで、またはソフトリセットが実行されるまで、現在のピアでは有効になりません。soft キーワードと in キーワードを指定して clear bgp ipv6 コマンドを使用すると、ソフトリセットが実行されます。
ステップ 8	exit 例： `Device(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、ルータをルータ コンフィギュレーション モードに戻します。

例

次に、EBGPv6 を設定する例を示します。

```
Device(config)# router bgp 2
Device(config-router)# bgp router-id 2.2.2.2
Device(config-router)# bgp log-neighbor-changes
Device(config-router)# no bgp default ipv4-unicast
Device(config-router)# neighbor 2500::1 remote-as 1
Device(config-router)# neighbor 4000::2 remote-as 3
Device(config-router)# address-family ipv6 vrf b1
Device(config-router-af)# network 2500::/64
Device(config-router-af)# network 4000::/64
Device(config-router-af)# neighbor 2500::1 remote-as 1
Device(config-router-af)# neighbor 2500::1 activate
Device(config-router-af)# neighbor 4000::2 remote-as 3
Device(config-router-af)# neighbor 4000::2 activate
Device(config-router-af)# exit-address-family
```

VRF-Lite に関する追加情報

ここでは、VRF-Lite に関する追加情報を提供します。

IPv4 と IPv6 間での VPN の共存

IPv4 を設定するための「以前の」CLI と、IPv6 用の「新しい」CLI 間には下位互換性があります。つまり、設定に両方の CLI を含めることができます。IPv4 CLI は、同じインターフェイス上で、VRF 内で定義されている IP アドレスとともにグローバル ルーティング テーブルで定義されている IPv6 アドレスも備える機能を保持しています。

次に例を示します。

```
vrf definition red
 rd 100:1
 address family ipv6
 route-target both 200:1
 exit-address-family
!
vrf definition blue
 rd 200:1
 route-target both 200:1
!
interface Ethernet0/0
 vrf forwarding red
 ip address 50.1.1.2 255.255.255.0
 ipv6 address 4000::72B/64
!
interface Ethernet0/1
 vrf forwarding blue
 ip address 60.1.1.2 255.255.255.0
 ipv6 address 5000::72B/64
```

この例では、Ethernet0/0 用に定義されたすべてのアドレス（v4 と v6）が VRF red を参照します。Ethernet0/1 については、IP アドレスは VRF blue を参照しますが、ipv6 アドレスはグローバル IPv6 アドレス ルーティング テーブルを参照します。

VRF-Lite 設定の確認

ここでは、VRF-Lite 設定を確認する手順について説明します。

IPv4 VRF-Lite ステータスの表示

VRF-Lite の設定およびステータスに関する情報を表示するには、次の作業のいずれかを行います。

コマンド	目的
`Device# show ip protocols vrf vrf-name`	VRF に対応付けられたルーティング プロトコル情報を表示します。

コマンド	目的
Device# **show ip route vrf** *vrf-name* [**connected**] [*protocol* [*as-number*]] [**list**] [**mobile**] [**odr**] [**profile**] [**static**] [**summary**][**supernets-only**]	VRF に対応付けられた IP ルーティング テーブル情報を表示します。
Device# **show vrf definition** [**brief** | **detail** | **interfaces**] [*vrf-name*]	定義された VRF インスタンスに関する情報を表示します。
Device# **bidir vrf** *instance-name a.b.c.d* | **active** | **bidriectional**| **count** | **interface** | **proxy** | **pruned** | **sparse** | **ssm** | **static** | **summary**	定義された VRF インスタンスに関する情報を表示します。

次に、VRF インスタンス内のマルチキャスト ルート テーブル情報を表示する例を示します。

```
Switch# show ip mroute 226.0.0.2
IP Multicast Routing Table
Flags: S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
       N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
       Q - Received BGP S-A Route, q - Sent BGP S-A Route,
       V - RD & Vector, v - Vector, p - PIM Joins on route,
       x - VxLAN group, c - PFP-SA cache created entry
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 226.0.0.2), 00:01:17/stopped, RP 1.11.1.1, flags: SJCF
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Vlan100, Forward/Sparse, 00:01:17/00:02:36

(5.0.0.11, 226.0.0.2), 00:01:17/00:01:42, flags: FT
  Incoming interface: Vlan5, RPF nbr 0.0.0.0
  Outgoing interface list:
    Vlan100, Forward/Sparse, 00:01:17/00:02:36
```

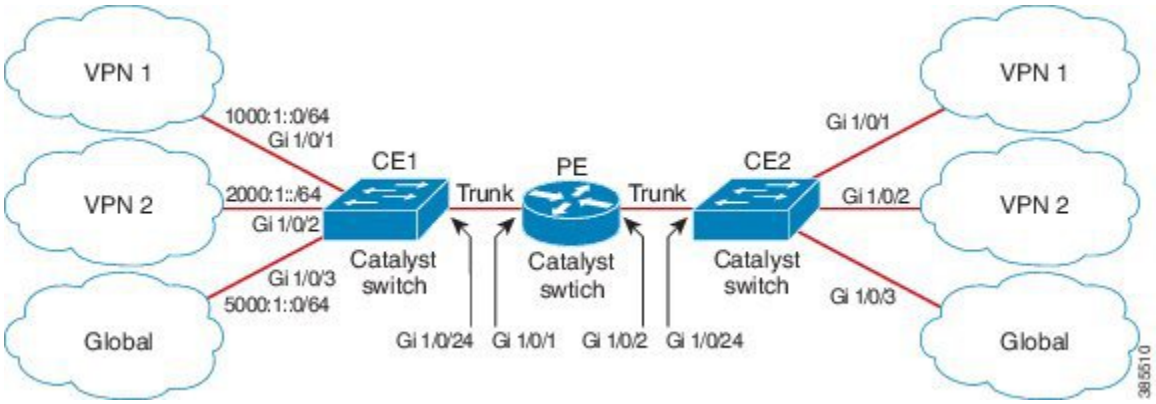
VRF-Lite の設定例

ここでは、VRF-Lite の設定例を示します。

IPv6 VRF-Lite の設定例

次に、CE-PE ルーティングに OSPFv3 を使用するトポロジを示します。

図 *12 : VRF-Lite* の設定例



CE1 スイッチの設定

```
ipv6 unicast-routing
vrf definition v1
 rd 100:1
 !
address-family ipv6
 exit-address-family
!

vrf definition v2
 rd 200:1
 !
address-family ipv6
 exit-address-family
!

interface Vlan100
 vrf forwarding v1
 ipv6 address 1000:1::1/64
 ospfv3 100 ipv6 area 0
!

interface Vlan200
 vrf forwarding v2
 ipv6 address 2000:1::1/64
 ospfv3 200 ipv6 area 0
!

interface GigabitEthernet 1/0/1
switchport access vlan 100
end

interface GigabitEthernet 1/0/2
switchport access vlan 200
end

interface GigabitEthernet 1/0/24
 switchport trunk encapsulation dot1q

switchport mode trunk
end

router ospfv3 100
 router-id 10.10.10.10
```

```
 !
 address-family ipv6 unicast vrf v1
  redistribute connected
  area 0 normal
 exit-address-family
!

router ospfv3 200
 router-id 20.20.20.20
 !
 address-family ipv6 unicast vrf v2
  redistribute connected
  area 0 normal
 exit-address-family
!
```

PE スイッチの設定

```
ipv6 unicast-routing

vrf definition v1
 rd 100:1
 !
address-family ipv6
 exit-address-family
!

vrf definition v2
 rd 200:1
 !
address-family ipv6
 exit-address-family
!
interface Vlan600
 vrf forwarding v1
 no ipv6 address
 ipv6 address 1000:1::2/64
 ospfv3 100 ipv6 area 0
!

interface Vlan700
 vrf forwarding v2
 no ipv6 address
 ipv6 address 2000:1::2/64
 ospfv3 200 ipv6 area 0
!

interface Vlan800
 vrf forwarding v1
 ipv6 address 3000:1::7/64
 ospfv3 100 ipv6 area 0
!
interface Vlan900
 vrf forwarding v2
 ipv6 address 4000:1::7/64
 ospfv3 200 ipv6 area 0
!

interface GigabitEthernet 1/0/1
 switchport trunk encapsulation dot1q
 switchport mode trunk
 exit

interface GigabitEthernet 1/0/2
```

```
 switchport trunk encapsulation dot1q

switchport mode trunk
 exit

router ospfv3 100
 router-id 30.30.30.30
 !
 address-family ipv6 unicast vrf v1
  redistribute connected
  area 0 normal
 exit-address-family
 !
address-family ipv6 unicast vrf v2
  redistribute connected
  area 0 normal
 exit-address-family
 !
```

