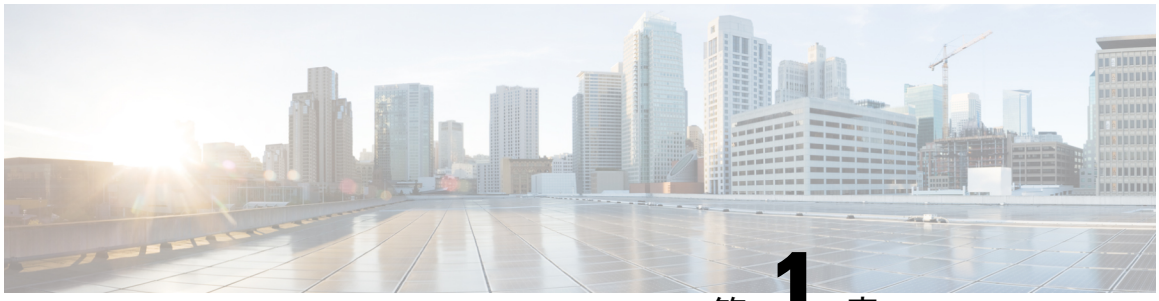
Cisco IOS XE Fuji 16.9.x（Catalyst 9300 スイッチ）ルーティング コンフィギュレーション ガイド

初版：2018 年 7 月 18 日

最終更新：2019 年 4 月 4 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/

目 次



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。

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。最新の機能情報および警告については、使用するプラットフォームおよびソフトウェア リリースの Bug Search Tool およびリリース ノートを参照してください。このモジュールに記載されている機能の詳細を検索し、各機能がサポートされているリリースのリストを確認する場合は、このモジュールの最後にある機能情報の表を参照してください。

プラットフォームのサポートおよびシスコ 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 には、http://www.cisco.com/go/cfn からアクセスします。Cisco.com のアカウントは必要ありません。

双方向フォワーディング検出の前提条件

• シスコ エクスプレス フォワーディングおよび IP ルーティングが、関連するすべてのスイッチでイネーブルになっていること。

- BFD を導入する前に、BFD でサポートされる IP ルーティング プロトコルのいずれかをスイッチで設定しておくこと。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。マルチホップのコンフィギュレーションはサポートされません。
- 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 のサポートについて確認し、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。
- BFD パケットは自己生成パケットの QoS ポリシーでは一致しません。
- BFE パケットは、**class class-default** コマンドで一致します。そのため、ユーザーは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

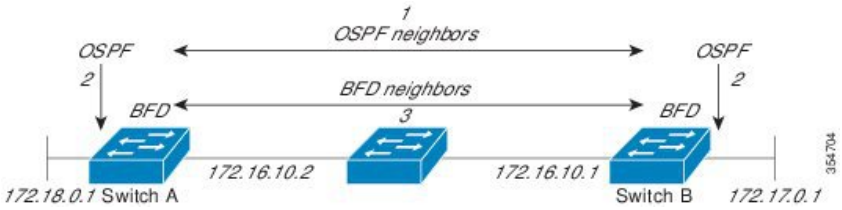
BFD は、インターフェイス、データリンク、および転送プレーンを含めて、2 つの隣接ルータ間の転送パスで、オーバーヘッドの少ない短期間の障害検出方法を提供します。

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 レベルでイネーブル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。このモードは、2台のシステム間でBFD制御パケットを送信することでルータ間のBFDネイバー セッションをアクティブ化して維持します。したがって、BFDセッションを作成するには、両方のシステムで（またはBFD ピアで）BFD を設定する必要があります。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 レベルおよびルータ レベルでBFDがイネーブルになっている場合、BFD セッションが作成されて BFD タイマーがネゴシエートされ、ネゴシエートされた間隔で BFD ピアが互いに BFD 制御パケットの送信を開始します。

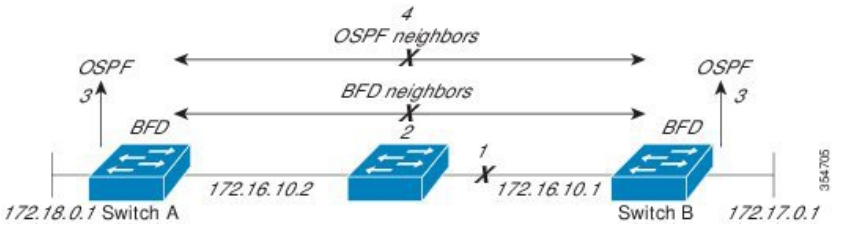
ネイバー関係

BFD はあらゆるメディア 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、およびOSPFの個別の高速BFDピア障害検出時間を提供します。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すると、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、OSPF ネイバー ルータ（2）で BFD ネイバー セッションを開始する要求が、ローカル BFD プロセスに送信されます。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）。代替パスを使用できる場合、ルータはただちにコンバージェンスを開始します。



ルーティング プロトコルでは、取得したネイバーそれぞれについて、BFD で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が、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、タイマーの取り消しが完了すると、BFD ピアは IGP hello プロトコルと同様に動作する（ただし、より高速な）、BFD制御パケットを送信して状態を検出します。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、障害が発生したピアをバイパスするには、ルーティング プロトコルがアクションを実行する必要があります。

- Cisco IOS XE Denali 16.3.1 では、シスコ デバイスは BFD バージョン 0 をサポートします。このバージョンでは、デバイスは実装時に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、BFD で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イネーブルになった 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 から、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 から、BFD 機能は、ルーティングされた 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）がイネーブル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、それによって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。ラインカードおよびフォワーディング プロセッサの機能はスイッチオーバーによって維持され、アクティブな RP の転送情報ベース（FIB）が NSF 動作で最新状態が維持されます。

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、それらの間で情報が同期されます。アクティブな RP に障害が発生したとき、ネットワーキング デバイスから削除されたとき、または手動でメンテナンスから排除されたときに、アクティブなプロセッサとスタンバイ プロセッサから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、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が保護されます。

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完全に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と見なされ、したがって、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 セッションが正常に確立されるように、ピア上のインターフェイスで BFD を設定し、ピア上の BFD クライアントに BFD ネイバーのアドレスを登録する必要があります。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、後者の要件は通常、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。

EIGRP、IS-IS、および OSPF の通常の導入で BFD に最も近い代替策は、EIGRP、IS-IS、および OSPF ルーティング プロトコルの変更された障害検出メカニズムを使用することです。

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 に fast hello を使用する場合、これらの Interior Gateway Protocol（IGP）プロトコル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、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。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。

手順の概要

1. **enable**
2. **configure terminal**
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*
4. **bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier*
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`	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。

	コマンドまたはアクション	目的
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 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

eBGP に対する BFD サポートの設定

ここでは、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するように、BGP 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。

始める前に

e BGP 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。

（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。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *as-tag*
4. **neighbor** *ip-address* **fall-over bfd**
5. **end**
6. **show bfd neighbors** [**details**]
7. **show ip bgp neighbor**

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** *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。
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 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、EIGRP 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。EIGRP 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、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。

手順の概要

1. **enable**
2. **configure terminal**
3. **router eigrp** *as-number*
4. 次のいずれかを実行します。
   - **bfd all-interfaces**
   - **bfd interface** *type number*
5. **end**
6. **show bfd neighbors** [**details**]
7. **show ip eigrp interfaces** [*type number*] [*as-number*] [**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：	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 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 をグローバルにイネーブルにします。
		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 をイネーブルにします。

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 サポートがイネーブル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。

（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）。

	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 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。 （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。

	コマンドまたはアクション	目的
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）。
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：	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。

	コマンドまたはアクション	目的
	`Device(config-if)# ip router isis tag1`	
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。

手順の概要

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **bfd all-interfaces**
5. **exit**
6. **interface** *type number*
7. **ip ospf bfd** [**disable**]
8. **end**
9. **show bfd neighbors** [**details**]
10. **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	**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。

	コマンドまたはアクション	目的
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 *OSPF* に対する *BFD* 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。

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） 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。

	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 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。

手順の概要

1. **enable**
2. **configure terminal**
3. **ip cef** [**distributed**]
4. **interface** *type number*
5. **ip address** *ip-address mask*
6. **standby** [*group-number*] **ip** [*ip-address* [**secondary**]]
7. **standby bfd**
8. **exit**
9. **standby bfd all-interfaces**
10. **exit**
11. **show standby 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。
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。

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

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。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface** *type number*
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*
5. **bfd interval** *milliseconds* **mix_rx** *milliseconds* **multiplier** *interval-multiplier*
6. **exit**
7. **ip route static bfd** *interface-type interface-number ip-address* [**group** *group-name* [**passive**]]
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*]
9. **exit**
10. **show ip static route**
11. **show ip static route bfd**
12. **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	**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 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： `Device(config-if)# bfd interval 500 min_rx 500 multiplier 5`	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

	コマンドまたはアクション	目的
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。
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、関連するすべてのルータで実行する必要があります。

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 エコーモードをディセーブルにする方法を示します。ルータからはエコー パケットが送信されず、ルータはネイバー ルータから受信する BFD エコー パケットを転送しません。

各 BFD ルータに対してこの手順を繰り返します。

手順の概要

1. **enable**
2. **configure terminal**
3. **no bfd echo**
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： `Router> 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： `Router# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**no bfd echo** 例： `Router(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： `Router(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） bfd-template 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。

手順の概要

1. **enable**
2. **configure terminal**
3. **bfd-template single-hop** *template-name*
4. **interval min-tx** *milliseconds* **min-rx** *milliseconds* **multiplier** *multiplier-value*
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`	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。必要に応じてこれらのタスクのコマンドを、正しい順序で入力します。

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

Catalyst 7600 シリーズ ルータのモニタリングと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**]

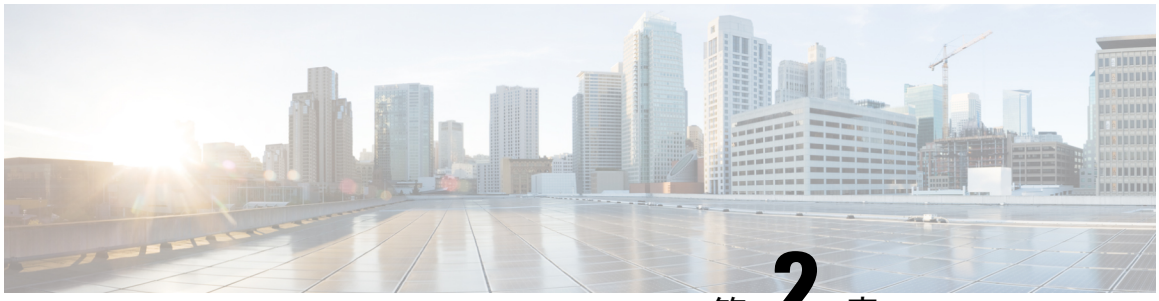
手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Router> 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： `Router# 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： `Router# 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

機能名	リリース	機能情報
双方向フォワーディング検出	Cisco IOS XE Everest 16.5.1a	この機能が導入されました



第 2 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 （32 ページ）を参照してください。

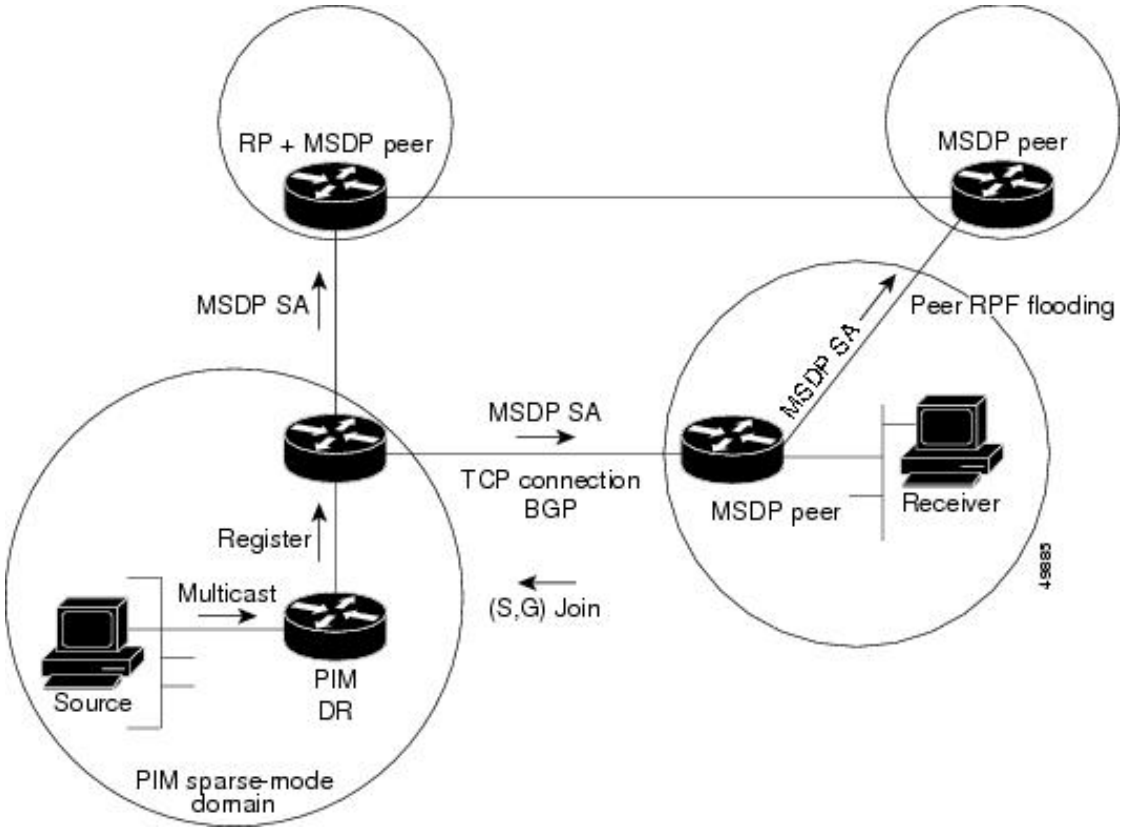
MSDP ピアは、非 RPF ピアから発信元 RP へ向かう同じ SA メッセージを受信すると、そのメッセージをドロップします。それ以外の場合、すべての MSDP ピアにメッセージが転送されます。

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： `デバイス> 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： `デバイス# 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
	`Router(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： `Router(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： `Router(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： `デバイス(config)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show running-config** 例： `デバイス# show running-config`	入力を確認します。
ステップ 8	**copy running-config startup-config** 例： `デバイス# 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： `デバイス> 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
	デバイス# **configure terminal**	
ステップ 3	**ip msdp cache-sa-state** [**list** *access-list-number*] 例： デバイス(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： デバイス(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： `デバイス(config)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show running-config** 例： `デバイス# show running-config`	入力を確認します。
ステップ 7	**copy running-config startup-config** 例： `デバイス# 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。

新しいメンバー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： `デバイス> 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： `デバイス# 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： `デバイス(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： `デバイス(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `デバイス# show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `デバイス# 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 （38 ページ）およびSA 要求メッセージのフィルタリング（40 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： `デバイス> 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： `デバイス# 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： `デバイス(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：
	`デバイス(config)# access list 21 permit 194.1.22.0`
	
	または
	`デバイス(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
	デバイス(config)# **end**	
ステップ 6	**show running-config** 例： デバイス# **show running-config**	入力を確認します。
ステップ 7	**copy running-config startup-config** 例： デバイス# **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： デバイス> **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： デバイス# **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： `デバイス(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： `デバイス(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： `デバイス(config)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show running-config** 例： `デバイス# 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
	デバイス# **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： デバイス> **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： デバイス# **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：
	`デバイス(config)# ip msdp sa-filter out switch.cisco.com`
	または
	`デバイス(config)# ip msdp sa-filter out list 100`
	または
	`デバイス(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： `デバイス(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： デバイス(config)# **end**	特権 EXEC モードに戻ります。
ステップ 6	**show running-config** 例： デバイス# **show running-config**	入力を確認します。
ステップ 7	**copy running-config startup-config** 例： デバイス# **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： デバイス> **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： デバイス# **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： デバイス(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： デバイス(config)# **end**	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： デバイス# **show running-config**	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： デバイス# **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： `デバイス> 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： `デバイス# 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：
	`デバイス(config)# ip msdp sa-filter in switch.cisco.com`
	または
	`デバイス(config)# ip msdp sa-filter in list 100`
	または
	`デバイス(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： `デバイス(config)# access list 100 permit ip`	（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。