CE2 スイッチの設定

```
ipv6 unicast-routing

vrf definition v1
 rd 100:1
 !
address-family ipv6
 exit-address-family
!

vrf definition v2
 rd 200:1
 !
address-family ipv6
 exit-address-family
!

interface Vlan100
 vrf forwarding v1

ipv6 address 1000:1::3/64
 ospfv3 100 ipv6 area 0
!

interface Vlan200
 vrf forwarding v2
 ipv6 address 2000:1::3/64
 ospfv3 200 ipv6 area 0
!

interface GigabitEthernet 1/0/1
switchport access vlan 100
end

interface GigabitEthernet 1/0/2
switchport access vlan 200
end

interface GigabitEthernet 1/0/24
 switchport trunk encapsulation dot1q
 switchport mode trunk
 end

router ospfv3 100
```

```
 router-id 40.40.40.40
 !
 address-family ipv6 unicast vrf v1
  redistribute connected
  area 0 normal
 exit-address-family
!

router ospfv3 200
 router-id 50.50.50.50
 !
 address-family ipv6 unicast vrf v2
  redistribute connected

area 0 normal
 exit-address-family
!
```

VRF-Lite に関するその他の参考資料

関連資料

関連項目	マニュアル タイトル
この章で使用するコマンドの完全な構文および使用方法の詳細。	の「IP マルチキャスト ルーティングのコマンド」の項を参照してください。*Command Reference (Catalyst 9300 Series Switches)*

標準および RFC

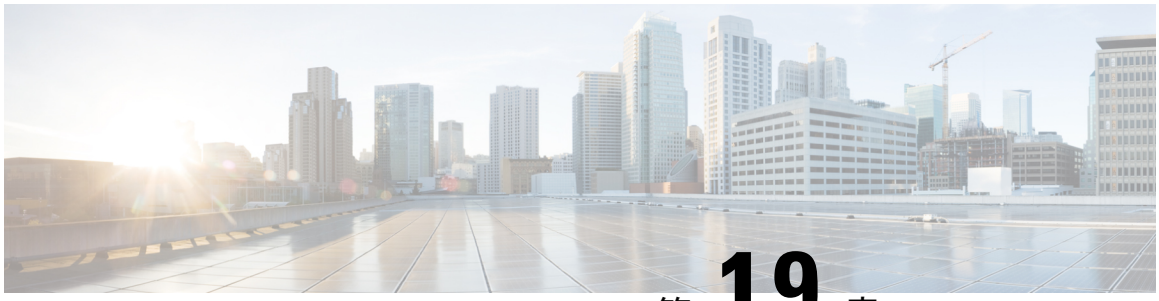
標準/RFC	タイトル
RFC 6763	『*DNS-Based Service Discovery*』
マルチキャスト DNS インターネット（ドラフト）	マルチキャスト

マルチキャスト VRF-Lite の機能履歴と情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。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。

プラットフォームのサポートおよびシスコ 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 にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

機能名	リリース	機能情報
VRF-Lite を使用した IPv6 マルチキャストのサポート	Cisco IOS XE Everest 16.6.1	IPv6 VRF-Lite によって、サービスプロバイダーは 1 つのインターフェイスを使用して、重複する IP アドレスを持つ複数の VPN をサポートできます。



第 19 章

Multi-VRF CE の設定

Multi-VRF CE に関する情報

バーチャル プライベート ネットワーク（VPN）は、ISP バックボーン ネットワーク上でお客様にセキュアな帯域幅共有を提供します。VPN は、共通ルーティング テーブルを共有するサイトの集合です。カスタマー サイトは、1 つまたは複数のインターフェイスでサービス プロバイダ ネットワークに接続され、サービス プロバイダは、VRF テーブルと呼ばれる VPN ルーティング テーブルと各インターフェイスを関連付けます。

スイッチが Network Advantageライセンスで稼働している場合、スイッチはカスタマーエッジ（CE）デバイスの Multiple VPN Routing/Forwarding（Multi-VRF）インスタンスをサポートします（Multi-VRF CE）。サービス プロバイダは、Multi-VRF CE により、重複する IP アドレスで複数の VPN をサポートできます。

（注） スイッチでは、VPN のサポートのためにマルチプロトコル ラベル スイッチング（MPLS）が使用されません。

Multi-VRF CE の概要

Multi-VRF CE は、サービス プロバイダが複数の VPN をサポートし、VPN 間で IP アドレスを重複して使用できるようにする機能です。Multi-VRF CE は入力インターフェイスを使用して、さまざまな VPN のルートを区別し、1 つまたは複数のレイヤ 3 インターフェイスと各 VRF を関連付けて仮想パケット転送テーブルを形成します。VRF 内のインターフェイスは、イーサ

ネット ポートのように物理的なもの、または VLAN SVI のように論理的なものにもできますが、複数の VRF に属すことはできません。

（注） Multi-VRF CE インターフェイスは、レイヤ 3 インターフェイスである必要があります。

Multi-VRF CE には、次のデバイスが含まれます。

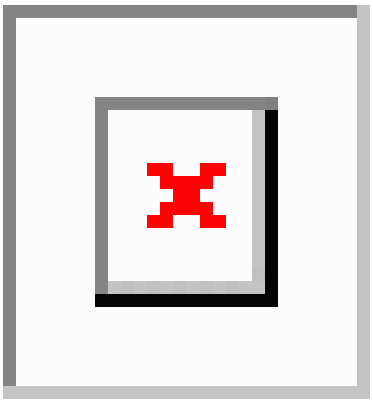
- お客様は、CEデバイスにより、1つまたは複数のプロバイダエッジ（PE）ルータへのデータ リンクを介してサービス プロバイダ ネットワークにアクセスできます。CE デバイスは、サイトのローカル ルートをルータにアドバタイズし、リモート VPN ルートをそこから学習します。スイッチを CE に設定することができます。
- PEルータは、スタティックルーティング、またはBGP、RIPv2、OSPF、EIGRPなどのルーティングプロトコルを使用して、CEデバイスとルーティング情報を交換します。PEは、直接接続している VPN に対する VPN ルートのみを保守する必要があります。そのため、すべてのサービス プロバイダ VPN ルートを PE が保守する必要はありません。各 PE ルータは、直接接続しているサイトごとに VRF を維持します。すべてのサイトが同じ VPN に存在する場合は、PE ルータの複数のインターフェイスを 1 つの VRF に関連付けることができます。各 VPN は、指定された VRF にマッピングされます。PE ルータは、ローカル VPN ルートを CE から学習したあとで、IBGP を使用して別の PE ルータと VPN ルーティング情報を交換します。
- CEデバイスに接続していないサービス プロバイダ ネットワークのルータは、プロバイダ ルータやコア ルータになります。

Multi-VRF CE では、複数のお客様が 1 つの CE を共有でき、CE と PE の間で 1 つの物理リンクだけが使用されます。共有 CE は、お客様ごとに別々の VRF テーブルを維持し、独自のルーティング テーブルに基づいて、お客様ごとにパケットをスイッチングまたはルーティングします。Multi-VRF CE は、制限付きの PE 機能を CE デバイスに拡張して、別々の VRF テーブルを維持し、VPN のプライバシーおよびセキュリティをブランチ オフィスに拡張します。

ネットワーク トポロジ

次の図に、スイッチを複数の仮想 CE として使用した構成例を示します。このシナリオは、中小企業など、VPNサービスの帯域幅要件の低いお客様に適しています。この場合、スイッチにはマルチ VRF CE のサポートが必要です。Multi-VRF CE はレイヤ 3 機能なので、VRF のそれぞれのインターフェイスはレイヤ 3 インターフェイスである必要があります。

図 *13* : 複数の仮想 *CE* として機能するスイッチ



CE スイッチは、レイヤ3インターフェイスを VRF に追加するコマンドを受信すると、Multi-VRF CE 関連のデータ構造で VLAN ID と Policy Label（PL）の間に適切なマッピングを設定し、VLAN ID と PL を VLAN データベースに追加します。

Multi-VRF CE を設定すると、レイヤ 3 フォワーディング テーブルは、次の 2 つのセクションに概念的に分割されます。

- Multi-VRF CE ルーティング セクションには、さまざまな VPN からのルートが含まれます。
- グローバル ルーティング セクションには、インターネットなど、VPN 以外のネットワークへのルートが含まれます。

さまざまな VRF の VLAN ID はさまざまな PL にマッピングされ、処理中に VRF を区別するために使用されます。レイヤ 3 設定機能では、学習した新しい VPN ルートごとに、入力ポートの VLAN ID を使用して PL を取得し、Multi-VRF CE ルーティング セクションに PL および新しいルートを挿入します。ルーテッド ポートからパケットを受信した場合は、ポート内部 VLAN ID 番号が使用されます。SVI からパケットを受信した場合は、VLAN 番号が使用されます。

パケット転送処理

Multi-VRF CE 対応ネットワークのパケット転送処理は次のとおりです。

- スイッチは、VPN からパケットを受信すると、入力 PL 番号に基づいてルーティング テーブルを検索します。ルートが見つかると、スイッチはパケットを PE に転送します。
- 入力 PE は、CE からパケットを受信すると、VRF 検索を実行します。ルートが見つかると、ルータは対応する MPLS ラベルをパケットに追加し、MPLS ネットワークに送信します。
- 出力 PE は、ネットワークからパケットを受信すると、ラベルを除去してそのラベルを使用し、正しい VPN ルーティング テーブルを識別します。次に、通常のルート検索を実行します。ルートが見つかると、パケットを正しい隣接デバイスに転送します。

- CE は、出力 PE からパケットを受信すると、入力 PL を使用して正しい VPN ルーティング テーブルを検索します。ルートが見つかると、パケットを VPN 内で転送します。

ネットワーク コンポーネント

VRF を設定するには、VRF テーブルを作成し、VRF に関連するレイヤ 3 インターフェイスを指定します。次に、VPN、および CE と PE 間でルーティング プロトコルを設定します。プロバイダのバックボーンで VPN ルーティング情報を配信する場合は、BGP が優先ルーティング プロトコルです。Multi-VRF CE ネットワークには、次の 3 つの主要コンポーネントがあります。

- VPN ルート ターゲット コミュニティ：VPN コミュニティのその他すべてのメンバのリスト。VPN コミュニティ メンバーごとに VPN ルート ターゲットを設定する必要があります。
- VPN コミュニティ PE ルータのマルチプロトコル BGP ピアリング：VPN コミュニティのすべてのメンバーに VRF 到達可能性情報を伝播します。VPN コミュニティのすべての PE ルータで BGP ピアリングを設定する必要があります。
- VPN 転送：VPN サービス プロバイダ ネットワークを介し、全 VPN コミュニティ メンバー間で、全トラフィックを伝送します。

VRF 認識サービス

IP サービスはグローバル インターフェイスに設定可能で、グローバル ルーティング インスタンスで稼働します。IP サービスは複数のルーティング インスタンス上で稼働するように拡張されます。これが、VRF 認識です。システム内の任意の設定済み VRF であればいずれも、VRF 認識サービス用に指定できます。

VRF 認識サービスは、プラットフォームに依存しないモジュールに実装されます。VRF とは、Cisco IOS 内の複数のルーティング インスタンスを意味します。各プラットフォームには、サポートする VRF 数に関して独自の制限があります。

VRF 認識サービスには、次の特性があります。

- ユーザーは、ユーザー指定の VRF 内のホストに ping を実行できます。
- ARP エントリは、個別の VRF で学習されます。ユーザーは、特定の VRF の ARP エントリを表示できます。

Multi-VRF CE の設定方法

Multi-VRF CE のデフォルト設定

表 *38 : VRF* のデフォルト設定

機能	デフォルト設定
VRF	ディセーブル。VRF は定義されていません。
マップ	インポートマップ、エクスポートマップ、ルートマップは定義されていません。
VRF 最大ルート数	ファスト イーサネット スイッチ：8000 ギガビット イーサネット スイッチ：12000
転送テーブル	インターフェイスのデフォルトは、グローバル ルーティング テーブルです。

Multi-VRF CE の設定時の注意事項

（注） Multi-VRF CE を使用するには、スイッチで Network Advantage ライセンスをイネーブルにする必要があります。

- Multi-VRF CE を含むスイッチは複数のお客様によって共有され、各お客様には独自のルーティング テーブルがあります。
- お客様は別々の VRF テーブルを使用するので、同じ IP アドレスを再利用できます。別々の VPN では IP アドレスの重複が許可されます。
- Multi-VRF CE では、複数のお客様が、PE と CE の間で同じ物理リンクを共有できます。複数の VLAN を持つトランク ポートでは、パケットがお客様間で分離されます。それぞれのお客様には独自の VLAN があります。
- Multi-VRF CE ではサポートされない MPLS-VRF 機能があります。ラベル交換、LDP 隣接関係、ラベル付きパケットはサポートされません。
- PE ルータの場合、Multi-VRF CE の使用と複数の CE の使用に違いはありません。図 41-6 では、複数の仮想レイヤ 3 インターフェイスが Multi-VRF CE デバイスに接続されています。
- スイッチでは、物理ポートか VLAN SVI、またはその両方の組み合わせを使用して、VRF を設定できます。SVI は、アクセス ポートまたはトランク ポートで接続できます。

- お客様は、別のお客様と重複しないかぎり、複数の VLAN を使用できます。お客様の VLAN は、スイッチに保存されている適切なルーティング テーブルの識別に使用される特定のルーティング テーブル ID にマッピングされます。
- スイッチは、1 つのグローバルネットワークおよび最大 256 の VRF をサポートします。
- CE と PE の間では、ほとんどのルーティング プロトコル（BGP、OSPF、RIP、およびスタティック ルーティング）を使用できます。ただし、次の理由から External BGP（EBGP）を使用することを推奨します。
  - BGP では、複数の CE とのやり取りに複数のアルゴリズムを必要としません。
  - BGP は、さまざまな管理者によって稼働するシステム間でルーティング情報を渡すように設計されています。
  - BGP では、ルートの属性を CE に簡単に渡すことができます。
- Multi-VRF CE は、パケットのスイッチング レートに影響しません。
- VPN マルチキャストはサポートされません。
- プライベート VLAN で VRF をイネーブルにできます（逆も同様です）。
- インターフェイスでポリシーベース ルーティング（PBR）がイネーブルになっている場合は、VRF をイネーブルにできません（逆も同様です）。
- インターフェイスで Web Cache Communication Protocol（WCCP）がイネーブルになっている場合は、VRF をイネーブルにできません（逆も同様です）。

VRF の設定

次の操作を行ってください。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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。