	コマンドまたはアクション	目的
	194.1.22.0 1.1.1.1 194.3.44.0 1.1.1.1	• **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。
ステップ 5	**end** **例**： デバイス(config)# **end**	特権 EXEC モードに戻ります。
ステップ 6	**show running-config** **例**： デバイス# **show running-config**	入力を確認します。
ステップ 7	**copy running-config startup-config** **例**： デバイス# **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： `デバイス> 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： `デバイス# 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： `デバイス(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： `デバイス(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 5	**show running-config** 例： デバイス# **show running-config**	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： デバイス# **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： デバイス> **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： デバイス# **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： デバイス(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：	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
	デバイス(config)# **end**	
ステップ 5	**show running-config** 例： デバイス# **show running-config**	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： デバイス# **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： デバイス> **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： `デバイス# 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 border sa-address** *interface-id* 例： `デバイス(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： `デバイス(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 （38 ページ）を参照してください。
ステップ 5	**end** 例： `デバイス(config)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show running-config** 例： `デバイス# show running-config`	入力を確認します。
ステップ 7	**copy running-config startup-config** 例： `デバイス# 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： `デバイス> 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： `デバイス# 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： `デバイス(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： `デバイス(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 5	**show running-config** 例： デバイス# **show running-config**	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： デバイス# **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 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 *2 : 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 *3: 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。
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

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

ルータ C

```
Router(config)# ip msdp default-peer 10.1.1.1 prefix-list site-a
Router(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。

```
デバイス(config)# ip msdp cache-sa-state 100
デバイス(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。

```
デバイス(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。

```
デバイス(config)# ip msdp filter sa-request 171.69.2.2 list 1
デバイス(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。

```
デバイス(config)# ip msdp peer switch.cisco.com connect-source gigabitethernet1/0/1
デバイス(config)# ip msdp sa-filter out switch.cisco.com list 100
デバイス(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

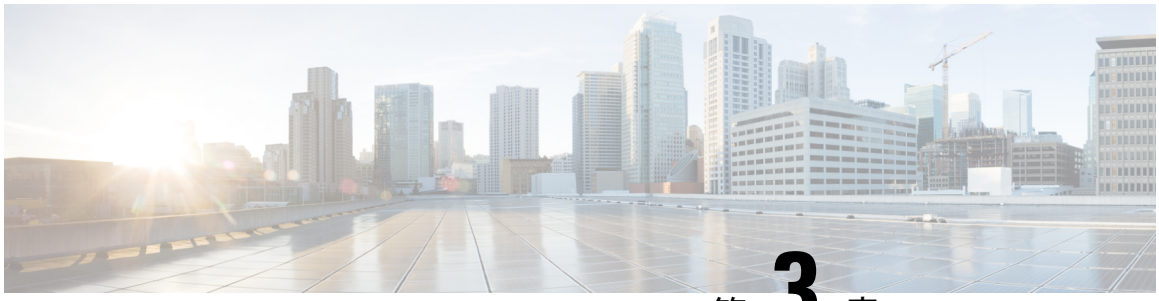
次に、*switch.cisco.com* という名前のピアからのすべての SA メッセージをフィルタリングする例を示します。

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

Multicast Source Discovery Protocol の機能情報

表 *4 : Multicast Source Discovery Protocol* の機能情報

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



第 3 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。

スイッチ スタックは、ネットワーク内のそれ以外のルータに対して、単一のルータとして動作し、認識されます。スタティック ルーティング、Routing Information Protocol（RIP）などの基本的なルーティング機能は、 Network Essentials ライセンスと Network Advantage ライセンスの両方で使用できます。拡張ルーティング機能およびその他のルーティングプロトコルを使用するには、スタンドアロンスイッチやアクティブスイッチで Network Advantage ライセンスを有効にする必要があります。

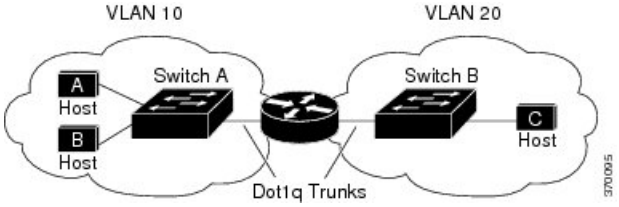
（注） IPv4 トラフィックに加えて、スイッチまたはスイッチ スタックが Network Essentials または Network Advantage ライセンスを実行している場合、IP バージョン 6（IPv6）ユニキャスト ルーティングをイネーブルにし、IPv6 トラフィックを転送するようにインターフェイスを設定できます。

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
- ルーティング プロトコルによるルートの動的な計算

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。

- ルーティング プロトコルを初期化し、設定します。
- ルーティング プロトコル メッセージおよびアップデートを他のルータに送信します。
- ピア ルータから受信したルーティング プロトコル メッセージおよびアップデートを処理します。
- 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。

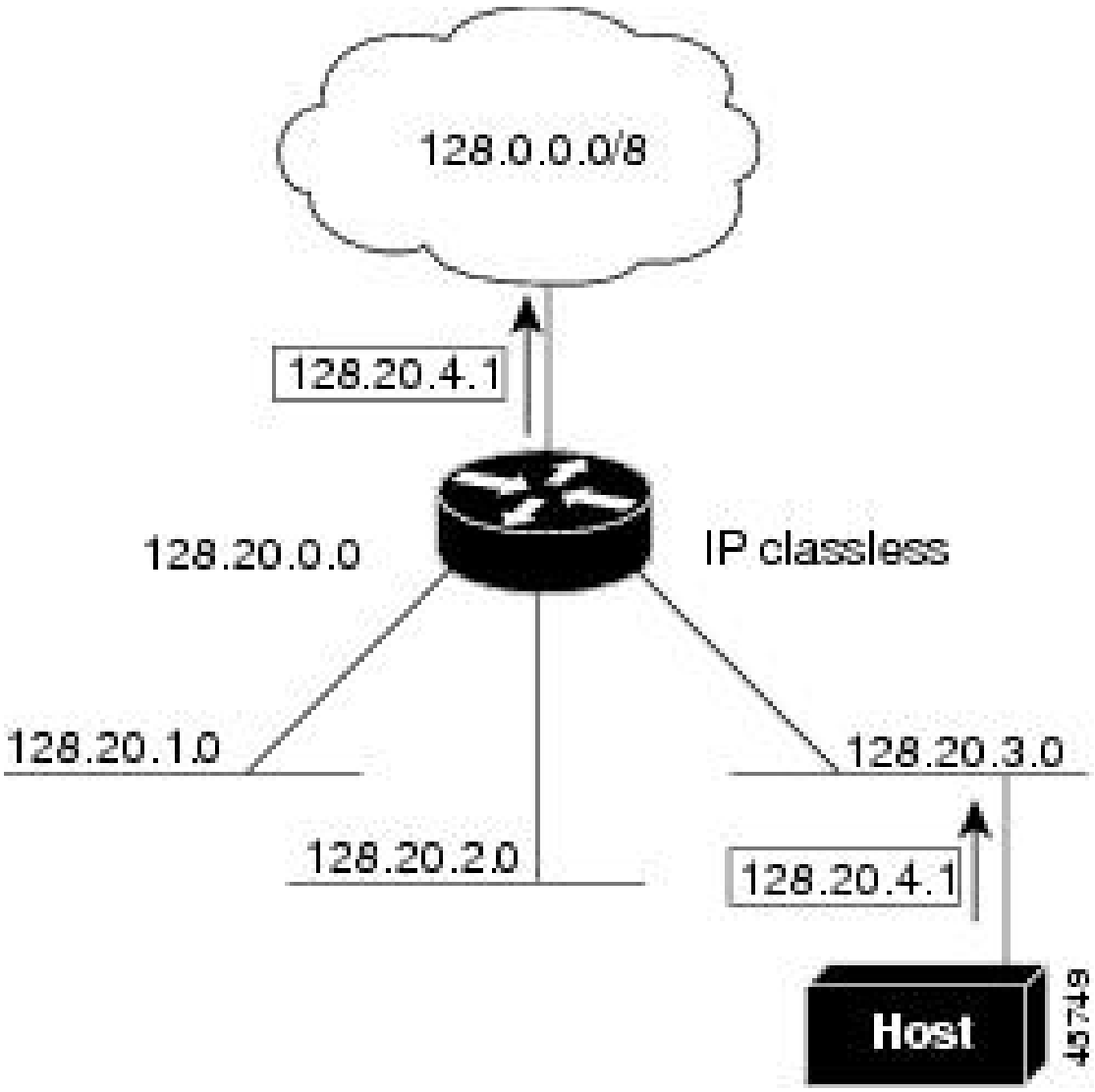
スイッチがリロードされると、NSF/SSO機能である場合でも、そのスイッチのポートがすべてダウンし、ルーティングに関わるインターフェイスにトラフィックの損失が発生します。

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。

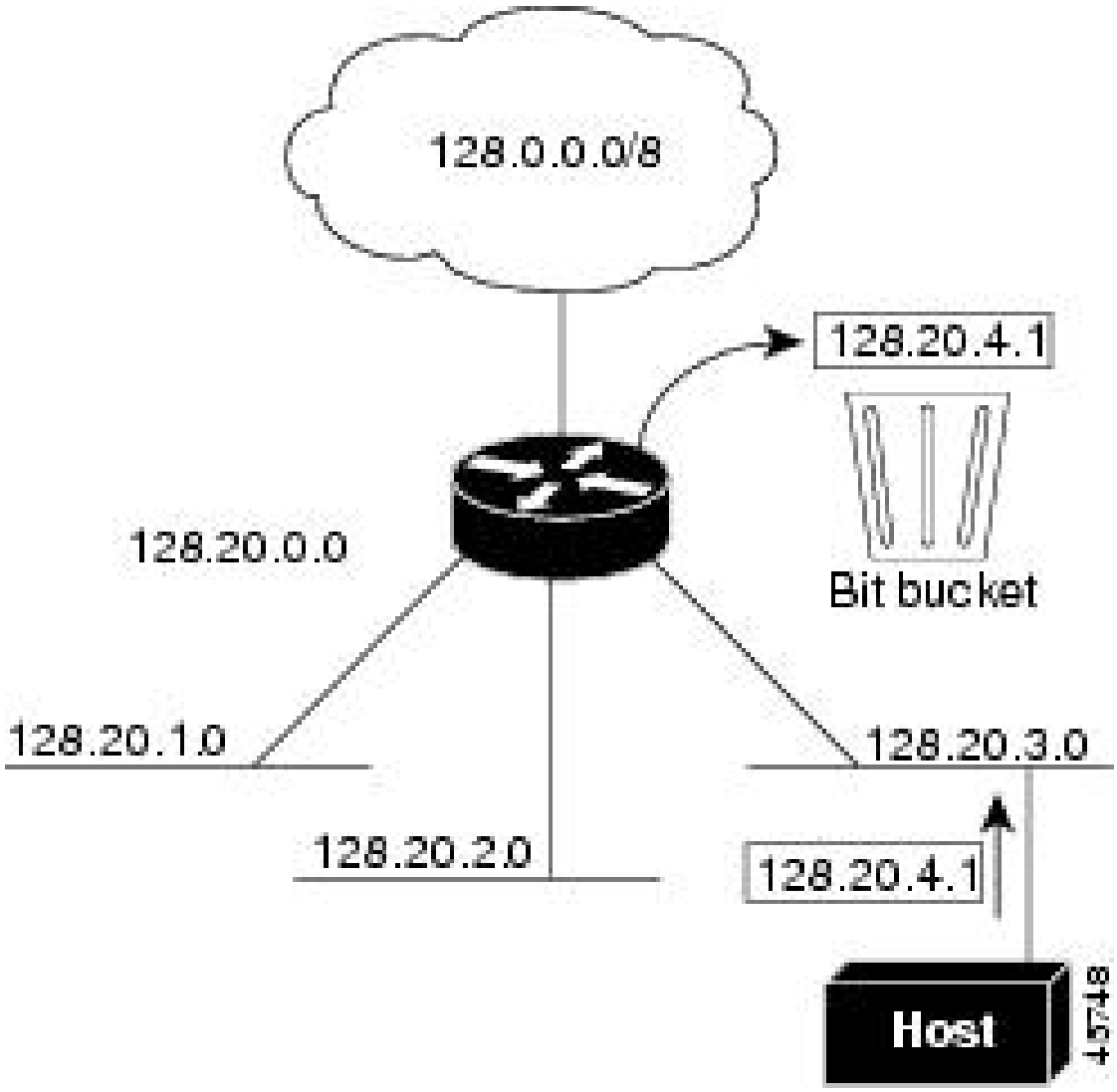
図では、クラスレスルーティングがイネーブルとなっています。ホストがパケットを128.20.4.1に送信すると、ルータはパケットを廃棄せずに、最適なスーパーネットルートに転送します。クラスレス ルーティングがディセーブルの場合、デフォルト ルートがないネットワークのサブネット宛てパケットを受信したルータは、パケットを廃棄します。

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。

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、ルーティングを有効にしたり、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。

デバイスでは、パケットの大部分がハードウェアで転送され、デバイスの 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 ルーティングの設定方法

デバイス上で、IPルーティングはデフォルトで無効となっているため、ルーティングを行う前に、IP ルーティングを有効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） レイヤ 3 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には、デバイスまたはスイッチ 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。

機能	デフォルト設定
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
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
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： デバイス> **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： デバイス# **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： デバイス(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： デバイス(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： デバイス(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： デバイス(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： デバイス(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 8	**show ip route** 例： デバイス# show ip route	入力を確認します。
ステップ 9	**show ip interface** [*interface-id*] 例： デバイス# show ip interface gigabitethernet 1/0/1	入力を確認します。
ステップ 10	**show running-config** 例： デバイス# **show running-config**	入力を確認します。
ステップ 11	**copy running-config startup-config** 例： デバイス# **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： デバイス> **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： `デバイス# 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： `デバイス(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。
ステップ 4	**end** 例： `デバイス(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `デバイス# show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `デバイス# 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： `デバイス> 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
	デバイス# **configure terminal**	
ステップ 3	**no ip classless** 例： デバイス(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： デバイス(config)# **end**	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： デバイス# **show running-config**	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： デバイス# **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： `デバイス> 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： `デバイス# 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	**arp** *ip-address hardware-address type* 例： `デバイス(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： `デバイス(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： `デバイス(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： `デバイス(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： `デバイス(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 8	**show interfaces** [*interface-id*] 例： デバイス# 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： デバイス# 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： デバイス# show ip arp	ARP キャッシュの内容を表示します。
ステップ 11	**copy running-config startup-config** 例： デバイス# **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。

カプセル化タイプを無効にするには、**no arp arpa** 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。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： デバイス> **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： デバイス# **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	**interface** *interface-id* 例： `デバイス(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： `デバイス(config-if)# arp arpa`	ARP カプセル化方法を指定します。
ステップ 5	**end** 例： `デバイス(config)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show interfaces** [*interface-id*] 例： `デバイス# show interfaces`	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。
ステップ 7	**copy running-config startup-config** 例： `デバイス# 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： `デバイス> 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
	デバイス# **configure terminal**	
ステップ 3	**interface** *interface-id* 例： デバイス(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： デバイス(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： デバイス(config)# **end**	特権 EXEC モードに戻ります。
ステップ 6	**show ip interface** [*interface-id*] 例： デバイス# 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： デバイス# **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： `デバイス> 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： `デバイス# 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： `デバイス(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： `デバイス(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show ip redirects** 例： `デバイス# 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。

	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： デバイス# **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： デバイス> **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： デバイス# **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： デバイス(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： デバイス(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。

	コマンドまたはアクション	目的
ステップ 5	**ip irdp multicast** 例： `デバイス(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： `デバイス(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： `デバイス(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。
ステップ 8	**ip irdp minadvertinterval** *seconds* 例： `デバイス(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： `デバイス(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： `デバイス(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：	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
	デバイス(config)# **end**	
ステップ 12	**show ip irdp** 例： デバイス# 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： デバイス# **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」のセクションの「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： `デバイス> 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： `デバイス# 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： `デバイス(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： `デバイス(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： `デバイス(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： `デバイス(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：	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
	デバイス(config)# **end**	
ステップ 8	**show ip interface** [*interface-id*] 例： デバイス# 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： デバイス# **show running-config**	入力を確認します。
ステップ 10	**copy running-config startup-config** 例： デバイス# **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： デバイス> **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： デバイス# **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	**interface** *interface-id* 例： `デバイス(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： `デバイス(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： `デバイス(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： `デバイス(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： `デバイス(config)# end`	特権 EXEC モードに戻ります。
ステップ 8	**show ip interface** [*interface-id*] 例： `デバイス# 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： `デバイス# show running-config`	入力を確認します。
ステップ 10	**copy running-config startup-config** 例： `デバイス# 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デバイス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： `デバイス> 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： `デバイス# 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： `デバイス(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： `デバイス(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： `デバイス(config)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show ip interface** [*interface-id*] 例： `デバイス# 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：	（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
	デバイス# **copy running-config startup-config**	

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： デバイス> **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： デバイス# **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： デバイス(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	**end** 例： デバイス(config)# **end**	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： デバイス# **show running-config**	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： デバイス# **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。

	コマンドまたはアクション	目的
ステップ 7	**configure terminal** 例： `デバイス# 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 8	**ip forward-protocol turbo-flood** 例： `デバイス(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 9	**end** 例： `デバイス(config)# end`	特権 EXEC モードに戻ります。
ステップ 10	**show running-config** 例： `デバイス# show running-config`	入力を確認します。
ステップ 11	**copy running-config startup-config** 例： `デバイス# 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

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 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 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

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 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。
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（エ
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各 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を

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：	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
	デバイス> **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： デバイス# **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： デバイス(config)# ip routing	IP ルーティングをイネーブルにします。
ステップ 4	**end** 例： デバイス(config)# **end**	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： デバイス# **show running-config**	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： デバイス# **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 ルーティングのイネーブル化の例

次に、ルーティングプロトコルとして RIP を使用し、 IP ルーティングを有効にする例を示します。