	コマンドまたはアクション	目的
ステップ 3	**ip routing** 例： `Device(config)#ip routing`	IP ルーティングを有効にします。
ステップ 4	**ip vrf** *vrf-name* 例： `Device(config)#ip vrf vpn1`	VRF 名を指定し、VRF コンフィギュレーション モードを開始します。
ステップ 5	**rd** *route-distinguisher* 例： `Device(config-vrf)#rd 100:2`	ルート識別子を指定して VRF テーブルを作成します。AS 番号と任意の番号（xxx:y）または IP アドレスと任意の番号（A.B.C.D:y）を入力します。
ステップ 6	**route-target** {**export** | **import** | **both**} *route-target-ext-community* 例： `Device(config-vrf)#route-target both 100:2`	指定された VRF のインポート、エクスポート、またはインポートおよびエクスポート ルート ターゲット コミュニティのリストを作成します。AS システム番号と任意の番号（xxx:y）または IP アドレスと任意の番号（A.B.C.D:y）を入力します。*route-target-ext-community* は、ステップ 4 で入力した *route-distinguisher* と同一にする必要があります。
ステップ 7	**import map** *route-map* 例： `Device(config-vrf)#import map importmap1`	（任意）VRF にルート マップを対応付けます。
ステップ 8	**interface** *interface-id* 例： `Device(config-vrf)#interface gigabitethernet 1/0/1`	VRF に関連付けるレイヤ 3 インターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。インターフェイスにはルーテッド ポートまたは SVI を設定できます。
ステップ 9	**ip vrf forwarding** *vrf-name* 例： `Device(config-if)#ip vrf forwarding vpn1`	VRF をレイヤ 3 インターフェイスに対応付けます。 （注） **ip vrf forwarding** が管理インターフェイスで有効になっている場合、アクセス ポイントは加入しません。
ステップ 10	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
ステップ 11	**show ip vrf** [**brief** | **detail** | **interfaces**] [*vrf-name*] 例： `Device#show ip vrf interfaces vpn1`	設定を確認します。設定した VRF に関する情報を表示します。
ステップ 12	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

Multi-VRF CE の設定方法

マルチキャスト VRF の設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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 routing** 例： `Device(config)#ip routing`	IP ルーティング モードをイネーブルにします
ステップ 4	**ip vrf** *vrf-name* 例： `Device(config)#ip vrf vpn1`	VRF 名を指定し、VRF 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 5	**rd** *route-distinguisher* 例： `Device(config-vrf)#rd 100:2`	ルート識別子を指定して VRF テーブルを作成します。AS 番号と任意の番号（xxx:y）または IP アドレスと任意の番号（A.B.C.D:y）を入力します。
ステップ 6	**route-target** {**export** | **import** | **both**} *route-target-ext-community* 例： `Device(config-vrf)#route-target import 100:2`	指定された VRF のインポート、エクスポート、またはインポートおよびエクスポート ルート ターゲット コミュニティのリストを作成します。AS システム番号と任意の番号（xxx:y）または IP アドレスと任意の番号（A.B.C.D:y）を入力します。*route-target-ext-community* は、ステップ 4 で入力した *route-distinguisher* と同一にする必要があります。
ステップ 7	**import map** *route-map* 例： `Device(config-vrf)#import map importmap1`	（任意）VRF にルート マップを対応付けます。
ステップ 8	**ip multicast-routing vrf** *vrf-name* **distributed** 例： `Device(config-vrf)#ip multicast-routing vrf vpn1 distributed`	（任意）VRF テーブルでグローバル マルチキャスト ルーティングをイネーブルにします。
ステップ 9	**interface** *interface-id* 例： `Device(config-vrf)#interface gigabitethernet 1/0/2`	VRF に関連付けるレイヤ 3 インターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。インターフェイスはルーテッド ポートまたは SVI に設定できます。
ステップ 10	**ip vrf forwarding** *vrf-name* 例： `Device(config-if)#ip vrf forwarding vpn1`	VRF をレイヤ 3 インターフェイスに対応付けます。
ステップ 11	**ip address** *ip-address* **mask** 例： `Device(config-if)#ip address 10.1.5.1 255.255.255.0`	レイヤ 3 インターフェイスの IP アドレスを設定します。
ステップ 12	**ip pim sparse-dense mode** 例： `Device(config-if)#ip pim sparse-dense mode`	VRF に関連付けられているレイヤ 3 インターフェイス上で、PIM をイネーブルにします。

	コマンドまたはアクション	目的
ステップ 13	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 14	**show ip vrf** [**brief** | **detail** | **interfaces**] [*vrf-name*] 例： `Device#show ip vrf detail vpn1`	設定を確認します。設定した VRF に関する情報を表示します。
ステップ 15	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

VPN ルーティング セッションの設定

VPN内のルーティングは、サポートされている任意のルーティングプロトコル（RIP、OSPF、EIGRP、BGP）、またはスタティックルーティングで設定できます。ここで説明する設定はOSPF のものですが、その他のプロトコルでも手順は同じです。

（注） VRF インスタンス内で EIGRP ルーティングプロセスが実行されるように設定するには、**autonomous-system** *autonomous-system-number* アドレス ファミリ コンフィギュレーション モード コマンドを入力して、自律システム番号を設定する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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。

	コマンドまたはアクション	目的
ステップ 3	**router ospf** *process-id* **vrf** *vrf-name* 例： `Device(config)#router ospf 1 vrf vpn1`	OSPF ルーティングをイネーブルにして VPN 転送テーブルを指定し、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	**log-adjacency-changes** 例： `Device(config-router)#log-adjacency-changes`	（任意）隣接ステートの変更を記録します。これは、デフォルトの状態です。
ステップ 5	**redistribute bgp** *autonomous-system-number* **subnets** 例： `Device(config-router)#redistribute bgp 10 subnets`	BGP ネットワークから OSPF ネットワークに情報を再配布するようにスイッチを設定します。
ステップ 6	**network** *network-number* **area** *area-id* 例： `Device(config-router)#network 1 area 2`	OSPF が動作するネットワーク アドレスとマスク、およびそのネットワーク アドレスのエリア ID を定義します。
ステップ 7	**end** 例： `Device(config-router)#end`	特権 EXEC モードに戻ります。
ステップ 8	**show ip ospf** *process-id* 例： `Device#show ip ospf 1`	OSPF ネットワークの設定を確認します。
ステップ 9	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

BGP PE/CE ルーティング セッションの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例：	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
	`Device#configure terminal`	
ステップ 2	**router bgp** *autonomous-system-number* 例： `Device(config)#router bgp 2`	その他の BGP ルータに AS 番号を渡す BGP ルーティング プロセスを設定し、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 3	**network** *network-number* **mask** *network-mask* 例： `Device(config-router)#network 5 mask 255.255.255.0`	BGP を使用してアナウンスするネットワークおよびマスクを指定します。
ステップ 4	**redistribute ospf** *process-id* **match internal** 例： `Device(config-router)#redistribute ospf 1 match internal`	OSPF 内部ルートを再配布するようにスイッチを設定します。
ステップ 5	**network** *network-number* **area** *area-id* 例： `Device(config-router)#network 5 area 2`	OSPF が動作するネットワーク アドレスとマスク、およびそのネットワーク アドレスのエリア ID を定義します。
ステップ 6	**address-family ipv4 vrf** *vrf-name* 例： `Device(config-router)#address-family ipv4 vrf vpn1`	PE/CE ルーティング セッションの BGP パラメータを定義し、VRF アドレス ファミリ モードを開始します。
ステップ 7	**neighbor** *address* **remote-as** *as-number* 例： `Device(config-router)#neighbor 10.1.1.2 remote-as 2`	PE と CE ルータの間の BGP セッションを定義します。
ステップ 8	**neighbor** *address* **activate** 例： `Device(config-router)#neighbor 10.2.1.1 activate`	IPv4 アドレス ファミリのアドバタイズメントをアクティブ化します。
ステップ 9	**end** 例： `Device(config-router)#end`	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
ステップ 10	**show ip bgp** [**ipv4**] [**neighbors**] 例： `Device#show ip bgp ipv4 neighbors`	BGP 設定を確認します。
ステップ 11	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

Multi-VRF CE のモニタリング

表 *39 : Multi-VRF CE* 情報を表示するコマンド

コマンド	目的
show ip protocols vrf *vrf-name*	VRF に対応付けられたルーティ 情報を表示します。
show ip route vrf *vrf-name* [**connected**] [*protocol* [*as-number*]] [**list**] [**mobile**] [**odr**] [**profile**] [**static**] [**summary**] [**supernets-only**]	VRF に対応付けられた IP ルーテ 情報を表示します。
show ip vrf [**brief** | **detail** | **interfaces**] [*vrf-name*]	定義された VRF インスタンスに 示します。

VRF 認識サービスの設定

次のサービスは、VRF 認識です。

- ARP
- ping
- 簡易ネットワーク管理プロトコル（SNMP）
- ユニキャスト RPF（uRPF）
- Syslog
- traceroute
- FTP および TFTP