```
デバイス# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
デバイス(config)# ip routing

デバイス(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。

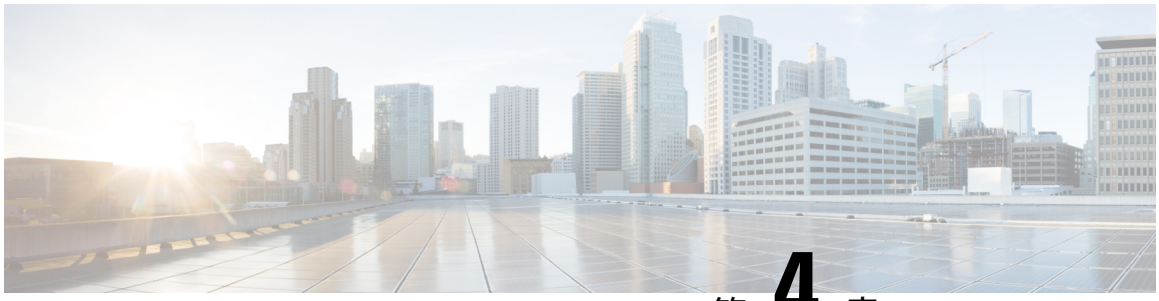
表 *8 : IP* ルートの削除またはルート ステータスの表示を行うコマンド

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

機能名	リリース	機能情報
IP ユニキャスト ルーティング	Cisco IOS XE Everest 16.5.1a	この機能が導入されました



第 4 章

RIP の設定

RIP に関する情報

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。

デバイスは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によって学習された場合、またはルータにラストリゾートゲートウェイがあり、RIP がデフォルトのメトリックによって設定されている場合、デバイス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。

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 *10 : 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
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

機能	デフォルト設定
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。デバイスでは、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： デバイス> **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： デバイス# **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： デバイス(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）。
ステップ 4	**router rip** 例： デバイス(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。

	コマンドまたはアクション	目的
ステップ 5	**network** *network number* 例： `デバイス(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： `デバイス(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： `デバイス(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： `デバイス(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： `デバイス(config-router)# version 2`	（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

	コマンドまたはアクション	目的
		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： `デバイス(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： `デバイス(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： `デバイス(config-router)# end`	特権 EXEC モードに戻ります。
ステップ 13	**show ip protocols** 例： `デバイス# show ip protocols`	入力を確認します。
ステップ 14	**copy running-config startup-config** 例： `デバイス# 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 つの認証モードがデバイスでサポートされます。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： デバイス> **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： デバイス# **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： デバイス(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： デバイス(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： デバイス(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： デバイス(config)# **end**	特権 EXEC モードに戻ります。
ステップ 7	**show running-config** 例： デバイス# **show running-config**	入力を確認します。
ステップ 8	**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
	デバイス# **copy running-config startup-config**	

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： デバイス> **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： デバイス# **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： デバイス(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
	デバイス(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： デバイス(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： デバイス(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： デバイス(config)# **end**	特権 EXEC モードに戻ります。
ステップ 8	**show ip interface** *interface-id* 例： デバイス# **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： デバイス# **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： デバイス> **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： デバイス# **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： デバイス(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： デバイス(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： デバイス(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： デバイス(config)# **end**	特権 EXEC モードに戻ります。
ステップ 7	**show ip interface** *interface-id* 例： デバイス# 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： デバイス# **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。

サマリーアドレスとスプリットホライズンの構成例

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。

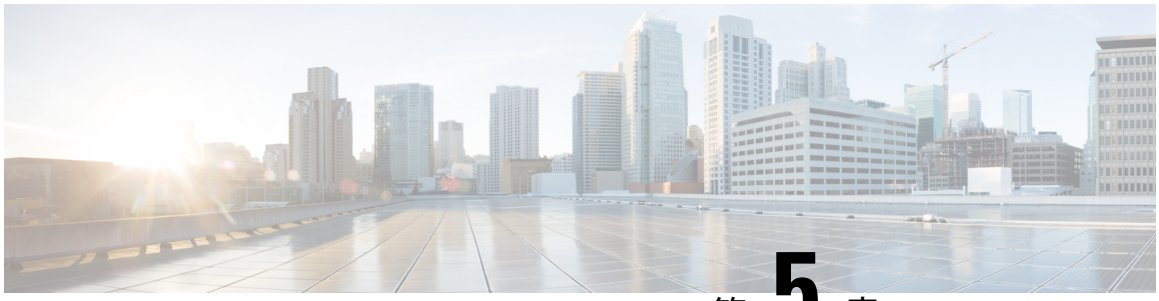
（注） スプリットホライズンが有効である場合、（**ip summary-address rip** ルータ コンフィギュレーション コマンドによって設定される）自動サマリーとインターフェイス サマリー アドレスはともにアドバタイズされません。

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

Routing Information Protocol に関する機能情報

表 *11 : IP* ユニキャスト ルーティングの機能情報

機能名	リリース	機能情報
ルーティング情報プロトコル	Cisco IOS XE Everest 16.5.1a	この機能が導入されました。



第 5 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 NSF

デバイスまたはスイッチスタックは2つのレベルのノンストップ フォワーディング（NSF）をサポートしています。

• OSPF NSF 認識 （102 ページ）
• OSPF NSF 対応 （102 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切り替えが生じた場合、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切り替え後、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。

詳細については、次の URL の『*Cisco Nonstop Forwarding*』を参照してください。
http://www.cisco.com/en/US/docs/ios/ha/configuration/guide/ha-nonstp_fwdg.html

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 *12 : 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
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 ルート タイプのデフォルトはタイプ 2 です。

機能	デフォルト設定
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ま
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ま
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
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	**configure terminal** 例： `デバイス# 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 ospf** process-id 例： `デバイス(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。
ステップ 3	**nsf cisco** [**enforce global**] 例： `デバイス(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 4	**nsf ietf** [**restart-interval** *seconds*] 例： `デバイス(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 5	**network** address wildcard-mask **area** area-id 例： `デバイス(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。

	コマンドまたはアクション	目的
ステップ 6	**end** 例： デバイス(config)# **end**	特権 EXEC モードに戻ります。
ステップ 7	**show ip protocols** 例： デバイス# **show ip protocols**	入力を確認します。
ステップ 8	**copy running-config startup-config** 例： デバイス# **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	**configure terminal** 例： デバイス# **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： デバイス(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。

	コマンドまたはアクション	目的
ステップ 3	**ip ospf** cost 例： `デバイス(config-if)# ip ospf 8`	（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。
ステップ 4	**ip ospf retransmit-interval** seconds 例： `デバイス(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 5	**ip ospf transmit-delay** seconds 例： `デバイス(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 6	**ip ospf priority** number 例： `デバイス(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 7	**ip ospf hello-interval** seconds 例： `デバイス(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 8	**ip ospf dead-interval** seconds 例： `デバイス(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 9	**ip ospf authentication-key** key 例： `デバイス(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 10	**ip ospf message digest-key** keyid **md5** key 例：	（任意）MDS 認証をイネーブルにします。 • *keyid*：1 ～ 255 の ID。

	コマンドまたはアクション	目的
	デバイス(config-if)# ip ospf message digest-key 16 md5 your1pass	• *key*：最大 16 バイトの英数字パスワード
ステップ 11	**ip ospf database-filter all out** 例： デバイス(config-if)# ip ospf database-filter all out	（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 12	**end** 例： デバイス(config)# **end**	特権 EXEC モードに戻ります。
ステップ 13	**show ip ospf interface** [*interface-name*] 例： デバイス# 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 14	**show ip ospf neighbor detail** 例： デバイス# 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 15	**copy running-config startup-config** 例： デバイス# **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	**configure terminal** 例： `デバイス# 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 ospf** process-id 例： `デバイス(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 3	**area** *area-id* **authentication** 例： `デバイス(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 4	**area** *area-id* **authentication message-digest** 例： `デバイス(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 5	**area** *area-id* **stub** [**no-summary**] 例： `デバイス(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 6	**area** *area-id* **nssa** [**no-redistribution**] [**default-information-originate**] [**no-summary**] 例： `デバイス(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。 • **no-redistribution**：サマリー LSA を NSSA に送信しない場合に選択します。

	コマンドまたはアクション	目的
ステップ 7	**area** *area-id* **range** *address mask* 例： `デバイス(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 8	**end** 例： `デバイス(config)# end`	特権 EXEC モードに戻ります。
ステップ 9	**show ip ospf** [*process-id*] 例： `デバイス# 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 10	**show ip ospf** [*process-id* [*area-id*]] **database** 例： `デバイス# 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 11	**copy running-config startup-config** 例： `デバイス# 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	**configure terminal** 例： `デバイス# 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 ospf** *process-id* 例： `デバイス(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。

	コマンドまたはアクション	目的
ステップ 3	**summary-address** *address mask* 例： `デバイス(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 4	**area** *area-id* **virtual-link** *router-id* [**hello-interval** *seconds*] [**retransmit-interval** *seconds*] [**trans**] [[**authentication-key** *key*] | **message-digest-key** *keyid* **md5** *key*]] 例： `デバイス(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 5	**default-information originate** [**always**] [**metric** *metric-value*] [**metric-type** *type-value*] [**route-map** *map-name*] 例： `デバイス(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 6	**ip ospf name-lookup** 例： `デバイス(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 7	**ip auto-cost reference-bandwidth** *ref-bw* 例： `デバイス(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 8	**distance ospf** {[**inter-area** *dist1*] [**inter-area** *dist2*] [**external** *dist3*]} 例： `デバイス(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 9	**passive-interface** *type number* 例： `デバイス(config)# passive-interface gigabitethernet 1/0/6`	（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
ステップ 10	**timers throttle spf** *spf-delay spf-holdtime spf-wait* 例： `デバイス(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 11	**ospf log-adj-changes** 例： `デバイス(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 12	**end** 例： `デバイス(config)# end`	特権 EXEC モードに戻ります。
ステップ 13	**show ip ospf** [*process-id* [*area-id*]] **database** 例： `デバイス# 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 14	**copy running-config startup-config** 例： `デバイス# 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	**configure terminal** 例： `デバイス# 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	**router ospf** *process-id* 例： デバイス(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 3	**timers lsa-group-pacing** *seconds* 例： デバイス(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 4	**end** 例： デバイス(config)# **end**	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： デバイス# **show running-config**	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： デバイス# **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	**configure terminal** 例： デバイス# **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 loopback 0** 例： デバイス(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。

	コマンドまたはアクション	目的
ステップ 3	**ip address** address mask 例： デバイス(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 4	**end** 例： デバイス(config)# **end**	特権 EXEC モードに戻ります。
ステップ 5	**show ip interface** 例： デバイス# show ip interface	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： デバイス# **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 *13 : IP OSPF* 統計情報の表示コマンド

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 ブル エントリ
show ip ospf interface [*interface-name*]	OSPF に関連す す。
show ip ospf neighbor [*interface-name*] [*neighbor-id*] **detail**	OSPF インター す。
show ip ospf virtual-links	OSPF に関連す

OSPF の設定例

例：基本的な OSPF パラメータの設定

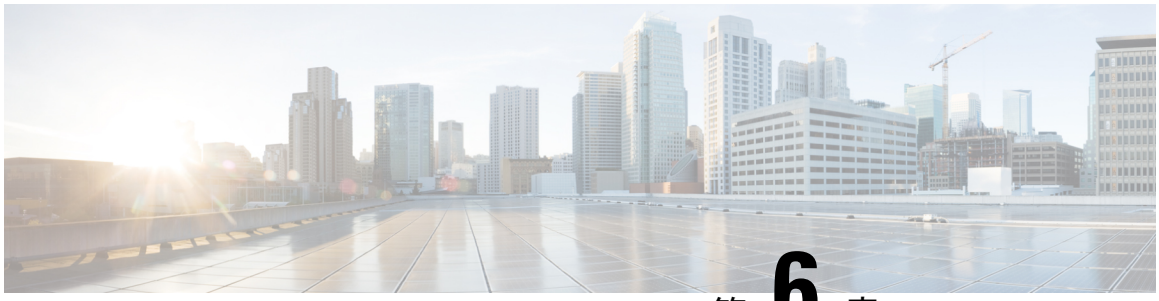
次に、OSPF ルーティング プロセスを設定し、プロセス番号 109 を割り当てる例を示します。

```
デバイス(config)# router ospf 109
デバイス(config-router)# network 131.108.0.0 255.255.255.0 area 24
```

OSPF の機能情報

表 *14 : OSPF* の機能情報

機能名	リリース	機能情報
OSPF（Open Shortest Path First）	Cisco IOS XE Everest 16.5.1a	この機能が導入されました



第 6 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 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 をイネーブルにするには、デバイスまたはアクティブスイッチ上で Network Advantage ライセンスが稼働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、EIGRP NSF 認識を IPv4 に対して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 ライセンスは、EIGRP Cisco NSF ルーティングをサポートし、スタックのアクティブスイッチ切り替え後のコンバージェンスの時間短縮と、トラフィック損失低減を実現します。

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切り替え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、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、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。

（注） EIGRP タブルーティング機能は、接続されたルートまたはサマリールートをルーティングテーブルからネットワーク内の別のdeviceへアドバタイズします。deviceはアクセス レイヤで EIGRP スタブ ルーティングを使用することにより、ほかのタイプのルーティング アドバタイズメントの必要性を排除しています。Network Essentialsライセンスが稼働するdevice上で、Multi-VRF-CE と EIGRP スタブルーティングを同時に設定しようとすると、設定は許可されません。IPv6 EIGRP スタブルーティングは、Network Essentialsライセンスではサポートされません。

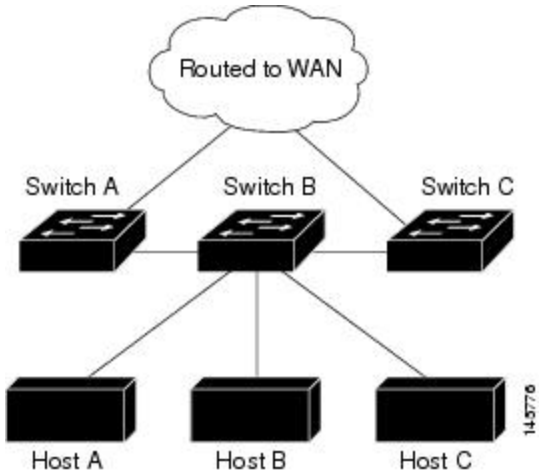
EIGRP スタブルーティングを使用するネットワークでは、ユーザーに対する IP トラフィックの唯一の許容ルートは、EIGRP スタブルーティングを設定しているdevice経由です。deviceは、ユーザーインターフェイスとして設定されているインターフェイスまたは他のデバイスに接続されているインターフェイスにルーテッドトラフィックを送信します。

EIGRP スタブルーティングを使用しているときは、EIGRP を使用してdeviceだけをスタブとして設定するように、分散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deviceから伝播されます。device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、device B は EIGRP スタブルータとして設定されています。デバイス A および C は残りの WAN に接続されています。デバイス 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デバイス A とデバイス C にアドバタイズします。 スイッチ B は、デバイス 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 *5 : EIGRP* スタブ ルータ設定