ARP 用 VRF 認識サービスの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**show ip arp vrf** *vrf-name* 例： `Device#show ip arp vrf vpn1`	指定された VRF 内の ARP テーブルを表示します。

ping 用 VRF 認識サービスの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**ping vrf***vrf-name***ip-host** 例： `Device#ping vrf vpn1 ip-host`	指定された VRF 内の ARP テーブルを表示します。

SNMP 用 VRF 認識サービスの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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	**snmp-server trap authentication vrf** 例： `Device(config)#snmp-server trap authentication vrf`	VRF で、パケットに対して SNMP トラップをイネーブルにします。

	コマンドまたはアクション	目的
ステップ 4	**snmp-server engineID remote** *host* **vrf** *vpn-instance engine-id string* 例： `Device(config)#snmp-server engineID remote 172.16.20.3 vrf vpn1 80000009030000B064EFE100`	スイッチ上で、リモート SNMP エンジンの名前を設定します。
ステップ 5	**snmp-server host** *host* **vrf** *vpn-instance* **traps** *community* 例： `Device(config)#snmp-server host 172.16.20.3 vrf vpn1 traps comaccess`	SNMP トラップ動作の受信側、および SNMP トラップの送信に使用される VRF テーブルを指定します。
ステップ 6	**snmp-server host** *host* **vrf** *vpn-instance* **informs** *community* 例： `Device(config)#snmp-server host 172.16.20.3 vrf vpn1 informs comaccess`	SNMP 通知動作の受信先を指定し、SNMP 通知の送信に使用される VRF テーブルを指定します。
ステップ 7	**snmp-server user** *user group* **remote** *host* **vrf** *vpn-instance security model* 例： `Device(config)#snmp-server user abcd remote 172.16.20.3 vrf vpn1 priv v2c 3des secure3des`	SNMP アクセス用に、VRF 上にあるリモート ホストの SNMP グループにユーザーを追加します。
ステップ 8	**end** 例： `Device(config-if)#end`	特権 EXEC モードに戻ります。

NTP 用 VRF 認識サービスの設定

NTP 用の VRF 認識サービスの設定には、NTP サーバーと、NTP サーバーに接続された NTP クライアント インターフェイスの設定が含まれます。

始める前に

NTP クライアントとサーバーの間の接続を確認します。NTP サーバーに接続されているクライアント インターフェイスで有効な IP アドレスおよびサブネットを設定します。

NTP クライアントでの NTP 用 VRF 認識サービスの設定

NTP サーバーに接続されているクライアント インターフェイスで次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • 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： `Device#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	**interface** *interface-id* 例： `Device(config)#interface gigabitethernet 1/0/1`	VRF に関連付けるレイヤ 3 インターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**vrf forwarding** *vrf-name* 例： `Device(config-if)#vrf forwarding A`	VRF をレイヤ 3 インターフェイスに対応付けます。
ステップ 5	**ip address** *ip-address subnet-mask* 例： `Device(config-if)#ip address 1.1.1.1 255.255.255.0`	インターフェイスの IP アドレスを入力します。
ステップ 6	**no shutdown** 例： `Device(config-if)#no shutdown`	インターフェイスをイネーブルにします。
ステップ 7	**exit** 例： `Device(config-if)exit`	インターフェイス コンフィギュレーション モードを終了します。
ステップ 8	**ntp authentication-key** *number* **md5** *md5-number* 例： `Device(config)#ntp authentication-key 1 md5 cisco123`	認証キーを定義します。デバイスが時刻源と同期するのは、時刻源がこれらの認証キーのいずれかを持ち、 **ntp trusted-key number** コマンドによってキー番号が指定されている場合だけです。 （注） 認証キー番号と MD5 パスワードは、クライアントとサーバーの両方で同じである必要があります。

	コマンドまたはアクション	目的
ステップ 9	**ntp authenticate** 例： `Device(config)#ntp authenticate`	NTP 認証機能をイネーブルにします。NTP 認証はデフォルトでディセーブルになっています。
ステップ 10	**ntp trusted-key** *key-number* 例： `Device(config)#ntp trusted-key 1`	NTP クライアントで同期をとれるようにするために、NTP サーバーによってその NTP パケットで提供される必要がある 1 つ以上のキーを指定します。trusted key の範囲は 1 ～ 65535 です。このコマンドにより、NTP クライアントが、信頼されていない NTP サーバーと誤って同期する、ということが防止されます。
ステップ 11	**ntp server vrf** *vrf-name* 例： `Device(config)#ntp server vrf A 1.1.1.2 key 1`	指定された VRF で NTP サーバーを設定します。

NTP サーバーでの NTP 用 VRF 認識サービスの設定

NTP サーバーで次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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	**ntp authentication-key** *number* **md5** *passowrd* 例： `Device(config)#ntp authentication-key 1 md5 cisco123`	認証キーを定義します。デバイスが時刻源と同期するのは、時刻源がこれらの認証キーのいずれかを持ち、 **ntp trusted-key number** コマンドによってキー番号が指定されている場合だけです。 （注） 認証キー番号と MD5 パスワードは、クライアントとサーバーの両方で同じである必要があります。

	コマンドまたはアクション	目的
ステップ 4	**ntp authenticate** 例： `Device(config)#ntp authenticate`	NTP 認証機能をイネーブルにします。NTP 認証はデフォルトでディセーブルになっています。
ステップ 5	**ntp trusted-key** *key-number* 例： `Device(config)#ntp trusted-key 1`	NTP クライアントで同期をとれるようにするために、NTP サーバーによってその NTP パケットで提供される必要がある 1 つ以上のキーを指定します。trusted key の範囲は 1 ～ 65535 です。このコマンドにより、NTP クライアントが、信頼されていない NTP サーバーと誤って同期する、ということが防止されます。
ステップ 6	**interface** *interface-id* 例： `Device(config)#interface gigabitethernet 1/0/3`	VRF に関連付けるレイヤ 3 インターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 7	**vrf forwarding** *vrf-name* 例： `Device(config-if)#vrf forwarding A`	VRF をレイヤ 3 インターフェイスに対応付けます。
ステップ 8	**ip address** *ip-address subnet-mask* 例： `Device(config-if)#ip address 1.1.1.2 255.255.255.0`	インターフェイスの IP アドレスを入力します。
ステップ 9	**exit** 例： `Device(config-if)exit`	インターフェイス コンフィギュレーション モードを終了します。

uRPF 用 VRF 認識サービスの設定

uRPF は、VRF に割り当てられたインターフェイス上で設定でき、送信元検索が VRF テーブルで実行されます。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	**configure terminal** 例： `Device#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	**interface***interface-id* 例： `Device(config)#interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 4	**no switchport** 例： `Device(config-if)#no switchport`	レイヤ 2 コンフィギュレーション モードからインターフェイスを削除します（物理インターフェイスの場合）。
ステップ 5	**ip vrf forwarding** *vrf-name* 例： `Device(config-if)#ip vrf forwarding vpn2`	インターフェイス上で VRF を設定します。
ステップ 6	**ip address** *ip-address* 例： `Device(config-if)#ip address 10.1.5.1`	インターフェイスの IP アドレスを入力します。
ステップ 7	**ip verify unicast reverse-path** 例： `Device(config-if)#ip verify unicast reverse-path`	インターフェイス上で uRPF を有効にします。
ステップ 8	**end** 例： `Device(config-if)#end`	特権 EXEC モードに戻ります。