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 *15 : 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	**configure terminal** 例： `デバイス# 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** autonomous-system 例： `デバイス(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 3	**nsf** 例： `デバイス(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 4	**network** network-number 例： `デバイス(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 5	**eigrp log-neighbor-changes** 例： `デバイス(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 6	**metric weights** *tos k1 k2 k3 k4 k5* 例： `デバイス(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 7	**offset-list** [*access-list number* | *name*] {**in** | **out**} *offset* [*type number*] 例： `デバイス(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 8	**auto-summary** 例： `デバイス(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 9	**interface***interface-id* 例： `デバイス(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 10	**ip summary-address eigrp** *autonomous-system-number address mask* 例： `デバイス(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 11	**end** 例： `デバイス(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 12	**show ip protocols** 例： デバイス# **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 13	**copy running-config startup-config** 例： デバイス# **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	**configure terminal** 例： デバイス# **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： デバイス(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 3	**ip bandwidth-percent eigrp** *percent* 例： デバイス(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 4	**ip summary-address eigrp** *autonomous-system-number address mask* 例： デバイス(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 5	**ip hello-interval eigrp** *autonomous-system-number seconds* 例： `デバイス(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 6	**ip hold-time eigrp** *autonomous-system-number seconds* 例： `デバイス(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 7	**no ip split-horizon eigrp** *autonomous-system-number* 例： `デバイス(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 8	**end** 例： `デバイス(config)# end`	特権 EXEC モードに戻ります。
ステップ 9	**show ip eigrp interface** 例： `デバイス# 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 10	**copy running-config startup-config** 例： `デバイス# 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 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	**configure terminal** 例： `デバイス# 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： `デバイス(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 3	**ip authentication mode eigrp** *autonomous-system***md5** 例： `デバイス(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 4	**ip authentication key-chain eigrp** *autonomous-system key-chain* 例： `デバイス(config-if)# ip authentication key-chain eigrp 105 chain1`	IP EIGRP パケットの認証をイネーブルにします。
ステップ 5	**exit** 例： `デバイス(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	**key chain** *name-of-chain* 例： `デバイス(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 7	**key** *number* 例： `デバイス(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 8	**key-string** *text* 例： `デバイス(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。

	コマンドまたはアクション	目的
ステップ 9	**accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} 例： `デバイス(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 10	**send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} 例： `デバイス(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 11	**end** 例： `デバイス(config)# end`	特権 EXEC モードに戻ります。
ステップ 12	**show key chain** 例： `デバイス# show key chain`	認証キーの情報を表示します。
ステップ 13	**copy running-config startup-config** 例： `デバイス# 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 *16 : IP EIGRP* の *clear* および *show* コマンド

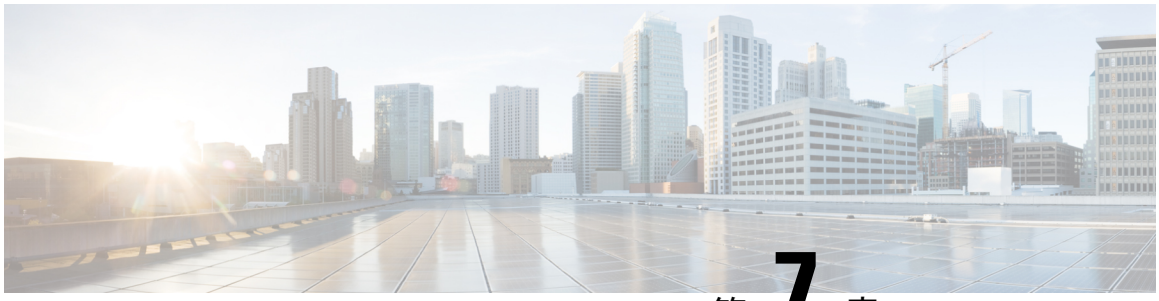
clear ip eigrp neighbors [*if-address* | *interface*]	ネイバー テーブルからネイバ

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

表 *17 : EIGRP* の機能情報

機能名	リリース	機能情報
EIGRP（Enhanced IGRP）	Cisco IOS XE Everest 16.5.1a	この機能が導入されました。



第 7 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

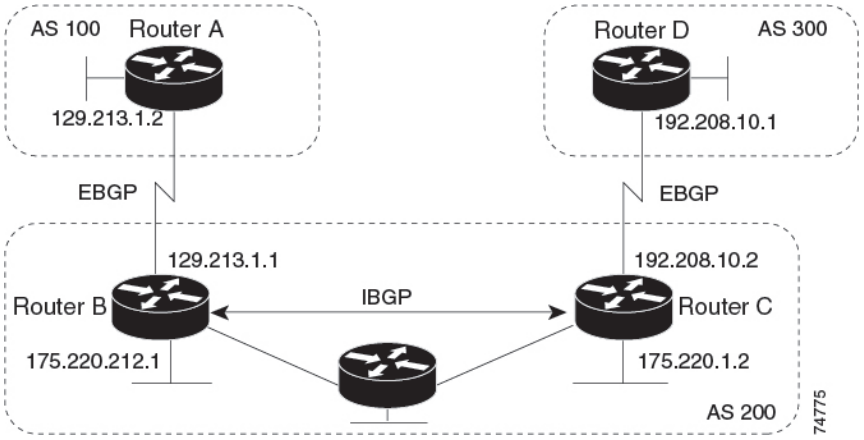
ボーダー ゲートウェイ プロトコル（BGP）は、Exterior Gateway Protocol です。自律システム間で、ループの発生しないルーティング情報交換を保証するドメイン間ルーティング システムを設定するために使用されます。自律システムは、同じ管理下で動作して RIP や OSPF などの Interior Gateway Protocol（IGP）を境界内で実行し、Exterior Gateway Protocol（EGP）を使用して相互接続されるルータで構成されます。BGPバージョン4は、インターネット内でドメイン間ルーティングを行うための標準 EGP です。このプロトコルは、RFC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

BGP コマンドおよびキーワードの詳細については、『*Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols*』の「IP Routing Protocols」を参照してください。

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 *6 : 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またはデバイスが 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、で IPv4 に対してサポートされます。Network Advantage ライセンス。。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。

この機能の詳細については、『*Cisco IOS IP Routing Protocols Configuration Guide, Release 12.4*』の「BGP Nonstop Forwarding (NSF) Awareness」を参照してください。

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 *18 :* 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 BG FIB テーブルのプレフィック 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 デートがリセットされない。
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。ルート マップの詳細については、「Using Route Maps to Redistribute Routing Information」の項を参照してください。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。

コミュニティに基づいて COMMUNITIES 属性および match 句を設定するには、「ルートマップによるルーティング情報の再配信」に記載されている **match community-list** および **set community** ルート マップ コンフィギュレーション コマンドを参照してください。

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。

BGP の追加情報

BGP 設定の詳しい説明については、『*Cisco IOS IP Configuration Guide, Release 12.4*』の「IP Routing Protocols」にある「Configuring BGP」を参照してください。特定コマンドの詳細については、『*Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols, Release 12.4*』を参照してください。

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。すべての特性の詳細については、『*Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols, Release 12.4*』の特定のコマンドを参照してください。

表 *19 : 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 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 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
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）。
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）

機能	デフォルト設定
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（有効値は • 内部ルート アドミニストレーティブ ディスタンス：200（有効値 • 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（有 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ン • 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に対 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、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 • 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 • 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）： • 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 れたルート：32768
NSF[1] 認識	無効にされた[2]。。有効な場合、レイヤ 3 スイッチでは、ハードウ ウェアの変更中に、隣接する NSF 対応ルータからのパケットを転 ができます。
ルート リフレクタ	未設定
同期化（BGP および IGP）	無効

機能	デフォルト設定
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] NSF 認識は、グレースフルリスタートを有効にすることにより、Network Advantage ライセンスを実行するスイッチ上で IPv4 に対して有効にできます

BGP ルーティングのイネーブル化

始める前に

（注） BGP を有効にするには、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	**configure terminal** 例： `デバイス# 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： `デバイス(config)# ip routing`	IP ルーティングを有効にします。
ステップ 3	**router bgp** *autonomous-system* 例： `デバイス(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。
ステップ 4	**network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*] 例： `デバイス(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。

	コマンドまたはアクション	目的
ステップ 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *number* 例： `デバイス(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 6	**neighbor** {*ip-address* | *peer-group-name*} **remove-private-as** 例： `デバイス(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 7	**synchronization** 例： `デバイス(config-router)# synchronization`	（任意）BGP と IGP の同期化を有効にします。
ステップ 8	**auto-summary** 例： `デバイス(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 9	**bgp graceful-restart** 例： `デバイス(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 10	**end** 例： `デバイス(config-router)#end`	特権 EXEC モードに戻ります。
ステップ 11	**show ip bgp network** *network-number* 例： `デバイス# show ip bgp network 10.108.0.0`	設定を確認します。
ステップ 12	**show ip bgp neighbor** 例：	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。

	コマンドまたはアクション	目的
	デバイス# show ip bgp neighbor	スイッチおよびネイバーで NSF 認識がイネーブルである場合は、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認識がイネーブルであり、ネイバーでディセーブルである場合は、次のメッセージが表示されます。 *Graceful Restart Capability: advertised*
ステップ 13	**copy running-config startup-config** 例： デバイス# **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： デバイス# 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： デバイス# 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。
ステップ 3	**clear ip bgp** {***** | *address* | *peer-group-name*} **soft out** 例：	（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、

	コマンドまたはアクション	目的
	デバイス# clear ip bgp * soft out	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： デバイス# 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： デバイス# 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	**configure terminal** 例： デバイス# **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* 例： デバイス(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 3	**bgp best-path as-path ignore** 例： デバイス(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。

	コマンドまたはアクション	目的
ステップ 4	**neighbor** {*ip-address* | *peer-group-name*} **next-hop-self** 例： `デバイス(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 5	**neighbor** {*ip-address* | *peer-group-name*} **weight** *weight* 例： `デバイス(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 6	**default-metric** *number* 例： `デバイス(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 7	**bgp bestpath med missing-as-worst** 例： `デバイス(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 8	**bgp always-compare med** 例： `デバイス(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 9	**bgp bestpath med confed** 例： `デバイス(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。
ステップ 10	**bgp deterministic med** 例： `デバイス(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 11	**bgp default local-preference** *value* 例： `デバイス(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。

	コマンドまたはアクション	目的
ステップ 12	**maximum-paths** *number* 例： `デバイス(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 13	**end** 例： `デバイス(config)# end`	特権 EXEC モードに戻ります。
ステップ 14	**show ip bgp** 例： `デバイス# 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 15	**show ip bgp neighbors** 例： `デバイス# 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 16	**copy running-config startup-config** 例： `デバイス# 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	**configure terminal** 例： `デバイス# 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	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] 例： `デバイス(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 3	**set ip next-hop** *ip-address* [*...ip-address*] [**peer-address**] 例： `デバイス(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 4	**end** 例： `デバイス(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show route-map** [*map-name*] 例： `デバイス# 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 6	**copy running-config startup-config** 例： `デバイス# 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	**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
	デバイス# **configure terminal**	
ステップ 2	**router bgp** *autonomous-system* 例： デバイス(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 3	**neighbor** {*ip-address* | *peer-group name*} **distribute-list** {**access-list-number** | *name*} {**in** | **out**} 例： デバイス(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 4	**neighbor** {*ip-address* | *peer-group name*} **route-map** *map-tag* {**in** | **out**} 例： デバイス(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。
ステップ 5	**end** 例： デバイス(config)# **end**	特権 EXEC モードに戻ります。
ステップ 6	**show ip bgp neighbors** 例： デバイス# show ip bgp neighbors	設定を確認します。
ステップ 7	**copy running-config startup-config** 例： デバイス# **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。（正規表現の作成方法については、『*Cisco IOS Dial Technologies Command Reference, Release 12.4*』の付録「Regular Expressions」を参照してください）。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	**configure terminal** 例： `デバイス# 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 as-path access-list** *access-list-number* {**permit** | **deny**} *as-regular-expressions* 例： `デバイス(config)# ip as-path access-list 1 deny _65535_`	BGP-related アクセス リストを定義します。
ステップ 3	**router bgp** *autonomous-system* 例： `デバイス(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 4	**neighbor** {*ip-address* | *peer-group name*} **filter-list** {*access-list-number* | *name*} {**in** | **out** | **weight** *weight*} 例： `デバイス(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 5	**end** 例： `デバイス(config)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show ip bgp neighbors** [**paths** *regular-expression*] 例： `デバイス# show ip bgp neighbors`	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： `デバイス# 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	**configure terminal** 例： `デバイス# 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 prefix-list** *list-name* [**seq** *seq-value*] **deny** | **permit** *network/len* [**ge** *ge-value*] [**le** *le-value*] 例： `デバイス(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 3	**ip prefix-list** *list-name* **seq** *seq-value* **deny** | **permit** *network/len* [**ge** *ge-value*] [**le** *le-value*] 例： `デバイス(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。

	コマンドまたはアクション	目的
ステップ 4	**end** 例： デバイス(config)# **end**	特権 EXEC モードに戻ります。
ステップ 5	**show ip prefix list** [**detail** | **summary**] *name* [*network/len*] [**seq** *seq-num*] [**longer**] [**first-match**] 例： デバイス# 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 6	**copy running-config startup-config** 例： デバイス# **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. **configure terminal**
2. **ip community-list** *community-list-number* {**permit** | **deny**} *community-number*
3. **router bgp** *autonomous-system*
4. **neighbor** {*ip-address* | *peer-group name*} **send-community**
5. **set comm-list** *list-num* **delete**
6. **exit**
7. **ip bgp-community new-format**
8. **end**
9. **show ip bgp community**
10. **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：	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
	デバイス# **configure terminal**	
ステップ 2	**ip community-list** *community-list-number* {**permit** | **deny**} *community-number* 例： デバイス(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 3	**router bgp** *autonomous-system* 例： デバイス(config)# router bgp 108	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*} **send-community** 例： デバイス(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 5	**set comm-list** *list-num* **delete** 例： デバイス(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 6	**exit** 例： デバイス(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 7	**ip bgp-community new-format** 例： デバイス(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。

	コマンドまたはアクション	目的
ステップ 8	**end** 例： デバイス(config)# **end**	特権 EXEC モードに戻ります。
ステップ 9	**show ip bgp community** 例： デバイス# show ip bgp community	設定を確認します。
ステップ 10	**copy running-config startup-config** 例： デバイス# **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	**configure terminal** 例： デバイス# **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*	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 3	**neighbor** *peer-group-name* **peer-group**	BGP ピア グループを作成します。
ステップ 4	**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 5	**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

	コマンドまたはアクション	目的
		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 6	**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 7	**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 8	**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 9	**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 10	**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 11	**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 12	**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 13	**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 14	**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 15	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 16	**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。

	コマンドまたはアクション	目的
ステップ 17	**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 18	**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。指定できる範囲は 1 ～ 4294967295 秒です。デフォルト値は 180 秒です。
ステップ 19	**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 20	**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 21	**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 22	**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 23	**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 24	**end** 例： `デバイス(config)# end`	特権 EXEC モードに戻ります。
ステップ 25	**show ip bgp neighbors**	設定を確認します。
ステップ 26	**copy running-config startup-config** 例： `デバイス# 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	**configure terminal** 例： `デバイス# 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* 例： `デバイス(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 3	**aggregate-address** *address mask* 例： `デバイス(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 4	**aggregate-address** *address mask* **as-set** 例： `デバイス(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 5	**aggregate-address** *address-mask* **summary-only** 例： `デバイス(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 6	**aggregate-address** *address mask* **suppress-map** *map-name* 例： `デバイス(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 7	**aggregate-address** *address mask* **advertise-map** *map-name* 例：	（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。

	コマンドまたはアクション	目的
	デバイス(config-router)# aggregate-address 10.0.0.0 255.0.0.0 advertise-map map2	
ステップ 8	**aggregate-address** *address mask* **attribute-map** *map-name* 例： デバイス(config-router)# aggregate-address 10.0.0.0 255.0.0.0 attribute-map map3	（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。
ステップ 9	**end** 例： デバイス(config)# **end**	特権 EXEC モードに戻ります。
ステップ 10	**show ip bgp neighbors** [**advertised-routes**] 例： デバイス# show ip bgp neighbors	設定を確認します。
ステップ 11	**copy running-config startup-config** 例： デバイス# **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	**configure terminal** 例： デバイス# **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* 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
	デバイス(config)# router bgp 100	
ステップ 3	**bgp confederation identifier** *autonomous-system* 例： デバイス(config)# bgp confederation identifier 50007	BGP 連合 ID を設定します。
ステップ 4	**bgp confederation peers** *autonomous-system* [*autonomous-system ...*] 例： デバイス(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 5	**end** 例： デバイス(config)# **end**	特権 EXEC モードに戻ります。
ステップ 6	**show ip bgp neighbor** 例： デバイス# show ip bgp neighbor	設定を確認します。
ステップ 7	**show ip bgp network** 例： デバイス# show ip bgp network	設定を確認します。
ステップ 8	**copy running-config startup-config** 例： デバイス# **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	**configure terminal** 例： `デバイス# 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* 例： `デバイス(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 3	**neighbor** {*ip-address* | *peer-group-name*} **route-reflector-client** 例： `デバイス(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 4	**bgp cluster-id** *cluster-id* 例： `デバイス(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 5	**no bgp client-to-client reflection** 例： `デバイス(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 6	**end** 例： `デバイス(config)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show ip bgp** 例： `デバイス# 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。

	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： デバイス# **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	**configure terminal** 例： デバイス# **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* 例： デバイス(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 3	**bgp dampening** 例： デバイス(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 4	**bgp dampening** *half-life reuse suppress max-suppress* [**route-map** *map*] 例： デバイス(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 5	**end** 例： デバイス(config)# **end**	特権 EXEC モードに戻ります。
ステップ 6	**show ip bgp flap-statistics** [{ **regexp** *regexp*} | { **filter-list** *list*} | {*address mask* [**longer-prefix**]}] 例：	（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。

	コマンドまたはアクション	目的
	デバイス# show ip bgp flap-statistics	
ステップ 7	**show ip bgp dampened-paths** 例： デバイス# show pi bgp dampened-paths	（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。
ステップ 8	**clear ip bgp flap-statistics** [{ **regexp** *regexp*} | { **filter-list** *list*} | {*address mask* [**longer-prefix**]} 例： デバイス# 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 9	**clear ip bgp dampening** 例： デバイス# 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 10	**copy running-config startup-config** 例： デバイス# **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 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。

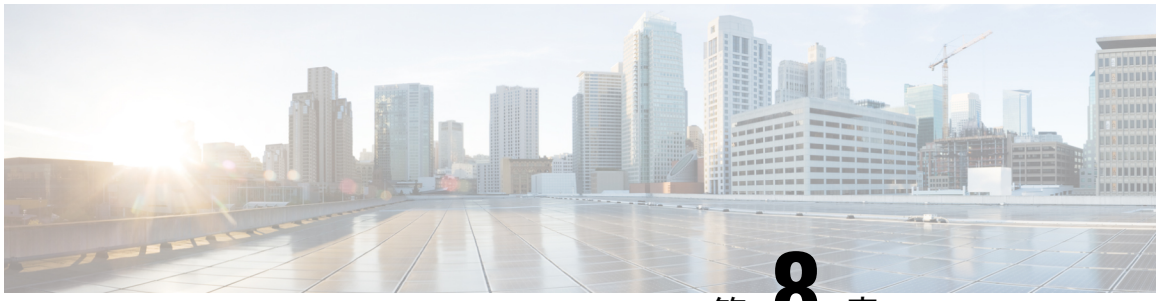
下の図に、BGP を消去および表示するために使用する特権 EXEC コマンドを示します。表示されるフィールドの詳細については、『*Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols, Release 12.4*』を参照してください。

表 *20 : 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
show ip bgp neighbors [*address*]	各ネイバーとの BGP 接続および TCP 接続に関 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
show ip bgp paths	データベース内のすべての BGP パスを表示しま
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。



第 8 章

マルチプロトコル BGP for IPv6 の実装

このモジュールでは、IPv6 用のマルチプロトコルのボーダー ゲートウェイ プロトコル（BGP）を設定する手順について説明します。BGP は、独立したルーティング ポリシーを持つ個別のルーティング ドメイン（自律システム）を接続する場合に主に使用される外部ゲートウェイ プロトコル（EGP）です。BGP の一般的な用途は、サービス プロバイダーに接続してインターネットにアクセスすることです。BGP は、自律システム内で使用することもできます。このタイプの BGP は、内部 BGP（iBGP）と呼ばれます。マルチプロトコル BGP は、複数のネットワーク層プロトコル アドレス ファミリ（IPv6 アドレス ファミリなど）、および IP マルチキャスト ルートに関するルーティング情報を伝送する拡張 BGP です。すべての BGP コマンドおよびルーティング ポリシー機能をマルチプロトコル BGP で使用できます。

マルチプロトコル BGP for IPv6 の実装に関する情報

Multiprotocol BGP Extensions for IPv6

マルチプロトコル BGP は、IPv6 でサポートされている外部ゲートウェイ プロトコル（EGP）です。マルチプロトコル BGP for IPv6 拡張では、IPv4 BGP と同じ機能および機能性の多くがサポートされています。マルチプロトコル BGP に対する IPv6 拡張には、IPv6 アドレス ファミリ、ネットワーク層到達可能性情報（NLRI）、および IPv6 アドレスを使用するネクスト ホップ（宛先パス内の次のデバイス）属性のサポートが含まれています。

リンクローカル アドレスを使用した IPv6 マルチプロトコル BGP ピアリング

リンクローカル アドレスを使用して、2 つの IPv6 デバイス（ピア）間で IPv6 マルチプロトコル BGP を設定できます。この機能を動作させるには、**neighbor update-source** コマンドを使用

してネイバーのインターフェイスを識別する必要があり、IPv6 グローバル ネクスト ホップを設定するようにルート マップを設定する必要があります。

IPv6 マルチキャスト アドレス ファミリのマルチプロトコル BGP

IPv6 マルチキャスト アドレス ファミリのマルチプロトコル BGP 機能では、マルチプロトコル BGP for IPv6 拡張を提供し、IPv4 BGP と同じ機能と機能性をサポートします。マルチキャスト BGP に対する IPv6 拡張には、IPv6 マルチキャスト アドレス ファミリ、ネットワーク層到達可能性情報（NLRI）、および IPv6 アドレスを使用するネクスト ホップ（宛先へのパス内の次のルータ）属性のサポートが含まれています。

マルチキャスト BGP は、ドメイン間 IPv6 マルチキャストの配布を可能にする、拡張された BGP です。マルチプロトコル BGP では、複数のネットワーク層プロトコル アドレス ファミリ（IPv6 アドレス ファミリなど）および IPv6 マルチキャスト ルートに関するルーティング情報を伝送します。IPv6 マルチキャスト アドレス ファミリには、IPv6 PIM プロトコルによる RPF ルックアップに使用される複数のルートが含まれており、マルチキャスト BGP IPv6 は、同じドメイン間転送を提供します。ユニキャスト BGP が学習したルートは IPv6 マルチキャストには使用されないため、ユーザーは、BGP で IPv6 マルチキャストを使用する場合は、マルチプロトコル BGP for IPv6 マルチキャストを使用する必要があります。

マルチキャスト BGP 機能は、個別のアドレス ファミリ コンテキストを介して提供されます。Subsequent Address Family Identifier（SAFI）では、属性で伝送されるネットワーク層到達可能性情報のタイプに関する情報を提供します。マルチプロトコル BGP ユニキャストでは SAFI 1 メッセージを使用し、マルチプロトコル BGP マルチキャストでは SAFI 2 メッセージを使用します。SAFI 1 メッセージは、ルートは IP ユニキャストだけに使用でき、IP マルチキャストには使用できないことを示します。この機能があるため、IPv6 ユニキャスト RIB 内の BGP ルートは、IPv6 マルチキャスト RPF ルックアップでは無視される必要があります。

IPv6 マルチキャスト RPF ルックアップを使用して、異なるポリシーおよびトポロジ（IPv6 ユニキャストとマルチキャストなど）を設定するために、個別の BGP ルーティング テーブルが維持されています。マルチキャスト RPF ルックアップは、IP ユニキャスト ルート ルックアップと非常によく似ています。

IPv6 マルチキャスト BGP テーブルと関連付けられている MRIB はありません。ただし、必要な場合、IPv6 マルチキャスト BGP は、ユニキャスト IPv6 RIB で動作します。マルチキャスト BGP では、IPv6 ユニキャスト RIB へのルートの挿入や更新は行いません。

MP-BGP IPv6 アドレス ファミリのノンストップ フォワーディングおよびグレースフル リスタート

グレースフル リスタート機能は、IPv6 BGP ユニキャスト、IPv6 BGP マルチキャスト、および VPNv6 アドレス ファミリでサポートされており、BGP IPv6 用の Cisco ノンストップ フォワーディング（NSF）機能をイネーブルにします。BGP グレースフル リスタート機能を使用すると、TCP 状態を維持することなく、BGP ルーティング テーブルをピアから回復できます。

NSF では、ルーティング プロトコルのコンバージェンス時にも引き続きパケットが転送されるため、スイッチオーバー時のルート フラップが回避されます。転送は、アクティブ RP とスタンバイ RP 間で FIB を同期することで維持されます。スイッチオーバー時、転送は FIB を使

用して維持されます。RIB の同期は維持されないため、RIB はスイッチオーバー時に空になります。RIB は、ルーティング プロトコルによって再入力され、次に、NSF_RIB_CONVERGED レジストリ コールを使用して RIB コンバージェンスに関する情報を FIB に伝えますFIB テーブルは、RIBから更新され、古いエントリが削除されます。RIBは、ルーティングプロトコルが RIB のコンバージェンスの通知に失敗した場合、RP スイッチオーバー時にフェールセーフタイマーを開始します。

Cisco BGP Address Family Identifier（AFI）モデルは、モジュラ式でスケーラブルな設計となっており、複数の AFI 設定および Subsequent Address Family Identifier（SAFI）設定をサポートするように設計されています。

マルチプロトコル BGP for IPv6 の設定方法

IPv6 BGP ルーティング プロセスおよび BGP ルータ ID の設定

IPv6 BGP ルーティング プロセスを設定し、オプションの BGP 対応デバイス用 BGP ルータ ID を設定するには、次の作業を実行します。

BGP では、ルータ ID を使用して、BGP スピーキング ピアを識別します。BGP ルータ ID は、32 ビット値であり、多くの場合、IPv4 アドレスで表されます。デフォルトでは、ルータ ID は、デバイスのループバック インターフェイスの IPv4 アドレスに設定されます。デバイス上でループバック インターフェイスが設定されていない場合は、BGP ルータ ID を表すためにデバイスの物理インターフェイスに設定されている最上位の IPv4 アドレスがソフトウェアによって選択されます。

IPv6 だけが有効になっているデバイス（IPv4 アドレスを持っていないデバイス）で BGP を設定する場合、そのデバイスの BGP ルータ ID を手動で設定する必要があります。IPv4 アドレス構文を使用して 32 ビット値で表される BGP ルータ ID は、デバイスの BGP ピアで一意である必要があります。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp default ipv4-unicast**
5. **bgp router-id** *ip-address*
6. **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：	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 65000`	BGP ルーティング プロセスを設定し、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	**no bgp default ipv4-unicast** 例： `Device(config-router)# no bgp default ipv4-unicast`	前の手順で指定した BGP ルーティング プロセスの IPv4 ユニキャスト アドレス ファミリを無効にします。 （注） IPv4 ユニキャスト アドレス ファミリのルーティング情報は、**neighbor remote-as** コマンドで設定された各 BGP ルーティング セッションに対して、デフォルトでアドバタイズされます。ただし、**neighbor remote-as** コマンドを設定する前に、**no bgp default ipv4-unicast** コマンドを設定した場合は例外です。
ステップ 5	**bgp router-id** *ip-address* 例： `Device(config-router)# bgp router-id 192.168.99.70`	（任意）固定 32 ビット ルータ ID を、BGP を実行するローカル デバイスの ID として設定します。 （注） **bgp router-id** コマンドを使用してルータ ID を設定すると、アクティブな BGP ピアリング セッションがすべてリセットされます。
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 モードに戻ります。

2 つのピア間での IPv6 マルチプロトコル BGP の設定

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** *as-number*
4. **neighbor** {*ip-address* | *ipv6-address* [**%**] | *peer-group-name*} **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number ...*]
5. **address-family ipv6** [**unicast** | **multicast**]
6. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address* **%**} **activate**
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	**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-number* 例： `Device(config)# router bgp 65000`	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	**neighbor** {*ip-address* | *ipv6-address* [**%**] | *peer-group-name*} **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number ...*] 例： `Device(config-router)# neighbor 2001:DB8:0:CC00::1 remote-as 64600`	指定された自律システムのネイバーの IPv6 アドレスを、ローカル デバイスの 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** [**unicast** | **multicast**] 例： `Device(config-router)# address-family ipv6`	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。 • **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** コマンドにキーワードが指定されていない場合、デバイスは 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。

	コマンドまたはアクション	目的
		• **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** {*ip-address* | *peer-group-name* | *ipv6-address* **%**} **activate** 例： `Device(config-router-af)# neighbor 2001:DB8:0:CC00::1 activate`	ローカル デバイスとの間で IPv6 アドレス ファミリのプレフィックスを交換できるようにネイバーを設定します。
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。

リンクローカルアドレスを使用した 2 つのピア間の IPv6 マルチプロトコル BGP の設定

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。

デフォルトでは、**neighbor route-map** コマンドを使用してルータ コンフィギュレーション モードで適用されるルート マップは、IPv4 ユニキャスト アドレス プレフィックスだけに適用されます。IPv6 アドレス ファミリなどのその他のアドレス ファミリのルート マップは、**neighbor route-map** コマンドを使用してアドレス ファミリ コンフィギュレーション モードで適用される必要があります。ルート マップは、指定したアドレス ファミリの下にあるネイバーの着信ルーティング ポリシーまたは発信ルーティング ポリシーとして適用されます。各アドレス ファミリ タイプで個別のルート マップを設定すると、各アドレス ファミリの複雑なポリシーまたはさまざまなポリシーを簡単に管理できるようになります。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast** | **vpnv6**
7. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} activate
8. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **route-map** *map-name* {**in** | **out**

9. **exit**
10. **exit**
11. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
12. **match ipv6 address** {**prefix-list** *prefix-list-name* | *access-list-name*
13. **set ipv6 next-hop** *ipv6-address* [*link-local-address*] [**peer-address**
14. **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 65000`	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	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* 例： `Device(config-router)# neighbor 2001:DB8:0000:0000:0000:0000:0000:0111 remote-as 64600`	指定したリモート自律システム内のネイバーのリンクローカル 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。 • **neighbor remote-as** コマンドの *ipv6-address* 引数は、RFC 2373 に記述されている形式のリンクローカル IPv6 アドレスにする必要があります。コロン区切りの 16 ビット値を使用して、アドレスを 16 進数で指定します。
ステップ 5	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* 例： `Device(config-router)# neighbor 2001:DB8:0000:0000:0000:0000:0000:0111 update-source gigabitethernet 0/0/0`	ピアリングが発生するリンクローカル アドレスを指定します。 • ネイバーへの接続が複数存在し、**neighbor update-source** コマンドで *interface-type* 引数と *interface-number* 引数を使用してネイバー インターフェイスを指定していない場合は、リンクローカルアドレスを使用してネイバーとの TCP 接続を確立することはできません。

	コマンドまたはアクション	目的
ステップ 6	**address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast** | **vpnv6** 例： `Device(config-router)# address-family ipv6`	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。 • **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** コマンドに **unicast** キーワードが指定されていない場合、ルータは 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 7	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} activate 例： `Device(config-router-af)# neighbor 2001:DB8:0000:0000:0000:0000:0000:0111 activate`	ネイバーが、指定したリンクローカル アドレスを使用して IPv6 アドレス ファミリのプレフィックスをローカル ルータと交換できるようにします。
ステップ 8	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **route-map** *map-name* {**in** | **out** 例： `Device(config-router-af)# neighbor 2001:DB8:0000:0000:0000:0000:0000:0111 route-map nh6 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。
ステップ 9	**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 コンフィギュレーション モードに戻ります。
ステップ 10	**exit** 例： `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 11	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] 例： `Device(config)# route-map nh6 permit 10`	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。
ステップ 12	**match ipv6 address** {**prefix-list** *prefix-list-name* | *access-list-name* 例：	プレフィックス リストで許可されている宛先 IPv6 ネットワーク番号アドレスを持つすべてのルートを配布するか、パケットに対してポリシー ルーティングを実行します。