VRF 認識 RADIUS の設定

VRF 認識 RADIUS を設定するには、まず RADIUS サーバー上で AAA をイネーブルにする必要があります。『*Per VRF AAA Feature Guide*』で説明されているとおり、スイッチで **ip vrf forwarding** *vrf-name* サーバーグループ コンフィギュレーション コマンドと **ip radius source-interface** グローバル コンフィギュレーション コマンドがサポートされます。

syslog 用 VRF 認識サービスの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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	**logging on** 例： `Device(config)#logging on`	ストレージ ルータ イベント メッセージのロギングを、イネーブルまたは一時的にディセーブルにします。
ステップ 4	**logging host** *ip-address* **vrf** *vrf-name* 例： `Device(config)#logging host 10.10.1.0 vrf vpn1`	ロギング メッセージが送信される Syslog サーバーのホスト アドレスを指定します。
ステップ 5	**logging buffered** *logging buffered size* **debugging** 例： `Device(config)#logging buffered critical 6000 debugging`	メッセージを内部バッファにロギングします。
ステップ 6	**logging trap debugging** 例： `Device(config)#logging trap debugging`	Syslog サーバーに送信されるロギング メッセージを制限します。
ステップ 7	**logging facility** *facility* 例： `Device(config)#logging facility user`	ロギング ファシリティにシステム ロギング メッセージを送信します。
ステップ 8	**end** 例：	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
	`Device(config-if)#end`	

traceroute 用 VRF 認識サービスの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**traceroute vrf** *vrf-name ipaddress* 例： `Device(config)#traceroute vrf vpn2 10.10.1.1`	宛先アドレスを取得する VPN VRF の名前を指定します。

FTP および TFTP 用 VRF 認識サービスの設定

FTP および TFTP を VRF 認識とするには、いくつかの FTP/TFTP CLI を設定する必要があります。たとえば、インターフェイスに付加される VRF テーブルを使用する場合、E1/0 であれば、ip tftp source-interface E1/0 コマンドまたは ip ftp source-interface E1/0 コマンドを設定して、特定のルーティング テーブルを使用するように TFTP または FTP サーバーに通知する必要があります。この例では、VRF テーブルが宛先 IP アドレスを検索するのに使用されます。これらの変更には下位互換性があり、既存の動作には影響を及ぼしません。つまり、VRF がそのインターフェイスに設定されていない場合でも、送信元インターフェイス CLI を使用して、特定のインターフェイスにパケットを送信できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#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 ftp source-interface** *interface-type interface-number* 例：	FTP 接続の発信元 IP アドレスを指定します。

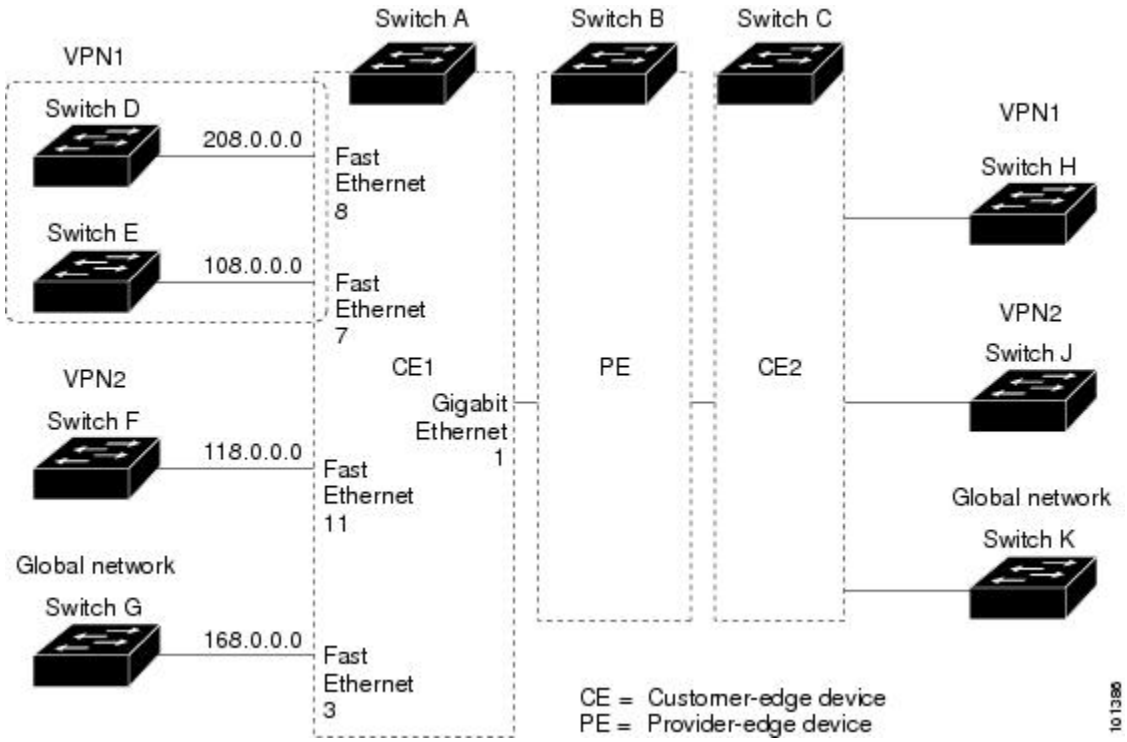
	コマンドまたはアクション	目的
	`Device(config)#ip ftp source-interface gigabitethernet 1/0/2`	
ステップ 4	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 5	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 6	**ip tftp source-interface** *interface-type interface-number* 例： `Device(config)#ip tftp source-interface gigabitethernet 1/0/2`	TFTP 接続用の送信元 IP アドレスを指定します。
ステップ 7	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。

Multi-VRF CE の設定例

Multi-VRF CE の設定例

VPN1、VPN2、およびグローバル ネットワークで使用されるプロトコルは OSPF です。CE/PE 接続には BGP が使用されます。図のあとに続く出力は、スイッチを CE スイッチ A として設定する例、およびカスタマー スイッチ D と F の VRF 設定を示しています。CE スイッチ C とその他のカスタマー スイッチを設定するコマンドは含まれていませんが、内容は同様です。

図 *14 : Multi-VRF CE* の設定例



スイッチ A では、ルーティングをイネーブルにして VRF を設定します。

```
Device#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)#ip routing
Device(config)#ip vrf v11
Device(config-vrf)#rd 800:1
Device(config-vrf)#route-target export 800:1
Device(config-vrf)#route-target import 800:1
Device(config-vrf)#exit
Device(config)#ip vrf v12
Device(config-vrf)#rd 800:2
Device(config-vrf)#route-target export 800:2
Device(config-vrf)#route-target import 800:2
Device(config-vrf)#exit
```

スイッチAのループバックおよび物理インターフェイスを設定します。ギガビットイーサネット ポート 1 は PE へのトランク接続です。ギガビット イーサネット ポート 8 と 11 は VPN に接続されます。

```
Device(config)#interface loopback1
Device(config-if)#ip vrf forwarding v11
Device(config-if)#ip address 8.8.1.8 255.255.255.0
Device(config-if)#exit

Device(config)#interface loopback2
Device(config-if)#ip vrf forwarding v12
Device(config-if)#ip address 8.8.2.8 255.255.255.0
Device(config-if)#exit
```

```
Device(config)#interface gigabitethernet1/0/5
Device(config-if)#switchport trunk encapsulation dot1q
Device(config-if)#switchport mode trunk
Device(config-if)#no ip address
Device(config-if)#exit
Device(config)#interface gigabitethernet1/0/8
Device(config-if)#switchport access vlan 208
Device(config-if)#no ip address
Device(config-if)#exit
Device(config)#interface gigabitethernet1/0/11
Device(config-if)#switchport trunk encapsulation dot1q
Device(config-if)#switchport mode trunk
Device(config-if)#no ip address
Device(config-if)#exit
```

スイッチ A で使用する VLAN を設定します。VLAN 10 は、CE と PE 間の VRF 11 によって使用されます。VLAN 20 は、CE と PE 間の VRF 12 によって使用されます。VLAN 118 と 208 は、それぞれスイッチ F とスイッチ D を含む VPN に使用されます。

```
Device(config)#interface vlan10
Device(config-if)#ip vrf forwarding v11
Device(config-if)#ip address 38.0.0.8 255.255.255.0
Device(config-if)#exit
Device(config)#interface vlan20
Device(config-if)#ip vrf forwarding v12
Device(config-if)#ip address 83.0.0.8 255.255.255.0
Device(config-if)#exit
Device(config)#interface vlan118
Device(config-if)#ip vrf forwarding v12
Device(config-if)#ip address 118.0.0.8 255.255.255.0
Device(config-if)#exit
Device(config)#interface vlan208
Device(config-if)#ip vrf forwarding v11
Device(config-if)#ip address 208.0.0.8 255.255.255.0
Device(config-if)#exit
```