	コマンドまたはアクション	目的
	`Device(config-route-map)# match ipv6 address prefix-list list1`	
ステップ 13	**set ipv6 next-hop** *ipv6-address* [*link-local-address*] [**peer-address** 例： `Device(config-route-map)# set ipv6 next-hop 2001:DB8::1`	ポリシー ルーティング用のルート マップの match 句を渡す IPv6 パケットのピアにアドバタイズされるネクスト ホップを上書きします。 • *ipv6-address* 引数には、ネクスト ホップの IPv6 グローバル アドレスを指定します。隣接ルータである必要はありません。 • *link-local-address* 引数には、ネクスト ホップの IPv6 リンクローカル アドレスを指定します。隣接ルータである必要があります。 （注） ルート マップによって、BGP アップデートに IPv6 ネクストホップ アドレス（グローバルおよびリンクローカル）が設定されます。ルート マップが設定されていない場合、デフォルトでは、BGP アップデートのネクストホップ アドレスは未指定の IPv6 アドレス（::）に設定され、ピアで拒否されます。手順 5 の **neighbor update-source** コマンドでネイバー インターフェイス（*interface-type* 引数）を指定した後に、**set ipv6 next-hop** コマンドでグローバル IPv6 ネクストホップ アドレス（*ipv6-address* 引数）だけを指定した場合は、*interface-type* 引数で指定したインターフェイスのリンクローカル アドレスが BGP アップデートのネクスト ホップとして含まれます。したがって、リンクローカル アドレスを使用する複数の BGP ピアに必要となるのは、BGP アップデートにグローバル IPv6 ネクストホップ アドレスを設定する 1 つのルート マップだけとなります。
ステップ 14	**end** 例： `Device(config-route-map)# end`	現在のルート マップ コンフィギュレーション モードを終了して、特権 EXEC モードに戻ります。

トラブルシューティングのヒント

このタスクを実行してもピアリングが確立されない場合は、ルートマップ **set ipv6 next-hop** コマンドが欠落している可能性があります。**debug bgp ipv6 update** コマンドを使用して、アップデートに関するデバッグ情報を表示すると、ピアリング状態の確認に役立ちます。

IPv6 マルチプロトコル BGP ピア グループの設定

- 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。
- デフォルトでは、**neighbor peer-group** コマンドを使用してルータ コンフィギュレーション モードで定義されたピアグループは、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、ピアグループをアクティブ化する必要があります。
- ピア グループのメンバは、そのピア グループのアドレス プレフィックス設定を自動的に継承します。
- アクティブな IPv4 ネイバーは、アクティブな IPv6 ネイバーと同じピア グループに存在することはできません。IPv4 ピアと IPv6 ピア用に個別のピア グループを作成します。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** *peer-group-name* **peer-group**
5. **neighbor** {*ip-address* | *ipv6-address*[**%**] | *peer-group-name*} **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number ...*]
6. **address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast** | **vpnv6**
7. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address* **%**} **activate**
8. **neighbor** *ip-address* | *ipv6-address*} **send-label**
9. **neighbor** {*ip-address* | *ipv6-address*} **peer-group** *peer-group-name*
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	**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-number* 例： `Device(config)# router bgp 65000`	指定した BGP 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	**neighbor** *peer-group-name* **peer-group** 例： `Device(config-router)# neighbor group1 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 5	**neighbor** {*ip-address* | *ipv6-address*[**%**] | *peer-group-name*} **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number* ...] 例： `Device(config-router)# neighbor 2001:DB8:0:CC00::1 remote-as 64600`	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 6	**address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast** | **vpnv6** 例： `Device(config-router)# address-family ipv6 unicast`	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。 • **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** コマンドに **unicast** キーワードが指定されていない場合、デバイスは 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 7	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address* **%**} **activate** 例：	ネイバーが、指定したファミリ タイプのプレフィックスをネイバーおよびローカル ルータと交換できるようにします。

	コマンドまたはアクション	目的
	`Device(config-router-af)# neighbor 2001:DB8:0:CC00::1 activate`	• 各ネイバーでの追加の設定手順を回避するために、この手順の代替として、*peer-group-name* 引数を指定して **neighbor activate** コマンドを使用します。
ステップ 8	**neighbor** *ip-address* | *ipv6-address*} **send-label** 例： `Device(config-router-af)# neighbor 192.168.99.70 send-label`	BGP ルートとともに MPLS ラベルを送信するデバイスの機能をアドバタイズします。 • IPv6 アドレス ファミリ コンフィギュレーション モードでは、このコマンドによって、BGP の IPv6 プレフィックスのアドバタイズ時に集約ラベルをバインドおよびアドバタイズできるようになります。
ステップ 9	**neighbor** {*ip-address* | *ipv6-address*} **peer-group** *peer-group-name* 例： `Device(config-router-af)# neighbor 2001:DB8:0:CC00::1 peer-group group1`	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 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。

IPv6 マルチプロトコル BGP プレフィックスのルートマップの設定

- 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。
- デフォルトでは、**neighbor route-map** コマンドを使用してルータ コンフィギュレーション モードで適用されるルート マップは、IPv4 ユニキャスト アドレス プレフィックスだけに適用されます。IPv6 アドレス ファミリなどのその他のアドレス ファミリのルート マップは、**neighbor route-map** コマンドを使用してアドレス ファミリ コンフィギュレーション モードで適用される必要があります。ルート マップは、指定したアドレス ファミリの下にあるネイバーの着信ルーティング ポリシーまたは発信ルーティング ポリシーとして適用されます。各アドレス ファミリ タイプで個別のルート マップを設定すると、各アドレス ファミリの複雑なポリシーまたはさまざまなポリシーを簡単に管理できるようになります。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
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** {*ip-address* | *peer-group-name* | *ipv6-address* **%**} **activate**
7. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address* [**%**]} **route-map** *map-name* {**in** | **out**}
8. **exit**
9. **exit**
10. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
11. **match ipv6 address** {**prefix-list** *prefix-list-name* | *access-list-name*}
12. **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** *as-number* 例： `Device(config)# router bgp 65000`	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	**neighbor** {*ip-address* | *ipv6-address*[**%**] | *peer-group-name*} **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number* ...] 例： `Device(config-router)# neighbor 2001:DB8:0:cc00::1 remote-as 64600`	指定したリモート自律システム内のネイバーのリンクローカル IPv6 アドレスをローカルデバイスの 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： `Device(config-router)# address-family ipv6`	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。 • **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** コマンドに **unicast** キーワー

	コマンドまたはアクション	目的
		ドが指定されていない場合、デバイスは 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** {*ip-address* | *peer-group-name* | *ipv6-address* **%**} **activate** 例： `Device(config-router-af)# neighbor 2001:DB8:0:cc00::1 activate`	ネイバーが、指定したリンクローカルアドレスを使用して IPv6 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 7	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address* [**%**]} **route-map** *map-name* {**in** | **out**} 例： `Device(config-router-af)# neighbor 2001:DB8:0:cc00::1 route-map rtp in`	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 コンフィギュレーション モードに戻ります。
ステップ 9	**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 10	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] 例： `Device(config)# route-map rtp permit 10`	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。 • **match** コマンドを使用して、この手順を実行します。
ステップ 11	**match ipv6 address** {**prefix-list** *prefix-list-name* | *access-list-name*} 例： `Device(config-route-map)# match ipv6 address prefix-list list1`	プレフィックス リストで許可されている宛先 IPv6 ネットワーク番号アドレスを持つすべてのルートを配布するか、パケットに対してポリシー ルーティングを実行します。

	コマンドまたはアクション	目的
ステップ 12	**end** 例： `Device(config-route-map)# end`	現在のルート マップ コンフィギュレーション モードを終了して、特権 EXEC モードに戻ります。

IPv6 マルチプロトコル BGP へのプレフィックスの再配布

再配布とは、あるルーティング プロトコルから別のルーティング プロトコルにプレフィックスを再配布、つまり挿入するプロセスです。ここでは、あるルーティング プロトコルのプレフィックスを IPv6 マルチプロトコル BGP に挿入する方法について説明します。具体的には、**redistribute** ルータ コンフィギュレーション コマンドを使用して IPv6 マルチプロトコル BGP に再配布されたプレフィックスは、IPv6 ユニキャスト データベースに挿入されます。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast** | **vpnv6**]
5. **redistribute bgp** [*process-id*] [**metric** *metric-value*] [**route-map** *map-name*]
6. **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** *as-number* 例： `Device(config)# router bgp 65000`	指定した BGP 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	**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`	• **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** コマンドにキーワードが指定されていない場合、デバイスは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 5	**redistribute bgp** [*process-id*] [**metric** *metric-value*] [**route-map** *map-name*] 例： `Device(config-router-af)# redistribute bgp 64500 metric 5`	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 6	**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。

IPv6 マルチプロトコル BGP へのルートのアドバタイズ

デフォルトでは、**network** コマンドを使用してルータ コンフィギュレーション モードで定義されたネットワークは、IPv4ユニキャストデータベースに挿入されます。IPv6 BGPデータベースなど、別のデータベースにネットワークを挿入するには、IPv6 BGP データベースの場合と同様に、そのデータベースについて、アドレス ファミリ コンフィギュレーション モードで **network** コマンドを使用してネットワークを定義する必要があります。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast** | **vpnv6**]
5. **network** {*network-number* [**mask** *network-mask*] | *nsap-prefix*} [**route-map** *map-tag*]
6. **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	**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 65000`	指定した BGP 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	**address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast** | **vpnv6**] 例： `Device(config-router)# address-family ipv6 unicast`	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。 • **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** コマンドにキーワードが指定されていない場合、デバイスは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 5	**network** {*network-number* [**mask** *network-mask*] | *nsap-prefix*} [**route-map** *map-tag*] 例： `Device(config-router-af)# network 2001:DB8::/24`	指定したプレフィックスを IPv6 BGP データベースにアドバタイズ（挿入）します（まず、IPv6 ユニキャスト ルーティング テーブルでルートを見つける必要があります）。 • 前の手順で指定したアドレス ファミリのデータベースにプレフィックスが挿入されます。 • ルートには指定したプレフィックスによって「local origin」のタグが付けられます。 • **network** コマンドの *ipv6-prefix* 引数には、RFC 2373 に記載されている形式を使用する必要があります。その場合、16 ビット値を使用した 16 進数でアドレスを指定し、コロンで区切ります。 • *prefix-length* 引数は、アドレスのうち連続する上位何ビットがプレフィックス（アドレスのネットワーク部）を構成するかを示す 10 進数値です。10 進数値の前にスラッシュ記号が必要です。

	コマンドまたはアクション	目的
ステップ 6	**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、デバイスをルータ コンフィギュレーション モードに戻します。 • この手順を繰り返して、ルータ コンフィギュレーション モードを終了し、デバイスをグローバル コンフィギュレーション モードに戻します。

IPv6 BGP ピア間での IPv4 ルートのアドバタイズ

IPv6 ネットワークによって 2 つの別々の IPv4 ネットワークが接続されている場合は、IPv6 を使用して IPv4 ルートをアドバタイズできます。IPv4 アドレス ファミリ内の IPv6 アドレスを使用して、ピアリングを設定します。アドバタイズされるネクストホップは、通常、到着不能であるため、スタティック ルートまたはインバウンド ルート マップを使用してネクスト ホップを設定します。2 つの IPv4 ピア間での IPv6 ルートのアドバタイズも同じモデルを使用して実行できます。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** *peer-group-name* **peer-group**
5. **neighbor** {*ip-address* | *ipv6-address*[**%**] | *peer-group-name*} **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number ...*]
6. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*]
7. **neighbor** *ipv6-address* **peer-group** *peer-group-name*
8. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address* [**%**]} **route-map** *map-name* {**in** | **out**}
9. **exit**
10. **exit**
11. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
12. **set ip next-hop** *ip-address* [*...ip-address*] [**peer-address**]
13. **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）。

	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** *as-number* 例： `Device(config)# router bgp 65000`	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	**neighbor** *peer-group-name* **peer-group** 例： `Device(config-router)# neighbor 6peers 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 5	**neighbor** {*ip-address* | *ipv6-address*[**%**] | *peer-group-name*} **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number* ...] 例： `Device(config-router)# neighbor 6peers remote-as 65002`	指定された自律システムのネイバーの IPv6 アドレスを、ローカル デバイスの 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 6	**address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] 例： `Device(config-router)# address-family ipv4`	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、標準 IPv4 アドレス プレフィックスを使用するルーティング セッションを設定します。
ステップ 7	**neighbor** *ipv6-address* **peer-group** *peer-group-name* 例： `Device(config-router-af)# neighbor 2001:DB8:1234::2 peer-group 6peers`	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 8	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address* [**%**]} **route-map** *map-name* {**in** | **out**} 例： `Device(config-router-af)# neighbor 6peers route-map rmap 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。

	コマンドまたはアクション	目的
ステップ 9	**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、デバイスをルータ コンフィギュレーション モードに戻します。
ステップ 10	**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、デバイスをグローバル コンフィギュレーション モードに戻します。
ステップ 11	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] 例： `Device(config)# route-map rmap permit 10`	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。
ステップ 12	**set ip next-hop** *ip-address* [*...ip-address*] [**peer-address**] 例： `Device(config-route-map)# set ip next-hop 10.21.8.10`	IPv4 パケットのピアにアドバタイズされるネクスト ホップをオーバーライドします。
ステップ 13	**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。

マルチキャスト BGP ルートの BGP アドミニストレーティブ ディスタンスの割り当て

RPF ルックアップでユニキャスト ルートとの比較に使用されるマルチキャスト BGP ルートのアドミニストレーティブ ディスタンスを指定するには、次の作業を実行します。

⚠

注意 BGP 内部ルートのアドミニストレーティブ ディスタンスの変更は推奨されません。発生する可能性のある 1 つの問題は、ルーティング テーブルの不整合が累積され、それによってルーティングが中断する可能性があることです。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast** | **vpnv6**]

5. **distance bgp** *external-distance internal-distance local-distance*
6. **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** *as-number* 例： `Device(config)# router bgp 65000`	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	**address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast** | **vpnv6**] 例： `Device(config-router)# address-family ipv6`	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。 • **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** コマンドに **unicast** キーワードが指定されていない場合、ルータは 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 5	**distance bgp** *external-distance internal-distance local-distance* 例： `Device(config-router-af)# distance bgp 10 50 100`	BGP ルートのアドミニストレーティブ ディスタンスを設定します。
ステップ 6	**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。

IPv6 マルチキャスト BGP アップデートの生成

ピアから受信したユニキャスト IPv6 アップデートに対応する IPv6 マルチキャスト BGP アップデートを生成するには、次の作業を実行します。

MBGP 変換アップデート機能は、一般に、BGP 対応ルータだけを持つカスタマー サイト（つまり、ルータを MBGP 対応イメージにアップグレードしていない、またはアップグレードできないカスタマー サイト）とピアリングする MBGP 対応ルータで使用されます。そのカスタマー サイトでは MBGP アドバタイズメントを発信できないため、カスタマー サイトがピアリングするルータは、BGP プレフィックスを、マルチキャストソース Reverse Path Forwarding（RPF）ルックアップに使用される MBGP プレフィックスに変換します。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast** | **vpnv6**
5. **neighbor** *ipv6-address* **translate-update ipv6 multicast** [**unicast**
6. **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** *as-number* 例： `Device(config)# router bgp 65000`	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	**address-family ipv6** [**vrf** *vrf-name*] [**unicast**	**multicast**	**vpnv6** 例： `Device(config-router)# address-family ipv6`	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。 • **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** コマンドに **unicast** キーワードが指定されていない場合、ルータは IPv6 ユニ

	コマンドまたはアクション	目的
		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 5	**neighbor** *ipv6-address* **translate-update ipv6 multicast** [**unicast** 例： `Device(config-router-af)# neighbor 2001:DB8::2 translate-update ipv6 multicast`	ピアから受信したユニキャスト IPv6 アップデートに対応するマルチプロトコル IPv6 BGP アップデートを生成します。
ステップ 6	**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。

IPv6 BGP グレースフル リスタート機能の設定

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **bgp graceful-restart** [**restart-time** *seconds* | **stalepath-time** *seconds*] [**all**]
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`	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-number* 例：	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。

	コマンドまたはアクション	目的
	`Device(config)# router bgp 65000`	
ステップ 4	**bgp graceful-restart** [**restart-time** *seconds* | **stalepath-time** *seconds*] [**all**] 例： `Device(config-router)# bgp graceful-restart`	BGP グレースフル リスタート機能をイネーブルにします。
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。