VPN1 と VPN2 で OSPF ルーティングを設定します。

```
Device(config)#router ospf 1 vrf vl1
Device(config-router)#redistribute bgp 800 subnets
Device(config-router)#network 208.0.0.0 0.0.0.255 area 0
Device(config-router)#exit
Device(config)#router ospf 2 vrf vl2
Device(config-router)#redistribute bgp 800 subnets
Device(config-router)#network 118.0.0.0 0.0.0.255 area 0
Device(config-router)#exit
```

CE/PE ルーティングに BGP を設定します。

```
Device(config)#router bgp 800
Device(config-router)#address-family ipv4 vrf vl2
Device(config-router-af)#redistribute ospf 2 match internal
Device(config-router-af)#neighbor 83.0.0.3 remote-as 100
Device(config-router-af)#neighbor 83.0.0.3 activate
Device(config-router-af)#network 8.8.2.0 mask 255.255.255.0
Device(config-router-af)#exit
Device(config-router)#address-family ipv4 vrf vl1
Device(config-router-af)#redistribute ospf 1 match internal
Device(config-router-af)#neighbor 38.0.0.3 remote-as 100
Device(config-router-af)#neighbor 38.0.0.3 activate
```

```
Device(config-router-af)#network 8.8.1.0 mask 255.255.255.0
Device(config-router-af)#end
```

スイッチ D は VPN 1 に属します。次のコマンドを使用して、スイッチ A への接続を設定します。

```
Device#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)#ip routing
Device(config)#interface gigabitethernet1/0/2
Device(config-if)#no switchport
Device(config-if)#ip address 208.0.0.20 255.255.255.0
Device(config-if)#exit

Device(config)#router ospf 101
Device(config-router)#network 208.0.0.0 0.0.0.255 area 0
Device(config-router)#end
```

スイッチ F は VPN 2 に属します。次のコマンドを使用して、スイッチ A への接続を設定します。

```
Device#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)#ip routing
Device(config)#interface gigabitethernet1/0/1
Device(config-if)#switchport trunk encapsulation dot1q
Device(config-if)#switchport mode trunk
Device(config-if)#no ip address
Device(config-if)#exit

Device(config)#interface vlan118
Device(config-if)#ip address 118.0.0.11 255.255.255.0
Device(config-if)#exit

Device(config)#router ospf 101
Device(config-router)#network 118.0.0.0 0.0.0.255 area 0
Device(config-router)#end
```

このコマンドをスイッチ B（PE ルータ）で使用すると、CE デバイス、スイッチ A に対する接続だけが設定されます。

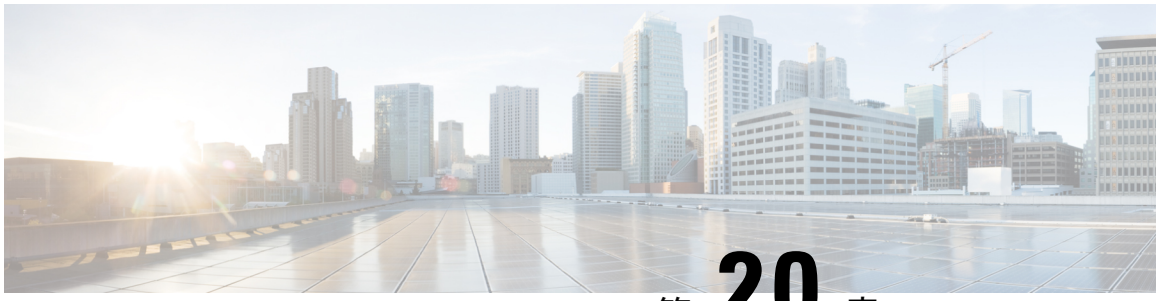
```
Device#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)#ip vrf v1
Device(config-vrf)#rd 100:1
Device(config-vrf)#route-target export 100:1
Device(config-vrf)#route-target import 100:1
Device(config-vrf)#exit

Device(config)#ip vrf v2
Device(config-vrf)#rd 100:2
Device(config-vrf)#route-target export 100:2
Device(config-vrf)#route-target import 100:2
Device(config-vrf)#exit
Device(config)#ip cef
Device(config)#interface Loopback1
Device(config-if)#ip vrf forwarding v1
Device(config-if)#ip address 3.3.1.3 255.255.255.0
Device(config-if)#exit

Device(config)#interface Loopback2
```

```
Device(config-if)#ip vrf forwarding v2
Device(config-if)#ip address 3.3.2.3 255.255.255.0
Device(config-if)#exit

Device(config)#interface gigabitethernet1/1/0.10
Device(config-if)#encapsulation dot1q 10
Device(config-if)#ip vrf forwarding v1
Device(config-if)#ip address 38.0.0.3 255.255.255.0
Device(config-if)#exit

Device(config)#interface gigabitethernet1/1/0.20
Device(config-if)#encapsulation dot1q 20
Device(config-if)#ip vrf forwarding v2
Device(config-if)#ip address 83.0.0.3 255.255.255.0
Device(config-if)#exit

Device(config)#router bgp 100
Device(config-router)#address-family ipv4 vrf v2
Device(config-router-af)#neighbor 83.0.0.8 remote-as 800
Device(config-router-af)#neighbor 83.0.0.8 activate
Device(config-router-af)#network 3.3.2.0 mask 255.255.255.0
Device(config-router-af)#exit
Device(config-router)#address-family ipv4 vrf vl
Device(config-router-af)#neighbor 38.0.0.8 remote-as 800
Device(config-router-af)#neighbor 38.0.0.8 activate
Device(config-router-af)#network 3.3.1.0 mask 255.255.255.0
Device(config-router-af)#end
```

マルチ VRF CE の機能情報

表 *40 :* マルチ *VRF CE* の機能情報

機能名	リリース	機能情報
マルチ VRF CE	Cisco IOS XE Everest 16.5.1a	この機能が導入されました



第 20 章

ユニキャスト リバース パス転送の設定

ユニキャスト リバース パス転送の設定

ユニキャスト リバース パス転送（ユニキャスト RPF）機能は、検証可能な送信元 IP アドレスが不足している IP パケットを廃棄することで、間違ったまたは偽造（スプーフィングされた）送信元 IP アドレスがネットワークに流れて発生する問題を軽減するのに役立ちます。たとえば、Smurf や Tribal Flood Network（TFN）など、多くの一般的なタイプの DoS 攻撃は、偽造された、または次々に変わる送信元 IP アドレスを使用して、攻撃を突き止めたりフィルタすることを攻撃者が阻止できるようにします。パブリック アクセスを提供するインターネット サービス プロバイダ（ISP）の場合、uRPF が IP ルーティング テーブルと整合性の取れた有効な送信元アドレスを持つパケットだけを転送することによって、そのような攻撃をそらします。この処理により、ISP のネットワーク、その顧客、および残りのインターネットが保護されます。

（注）
- ユニキャスト RPF は、Network Essentials でサポートされています。
- スイッチが複数のスイッチ タイプが混在する混合ハードウェア スタック内にある場合は、ユニキャスト RPF を設定しないでください。

IP uRPF 設定の詳細については、『*Cisco IOS Security Configuration Guide*』の「Other Security Features」の章を参照してください。