IPv6 BGP セッションのリセット

手順の概要

1. **enable**
2. **clear bgp ipv6** {**unicast** | **multicast**} {***** | *autonomous-system-number* | *ip-address* | *ipv6-address* | **peer-group** *peer-group-name*} [**soft**] [**in** | **out**]
3. **clear bgp ipv6** {**unicast** | **multicast**} **external** [**soft**] [**in** | **out**]
4. **clear bgp ipv6** {**unicast** | **multicast**} **peer-group** *name*
5. **clear bgp ipv6** {**unicast** | **multicast**} **dampening** [*ipv6-prefix*/*prefix-length*]
6. **clear bgp ipv6** {**unicast** | **multicast**} **flap-statistics** [*ipv6-prefix*/*prefix-length* | **regexp** *regexp* | **filter-list** *list*]

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	**clear bgp ipv6** {**unicast** | **multicast**} {***** | *autonomous-system-number* | *ip-address* | *ipv6-address* | **peer-group** *peer-group-name*} [**soft**] [**in** | **out**] 例： `Device# clear bgp ipv6 unicast peer-group marketing soft out`	IPv6 BGP セッションをリセットします。
ステップ 3	**clear bgp ipv6** {**unicast** | **multicast**} **external** [**soft**] [**in** | **out**]	外部 IPv6 BGP ピアをクリアします。

	コマンドまたはアクション	目的
	例： `Device# clear bgp ipv6 unicast external soft in`	
ステップ 4	**clear bgp ipv6** {**unicast** | **multicast**} **peer-group** *name* 例： `Device# clear bgp ipv6 unicast peer-group marketing`	IPv6 BGP ピア グループのすべてのメンバをクリアします。
ステップ 5	**clear bgp ipv6** {**unicast** | **multicast**} **dampening** [*ipv6-prefix*/*prefix-length*] 例： `Device# clear bgp ipv6 unicast dampening 2001:DB8::/64`	IPv6 BGP ルート ダンプニング情報をクリアし、抑制されたルートの抑制を解除します。
ステップ 6	**clear bgp ipv6** {**unicast** | **multicast**} **flap-statistics** [*ipv6-prefix*/*prefix-length* | **regexp** *regexp* | **filter-list** *list*] 例： `Device# clear bgp ipv6 unicast flap-statistics filter-list 3`	IPv6 BGP フラップ統計情報をクリアします。

IPv6 マルチプロトコル BGP の構成の確認

手順の概要

1. **enable**
2. **show bgp ipv6 unicast** | **multicast**} [*ipv6-prefix*/*prefix-length*] [**longer-prefixes**] [**labels**]
3. **show bgp ipv6** {**unicast** | **multicast**} **summary**
4. **show bgp ipv6** {**unicast** | **multicast**} **dampening dampened-paths**
5. **debug bgp ipv6** {**unicast** | **multicast**} **dampening**[**prefix-list** *prefix-list-name*]
6. **debug bgp ipv6 unicast** | **multicast**} **updates**[*ipv6-address*] [**prefix-list** *prefix-list-name*] [**in**| **out**]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `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	**show bgp ipv6 unicast** | **multicast**} [*ipv6-prefix*/*prefix-length*] [**longer-prefixes**] [**labels**] 例： `Device> show bgp ipv6 unicast`	（任意）IPv6 BGP ルーティング テーブルのエントリを表示します。
ステップ 3	**show bgp ipv6** {**unicast** | **multicast**} **summary** 例： `Device> show bgp ipv6 unicast summary`	（任意）すべての IPv6 BGP 接続のステータスを表示します。
ステップ 4	**show bgp ipv6** {**unicast** | **multicast**} **dampening dampened-paths** 例： `Device> show bgp ipv6 unicast dampening dampened-paths`	（任意）IPv6 BGP ダンプされたルートを表示します。
ステップ 5	**debug bgp ipv6** {**unicast** | **multicast**} **dampening**[**prefix-list** *prefix-list-name*] 例： `Device# debug bgp ipv6 unicast dampening`	（任意）IPv6 BGP ダンプニングパケットのデバッグ情報を表示します。 • プレフィックス リストが指定されていない場合は、すべての IPv6 BGP 減衰パケットのデバッグ メッセージが表示されます。
ステップ 6	**debug bgp ipv6 unicast** | **multicast**} **updates**[*ipv6-address*] [**prefix-list** *prefix-list-name*] [**in**| **out**] 例： `Device# debug bgp ipv6 unicast updates`	（任意）IPv6 BGP アップデート パケットのデバッグ情報を表示します。 • *ipv6-address* 引数が指定されている場合は、指定したネイバーへの IPv6 BGP アップデートのデバッグメッセージが表示されます。 • **in** キーワードを使用して、インバウンドアップデートのデバッグメッセージだけを表示するようにします。 • **out** キーワードを使用して、アウトバウンドアップデートのデバッグメッセージだけを表示するようにします。

マルチプロトコル BGP for IPv6 を導入するための設定例

例：BGP プロセス、BGP ルータ ID、IPv6 マルチプロトコル BGP ピアの設定

次の例では、IPv6 をグローバルに有効にし、BGP プロセスを設定して、BGP ルータ ID を確立します。また、IPv6 マルチプロトコル BGP ピア 2001:DB8:0:CC00::1 を設定してアクティブ化します。

```
Device> enable
Device# configure terminal
Device(config)# ipv6 unicast-routing
Device(config)# router bgp 65000
Device(config-router)# no bgp default ipv4-unicast
Device(config-router)# bgp router-id 192.168.99.70
Device(config-router)# neighbor 2001:DB8:0:CC00::1 remote-as 64600
Device(config-router)# address-family ipv6 unicast
Device(config-router-af)# neighbor 2001:DB8:0:CC00::1 activate
Device(config-router-af)# end
```

例：リンクローカル アドレスを使用した IPv6 マルチプロトコル BGP ピアの設定

次の例では、ギガビットイーサネットインターフェイス 0/0/0 上で IPv6 マルチプロトコル BGP ピア FE80::XXXX:BFF:FE0E:A471 を設定し、ギガビットイーサネットインターフェイス 0/0/0 の IPv6 ネクストホップ グローバル アドレスを BGP アップデートに含めるために nh6 という名前のルートマップを設定します。IPv6 ネクストホップ リンクローカルアドレスは、nh6 ルートマップ（次の例には記載なし）によって、または **neighbor update-source** コマンド（次の例を参照）で指定したインターフェイスから設定できます。

```
Device> enable
Device# configure terminal
Device(config)# router bgp 65000
Device(config-router)# neighbor 2001:DB8:0000:0000:0000:0000:0000:0111 remote-as 64600
Device(config-router)# neighbor 2001:DB8:0000:0000:0000:0000:0000:0111 update-source
gigabitethernet 0/0/0
Device(config-router)# address-family ipv6
Device(config-router-af)# neighbor 2001:DB8:0000:0000:0000:0000:0000:0111 activate
Device(config-router-af)# neighbor 2001:DB8:0000:0000:0000:0000:0000:0111 route-map nh6
 out
Device(config-router-af)# exit
Device(config-router)# exit
Device(config)# route-map nh6 permit 10
Device(config-route-map)# match ipv6 address prefix-list list1
Device(config-route-map)# set ipv6 next-hop 2001:DB8:5y6::1
Device(config-route-map)# exit
Device(config)# ipv6 prefix-list list1 permit 2001:DB8:2Fy2::/48 le 128
Device(config)# ipv6 prefix-list list1 deny ::/0
```

```
Device(config)# end
```

（注） **neighbor update-source** コマンドでネイバーインターフェイス（*interface-type* 引数）を指定した後に、**set ipv6 next-hop** コマンドでグローバル IPv6 ネクストホップアドレス（*ipv6-address* 引数）だけを指定した場合は、*interface-type* 引数で指定したインターフェイスのリンクローカルアドレスが BGP アップデートのネクストホップとして含まれます。したがって、リンクローカル アドレスを使用する複数の BGP ピアに必要となるのは、BGP アップデートにグローバル IPv6 ネクストホップ アドレスを設定する 1 つのルート マップだけとなります。

例：IPv6 マルチプロトコル BGP ピアグループの設定

次に、group1 という名前の IPv6 マルチプロトコル BGP ピア グループを設定する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# router bgp 65000
Device(config-router)# no bgp default ipv4-unicast
Device(config-router)# neighbor group1 peer-group
Device(config-router)# neighbor group1 remote-as 100
Device(config-router)# neighbor group1 update-source Loopback0
Device(config-router)# neighbor 2001:DB8::1 peer-group group1
Device(config-router)# neighbor 2001:DB8:2:2 peer-group group1
Device(config-router)# address-family ipv6 multicast
Device(config-router-af)# neighbor 2001:DB8::1 activate
Device(config-router-af)# neighbor 2001:DB8:2:2 activate
Device(config-router-af)# exit-address-family
Device(config-router)# end
```

例：IPv6 マルチプロトコル BGP プレフィックスのルートマップの設定

次に、rtp という名前のルートマップを設定して、ネットワーク 2001:DB8::/24 からの IPv6 ユニキャストルートが list1 という名前のプレフィックスリストに一致する場合は、その IPv6 ユニキャストルートを許可する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# router bgp 64900
Device(config-router)# no bgp default ipv4-unicast
Device(config-router)# neighbor 2001:DB8:0:CC00::1 remote-as 64700
Device(config-router)# address-family ipv6 unicast
Device(config-router-af)# neighbor 2001:DB8:0:CC00::1 activate
Device(config-router-af)# neighbor 2001:DB8:0:CC00::1 route-map rtp in
Device(config-router-af)# exit
Device(config)# ipv6 prefix-list cisco seq 10 permit 2001:DB8::/24
Device(config)# route-map rtp permit 10
Device(config-route-map)# match ipv6 address prefix-list list1
Device(config-route-map)# end
```

例：IPv6 マルチプロトコル BGP へのプレフィックスの再配布

次に、ローカル ルータの IPv6 マルチキャスト データベースに BGP ルートを再配布する例を示します。

```
router bgp 64900
 no bgp default ipv4-unicast
address-family ipv6 multicast
 redistribute BGP
```

例：IPv6 マルチプロトコル BGP へのルートのアドバタイズ

次に、ローカルデバイスの IPv6 ユニキャスト データベースに IPv6 ネットワーク 2001:DB8::/24 を挿入する例を示します（BGP は、ネットワークをアドバタイズする前に、ネットワークのルートがローカルデバイスの IPv6 ユニキャスト データベースに存在することを確認します）。

```
Device> enable
Device# configure terminal
Device(config)# router bgp 65000
Device(config-router)# no bgp default ipv4-unicast
Device(config-router)# address-family ipv6 unicast
Device(config-router-af)# network 2001:DB8::/24
Device(config-router-af)# end
```

例：IPv6 ピア間での IPv4 ルートのアドバタイズ

次の例では、IPv6 ネットワークが 2 つの個別 IPv4 ネットワークに接続している場合に、IPv6 ピア間で IPv4 ルートをアドバタイズしています。ピアリングは、IPv4 アドレス ファミリ コンフィギュレーション モードで IPv6 アドレスを使用して設定されています。アドバタイズされたネクスト ホップは到達不能である可能性があるため、rmap という名前のインバウンド ルート マップによってネクスト ホップが設定されます。

```
Device> enable
Device# configure terminal
Device(config)# router bgp 65000
Device(config-router)# neighbor 6peers peer-group
Device(config-router)# neighbor 2001:DB8:1234::2 remote-as 65002
Device(config-router)# address-family ipv4
Device(config-router)# neighbor 6peers activate
Device(config-router)# neighbor 6peers soft-reconfiguration inbound
Device(config-router)# neighbor 2001:DB8:1234::2 peer-group 6peers
Device(config-router)# neighbor 2001:DB8:1234::2 route-map rmap in
Device(config-router)# exit
Device(config)# route-map rmap permit 10
Device(config-route-map)# set ip next-hop 10.21.8.10
Device(config-route-map)# end
```

マルチプロトコル BGP for IPv6 の導入に関するその他の参考資料

標準および RFC

RFC	タイトル
RFC 2545	『*Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing*』
RFC 2858	『*Multiprotocol Extensions for BGP-4*』
RFC 4007	『*IPv6 Scoped Address Architecture*』
RFC 4364	『BGP MPLS/IP Virtual Private Networks (VPNs)』
RFC 4382	『MPLS/BGP Layer 3 Virtual Private Network (VPN) Management Information Base』
RFC 4659	『BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN』
RFC 4724	『*Graceful Restart Mechanism for BGP*』

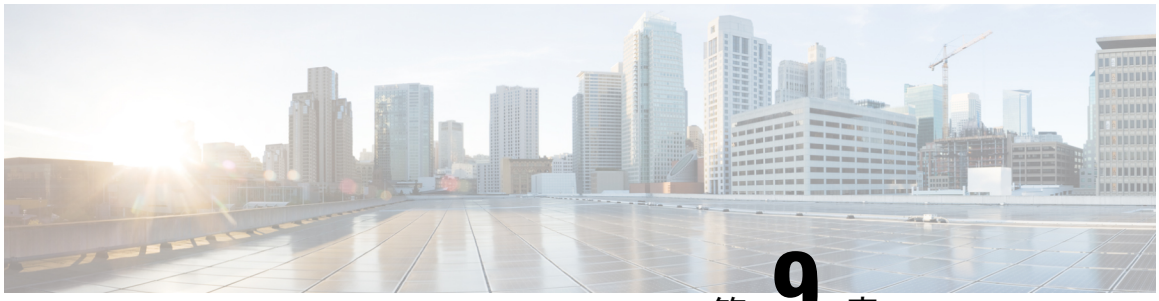
マルチプロトコル BGP for IPv6 の実装の機能情報

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。

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 *21 :* マルチプロトコル *BGP for IPv6* の実装の機能情報

機能名	リリース	機能情報
IPv6 のマルチプロトコル BGP	Cisco IOS XE Everest 16.5.1a	マルチプロトコル BGP for IPv6 拡張では、IPv4 BGP と同じ機能および機能性がサポートされています。



第 9 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 device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devices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 devices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、devices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、devices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deviceは他のエリアに到達する方法を認識しています。

Devicesはレベル 1 隣接を確立して、ローカルエリア内でルーティングを実行します（ステーションルーティング）。Devicesはレベル 2 隣接を確立して、レベル 1 エリア間の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 device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device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。device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deviceインスタンスをレベル 2 deviceとして設定する場合にも **is-type** コマンドを使用します。

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）devicesが、NFS 対応devices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device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device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device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device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、device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device、隣接レベル 1 devices、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devices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devices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device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device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 *22 : 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 devicesでは、ハードウェアやソフトウェに、隣接するノンストップ フォワーディング対応ルータから 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。

機能	デフォルト設定
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、過負ちに設定され、**no set-overload-bit** コマンドが入力されるままになります。
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。その後、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	**configure terminal** 例： `デバイス# 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 isis** [*area tag*] 例： `デバイス(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、値を入力する必要があります。 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 になります。**is-type** 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のレベルを変更できます。
ステップ 3	**net** network-entity-title 例：	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、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できます。

	コマンドまたはアクション	目的
	デバイス(config-router)# net 47.0004.004d.0001.0001.0c11.1111.00	
ステップ 4	**is-type** {**level-1** | **level-1-2** | **level-2-only**} 例： デバイス(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 5	**exit** 例： デバイス(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 6	**interface** *interface-id* 例： デバイス(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 7	**ip router isis** [*area tag*] 例： デバイス(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 8	**ip address** *ip-address-mask* 例： デバイス(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で IP アドレスが必要です。
ステップ 9	**end** 例： デバイス(config)# **end**	特権 EXEC モードに戻ります。
ステップ 10	**show isis** [*area tag*] **database detail** 例：	入力を確認します。

	コマンドまたはアクション	目的
	デバイス# show isis database detail	
ステップ 11	**copy running-config startup-config** 例： デバイス# **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。

IS-IS グローバル 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	**configure terminal** 例： デバイス# **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 isis** 例： デバイス(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 3	**default-information originate** [**route-map** *map-name*] 例： デバイス(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* を入力すると、ルートマップが条件に一致している場合にルーティングロセスによってデフォルトルートが生成されます。
ステップ 4	**ignore-lsp-errors** 例： デバイス(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 を無視するようにルータを設定します。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、**no ignore-lsp-errors** ルータ コンフィギュレーション コマンドを入力します。
ステップ 5	**area-password** *password* 例： デバイス(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。

	コマンドまたはアクション	目的
ステップ 6	**domain-password** *password* 例： `デバイス(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 7	**summary-address** *address mask* [**level-1** | **level-1-2** | **level-2**] 例： `デバイス(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 8	**set-overload-bit** [**on-startup** {*seconds* | **wait-for-bgp**}] 例： `デバイス(config-router)# set-overload-bit on-startup wait-for-bgp`	（任意）ルータに問題がある場合に、他のルータが最短パス優先（SPF）計算でこのルータ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 9	**lsp-refresh-interval** *seconds* 例： `デバイス(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。
ステップ 10	**max-lsp-lifetime** *seconds* 例：	（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