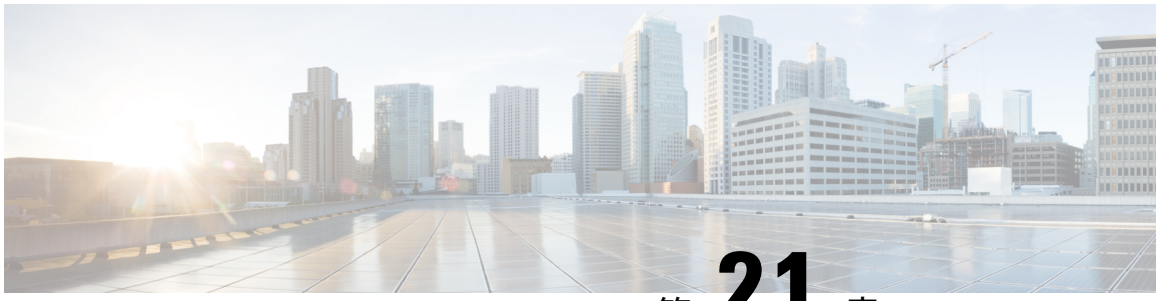
IPv6 ユニキャスト リバース パス転送の設定

ユニキャスト リバース パス転送（ユニキャスト RPF）機能は、検証できない送信元 IP アドレスの IP パケットを廃棄することで、間違ったまたは偽造（スプーフィングされた）送信元 IP アドレスがネットワークに流れて発生する問題を軽減するのに役立ちます。たとえば、Smurf や Tribal Flood Network（TFN）など、多くの一般的なタイプの DoS 攻撃は、偽造された、または次々に変わる送信元 IP アドレスを使用して、攻撃を突き止めたりフィルタすることを攻撃

者が阻止できるようにします。パブリック アクセスを提供するインターネット サービス プロバイダ（ISP）の場合、uRPF が IP ルーティング テーブルと整合性の取れた有効な送信元アドレスを持つパケットだけを転送することによって、そのような攻撃をそらします。この処理により、ISP のネットワーク、その顧客、および残りのインターネットが保護されます。

（注）
- スイッチが複数のスイッチタイプが混在する混合ハードウェア スタック内にある場合は、ユニキャスト RPF を設定しないでください。

IP ユニキャスト RPF 設定の詳細については、『*Cisco IOS Security Configuration Guide, Release 12.4*』の「*Other Security Features*」の章を参照してください。



第 21 章

Generic Routing Encapsulation（GRE）トンネルIP送信元および宛先VRFメンバーシップの設定

GRE トンネル IP 送信元および宛先 VRF メンバーシップの制約事項

- トンネルの両端は同じ VRF 内に存在する必要があります。
- tunnel vrf コマンドで関連付けられた VRF は、トンネルがパケットを送信する際に経由する物理インターフェイスに関連付けられている VRF と同じです（外部 IP パケット ルーティング）。
- ip vrf forwarding コマンドを使用してトンネルに関連付けられた VRF は、パケットがトンネルを出る際に転送される VRF です（内部 IP パケット ルーティング）。
- この機能では、マルチキャスト トンネルを通過するマルチキャスト パケットのフラグメンテーションはサポートされません。
- この機能では、ISIS（Intermediate System to Intermediate System）プロトコルはサポートされません。
- IPv6 ICMP 応答パケットは、IPv4 GRE トンネルではサポートされていません。

GRE トンネル IP 送信元および宛先 VRF メンバーシップについての情報

この機能では、トンネルの送信元と宛先を任意のバーチャルプライベートネットワーク（VPN）ルーティングおよび転送（VRF）テーブルに所属するように設定できます。VRF テーブルには、各 VPN のルーティング データが保管されます。VRF テーブルでは、ネットワーク アクセス サーバー（NAS）に接続されているカスタマー サイトの VPN メンバーシップを定義します。各 VRF テーブルは、IP ルーティング テーブル、派生したシスコ エクスプレス フォワーディング（CEF）テーブル、およびルーティング テーブルに含まれる情報を制御するガイドラインおよびルーティング プロトコル パラメータから構成されます。

以前は、GRE IP トンネルでは IP トンネルの宛先がグローバル ルーティング テーブルに含まれている必要がありました。この機能の実装により、トンネルの送信元と宛先が任意の VRF に所属するよう設定できます。既存の GRE トンネルと同様、トンネルの宛先へのルートが定義されていない場合は、トンネルはディセーブルになります。

GRE トンネル IP 送信元および宛先 VRF メンバーシップの設定方法

GRE トンネル IP 送信元および宛先 VRF メンバーシップを設定するには、次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface tunnel***number*
4. **ip vrf forwarding***vrf-name*
5. **ip address***ip-address subnet-mask*
6. **tunnel source** {*ip-address* | *type number*}
7. **tunnel destination** {*hostname* | *ip-address*}
8. **tunnel vrf***vrf-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： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	**configure terminal** 例： `Device#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	**interface tunnel***number* 例： `Device(config)#interface tunnel 0`	指定したインターフェイスのインターフェイス コンフィギュレーション モードを開始します。 • *number* はトンネルインターフェイスに関連付けられた番号です。
ステップ 4	**ip vrf forwarding***vrf-name* 例： `Device(config-if)#ip vrf forwarding green`	バーチャルプライベート ネットワーク（VPN）ルーティングおよび転送（VRF）インスタンスをインターフェイスまたはサブインターフェイスに関連付けます。 • *vrf-name* は VRF に割り当てられる名前です。
ステップ 5	**ip address***ip-address subnet-mask* 例： `Device(config-if)#ip address 10.7.7.7 255.255.255.255`	インターフェイス IP アドレスとサブネット マスクを指定します。 • *ip-address* でインターフェイスの IP アドレスを指定します。 • *subnet-mask* でインターフェイスのサブネットマスクを指定します。
ステップ 6	**tunnel source** {*ip-address* | *type number*} 例： `Device(config-if)#tunnel source loop 0`	トンネルインターフェイスの送信元を指定します。 • *ip-address* でトンネル内のパケットの送信元アドレスとして使用する IP アドレスを指定します。 • *type* でインターフェイスのタイプ（シリアルなど）を指定します。 • *number* でポート、コネクタ、またはインターフェイスカードの番号を指定します。この番号は、設置時、またはシステムへの追加時に、工場で割り当てられます。また、show interfaces コマンドを使用して表示できます。
ステップ 7	**tunnel destination** {*hostname* | *ip-address*} 例： `Device(config-if)#tunnel destination 10.5.5.5`	トンネルの宛先を指定します。 • *hostname* で宛先ホストの名前を指定します。 • *ip-address* で宛先ホストの IP アドレスを指定します。

	コマンドまたはアクション	目的
ステップ 8	**tunnel vrf***vrf-name* 例： `Device(config-if)#tunnel vrf finance1`	特定のトンネル宛先に VPN ルーティングおよび転送（VRF）インスタンスを関連付けます。 • *vrf-name* は VRF に割り当てられる名前です。

GRE トンネル IP 送信元および宛先 VRF メンバーシップの設定例

次に、VRF green を使用してインターフェイス e0 で受信されたパケットを、VRF blue を使用し、インターフェイス e1 を通じてトンネルから外部へ転送する例を示します。

```
ip vrf blue rd 1:1

ip vrf green rd 1:2

interface loop0
ip vrf forwarding blue
ip address 10.7.7.7 255.255.255.255

interface tunnel0
ip vrf forwarding green
ip address 10.3.3.3 255.255.255.0 tunnel source loop 0
tunnel destination 10.5.5.5 tunnel vrf blue

interface ethernet0
ip vrf forwarding green
ip address 10.1.1.1 255.255.255.0

interface ethernet1
ip vrf forwarding blue
ip address 10.2.2.2 255.255.255.0

ip route vrf blue 10.5.5.5 255.255.255.0 ethernet 1
```

その他の参考資料

表 *41* : 関連資料

関連項目	マニュアル タイトル
VRF テーブル	『Cisco IOS Switching Services Configuration Guide, Release 12.2』の「Configuring Multiprotocol Label Switching」の章
トンネル	『Cisco IOS Interface Configuration Guide, Release 12.2』

Generic Routing Encapsulation（GRE）トンネル IP 送信元および宛先 VRF メンバーシップの機能履歴

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。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 *42 : Generic Routing Encapsulation*（*GRE*）トンネル *IP* 送信元および宛先 *VRF* メンバーシップの機能履歴

機能名	リリース	機能情報
Generic Routing Encapsulation トンネル IP 送信元および宛先 VRF メンバーシップ	Cisco IOS 16.6.1	Generic Routing Encapsulation トンネルの IP 送信元および宛先の VRF メンバーシップ機能では、トンネルの送信元および宛先が任意のバーチャル プライベート ネットワーク（VPN）ルーティングおよび転送（VRF）テーブルに属するように設定できます。

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。