	コマンドまたはアクション	目的
	`デバイス(config-router)# max-lsp-lifetime 1000`	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 11	**lsp-gen-interval** [**level-1** | **level-2**] *lsp-max-wait* [*lsp-initial-wait lsp-second-wait*] 例： `デバイス(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 12	**spf-interval** [**level-1** | **level-2**] *spf-max-wait* [*spf-initial-wait spf-second-wait*] 例： `デバイス(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 13	**prc-interval** *prc-max-wait* [*prc-initial-wait prc-second-wait*] 例： `デバイス(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。 • *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 ～

	コマンドまたはアクション	目的
		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 14	**log-adjacency-changes** [**all**] 例： `デバイス(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 およびリンクステートパケット（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 15	**lsp-mtu** *size* 例： `デバイス(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devicesで LSP MTU サイズを変更する必要があります。
ステップ 16	**partition avoidance** 例： `デバイス(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 17	**end** 例： `デバイス(config)# end`	特権 EXEC モードに戻ります。
ステップ 18	**copy running-config startup-config** 例： `デバイス# 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。

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	**configure terminal** 例： `デバイス# 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： `デバイス(config)# interface gigabitethernet 1/0/1`	設定するインターフェイスを指定して、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 3	**isis metric** *default-metric* [**level-1** | **level-2**] 例： `デバイス(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 およびレベル 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 4	**isis hello-interval** {*seconds* | **minimal**} [**level-1** | **level-2**] 例： `デバイス(config-if)# isis hello-interval minimal`	（任意）スイッチが 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 5	**isis hello-multiplier** *multiplier* [**level-1** | **level-2**] 例： `デバイス(config-if)# isis hello-multiplier 5`	（任意）deviceが隣接装置のダウンを宣言するまでに、ネイバーが損失する IS-IS hello パケット数を指定します。指定できる範囲は 3 ～ 1000 です。デフォルトは 3 です （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。

	コマンドまたはアクション	目的
ステップ 6	**isis csnp-interval** *seconds* [**level-1** | **level-2**] 例： `デバイス(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 7	**isis retransmit-interval** *seconds* 例： `デバイス(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 8	**isis retransmit-throttle-interval** *milliseconds* 例： `デバイス(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。範囲は 0 ～ 65535 です。デフォルト値は、**isis lsp-interval** コマンドにより決定します。
ステップ 9	**isis priority** *value* [**level-1** | **level-2**] 例： `デバイス(config-if)# isis priority 50`	（任意）指定deviceで使用する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 10	**isis circuit-type** {**level-1** | **level-1-2** | **level-2-only**} 例： `デバイス(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。 • **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。

	コマンドまたはアクション	目的
ステップ 11	**isis password** *password* [**level-1** | **level-2**] 例： デバイス(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 12	**end** 例： デバイス(config)# **end**	特権 EXEC モードに戻ります。
ステップ 13	**copy running-config startup-config** 例： デバイス# **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。

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 *23 : IS-IS show* コマンド

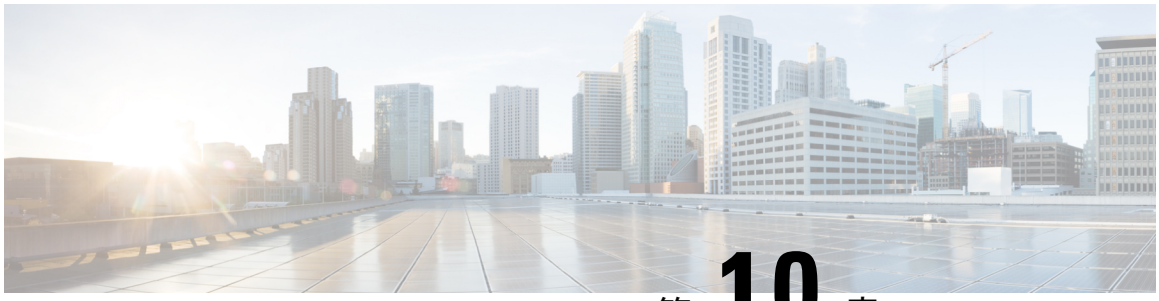
コマンド
show ip route isis
show isis database
show isis routes
show isis spf-log
show isis topology
show route-map

コマンド
trace clns *[接続先（Destination）]*

IS-IS の機能情報

表 *24 : 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 10 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。

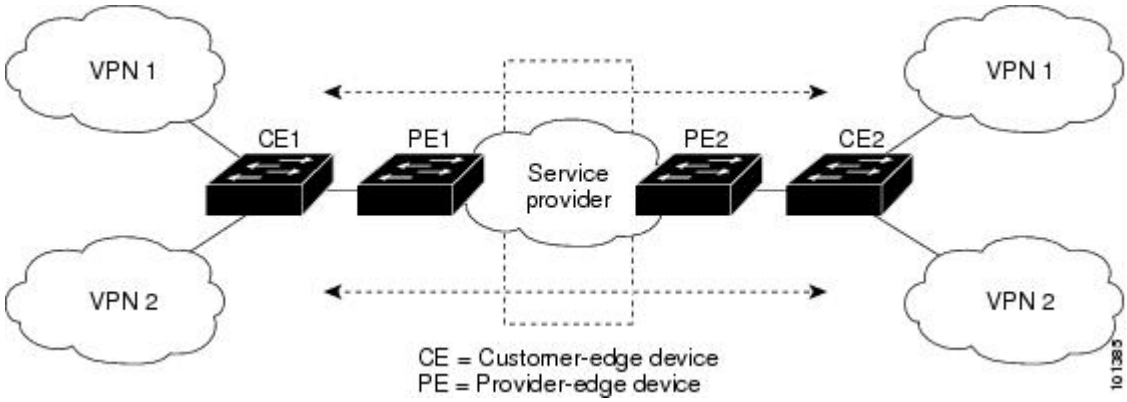
- お客様は、CEデバイスにより、1つまたは複数のプロバイダエッジ（PE）ルータへのデータ リンクを介してサービス プロバイダ ネットワークにアクセスできます。CE デバイスは、サイトのローカル ルートをルータにアドバタイズし、リモート VPN ルートをそこから学習します。スイッチを CE に設定することができます。
- PEルータは、スタティックルーティング、またはBGP、RIPv2、OSPF、EIGRPなどのルーティングプロトコルを使用して、CEデバイスとルーティング情報を交換します。PEは、直接接続している VPN に対する VPN ルートのみを保守する必要があります。そのため、すべてのサービス 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 *7*: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 *25 : 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。

（注） スタック スイッチで VRF 設定を変更した場合は、スタック全体をリロードすることをお勧めします。これは、CEF と VRF コントロール プレーン間の整合性を維持し、マスター スイッチ オーバーの場合に不整合により表示されるエラー メッセージを避けるために不可欠です。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `デバイス# 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： `デバイス(config)# ip routing`	IP ルーティングを有効にします。
ステップ 3	**ip vrf** *vrf-name* 例： `デバイス(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 4	**rd** *route-distinguisher* 例： `デバイス(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 5	**route-target** {**export** | **import** | **both**} *route-target-ext-community* 例： `デバイス(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 6	**import map** *route-map* 例： `デバイス(config-vrf)# import map importmap1`	（任意）VRF にルート マップを対応付けます。

	コマンドまたはアクション	目的
ステップ 7	**interface** *interface-id* 例： `デバイス(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 8	**ip vrf forwarding** *vrf-name* 例： `デバイス(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 9	**end** 例： `デバイス(config)# end`	特権 EXEC モードに戻ります。
ステップ 10	**show ip vrf** [**brief** | **detail** | **interfaces**] [*vrf-name*] 例： `デバイス# 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 11	**copy running-config startup-config** 例： `デバイス# 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 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： `デバイス# 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： `デバイス# 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	**configure terminal** 例： `デバイス# 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	**snmp-server trap authentication vrf** 例： `デバイス(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。
ステップ 3	**snmp-server engineID remote** *host* **vrf** *vpn-instance engine-id string* 例： `デバイス(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。

	コマンドまたはアクション	目的
ステップ 4	**snmp-server host** *host* **vrf** *vpn-instance* **traps** *community* 例： `デバイス(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 5	**snmp-server host** *host* **vrf** *vpn-instance* **informs** *community* 例： `デバイス(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 6	**snmp-server user** *user group* **remote** *host* **vrf** *vpn-instance security model* 例： `デバイス(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 7	**end** 例： `デバイス(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：	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** *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。
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。

	コマンドまたはアクション	目的
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。
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。

	コマンドまたはアクション	目的
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	**configure terminal** 例： `デバイス# 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： `デバイス(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。

	コマンドまたはアクション	目的
ステップ 3	**no switchport** 例： `デバイス(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	**ip vrf forwarding** *vrf-name* 例： `デバイス(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 5	**ip address** *ip-address* 例： `デバイス(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 6	**ip verify unicast reverse-path** 例： `デバイス(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 7	**end** 例： `デバイス(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	**configure terminal** 例： `デバイス# 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	**logging on** 例： `デバイス(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 3	**logging host** *ip-address* **vrf** *vrf-name* 例： `デバイス(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 4	**logging buffered** *logging buffered size* **debugging** 例： `デバイス(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 5	**logging trap debugging** 例： `デバイス(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 6	**logging facility** *facility* 例： `デバイス(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 7	**end** 例： `デバイス(config-if)# end`	特権 EXEC モードに戻ります。

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： `デバイス(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	**configure terminal** 例： `デバイス# 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 ftp source-interface** *interface-type interface-number* 例： `デバイス(config)# ip ftp source-interface gigabitethernet 1/0/2`	FTP 接続の発信元 IP アドレスを指定します。
ステップ 3	**end** 例： `デバイス(config)#end`	特権 EXEC モードに戻ります。
ステップ 4	**configure terminal** 例： `デバイス# 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 5	**ip tftp source-interface** *interface-type interface-number* 例： `デバイス(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 6	**end** 例： `デバイス(config)# end`	特権 EXEC モードに戻ります。

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	**configure terminal** 例： デバイス# **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： デバイス(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 3	**ip vrf** *vrf-name* 例： デバイス(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 4	**rd** *route-distinguisher* 例： デバイス(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 5	**route-target** {**export** | **import** | **both**} *route-target-ext-community* 例： デバイス(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 6	**import map** *route-map* 例： デバイス(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 7	**ip multicast-routing vrf** *vrf-name* **distributed** 例： デバイス(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。

	コマンドまたはアクション	目的
ステップ 8	**interface** *interface-id* 例： `デバイス(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 9	**ip vrf forwarding** *vrf-name* 例： `デバイス(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 10	**ip address** *ip-address* **mask** 例： `デバイス(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 11	**ip pim sparse-dense mode** 例： `デバイス(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。
ステップ 12	**end** 例： `デバイス(config)# end`	特権 EXEC モードに戻ります。
ステップ 13	**show ip vrf** [**brief** | **detail** | **interfaces**] [*vrf-name*] 例： `デバイス# 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 14	**copy running-config startup-config** 例： `デバイス# 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	**configure terminal** 例： `デバイス# 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 ospf** *process-id* **vrf** *vrf-name* 例： `デバイス(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 3	**log-adjacency-changes** 例： `デバイス(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 4	**redistribute bgp** *autonomous-system-number* **subnets** 例： `デバイス(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 5	**network** *network-number* **area** *area-id* 例： `デバイス(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 6	**end** 例： `デバイス(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： `デバイス# show ip ospf 1`	OSPF ネットワークの設定を確認します。

	コマンドまたはアクション	目的
ステップ 8	**copy running-config startup-config** 例： `デバイス# 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： `デバイス# 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： `デバイス(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： `デバイス(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： `デバイス(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： `デバイス(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： `デバイス(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。

	コマンドまたはアクション	目的
ステップ 7	**neighbor** *address* **remote-as** *as-number* 例： `デバイス(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： `デバイス(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： `デバイス(config-router)# end`	特権 EXEC モードに戻ります。
ステップ 10	**show ip bgp** [**ipv4**] [**neighbors**] 例： `デバイス# 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： `デバイス# 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 *26 : Multi-VRF CE* 情報を表示するコマンド

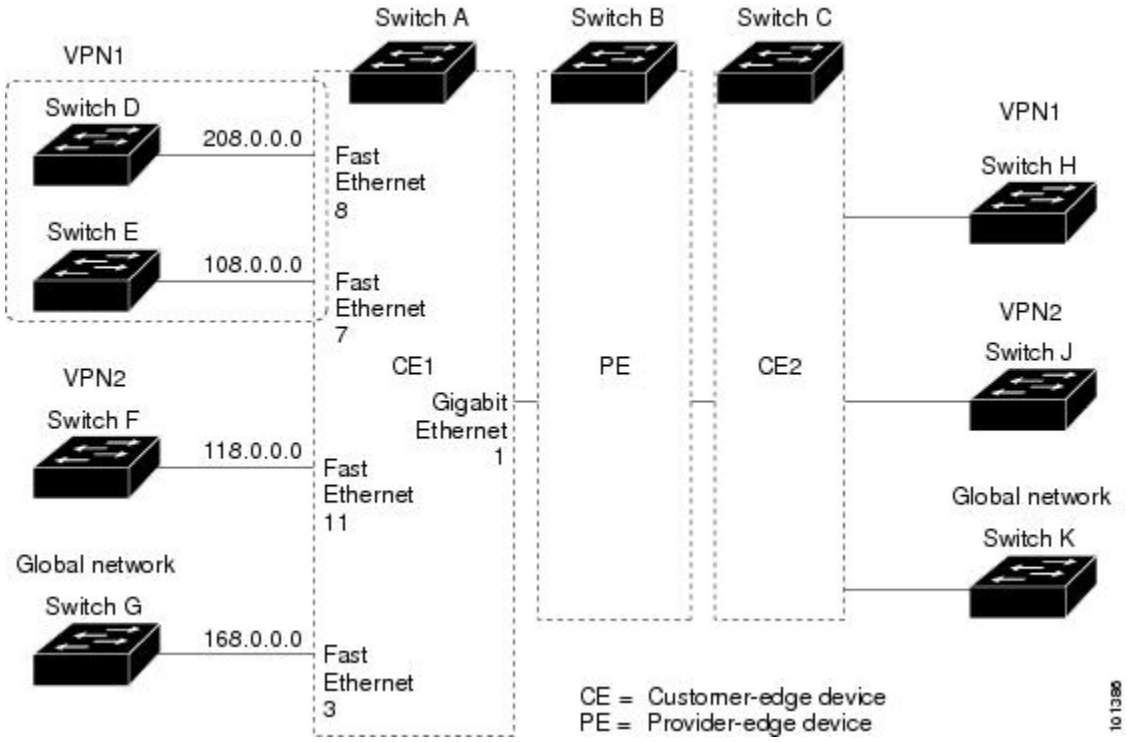
show ip protocols vrf *vrf-name*	VRF に対応付けられたルーティング情報を表示します。
show ip route vrf *vrf-name* [**connected**] [*protocol* [*as-number*]] [**list**] [**mobile**] [**odr**] [**profile**] [**static**] [**summary**] [**supernets-only**]	VRF に対応付けられた IP ルーティン情報を表示します。
show ip vrf [**brief** | **detail** | **interfaces**] [*vrf-name*]	定義された VRF インスタンスに関す示します。

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 *8 : 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。

```
デバイス# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
デバイス(config)# ip routing
デバイス(config)# ip vrf v11
デバイス(config-vrf)# rd 800:1
デバイス(config-vrf)# route-target export 800:1
デバイス(config-vrf)# route-target import 800:1
デバイス(config-vrf)# exit
デバイス(config)# ip vrf v12
デバイス(config-vrf)# rd 800:2
デバイス(config-vrf)# route-target export 800:2
```

```
デバイス(config-vrf)# route-target import 800:2
デバイス(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。

```
デバイス(config)# interface loopback1
デバイス(config-if)# ip vrf forwarding v11
デバイス(config-if)# ip address 8.8.1.8 255.255.255.0
デバイス(config-if)# exit

デバイス(config)# interface loopback2
デバイス(config-if)# ip vrf forwarding v12
デバイス(config-if)# ip address 8.8.2.8 255.255.255.0
デバイス(config-if)# exit

デバイス(config)# interface gigabitethernet1/0/5
デバイス(config-if)# switchport trunk encapsulation dot1q
デバイス(config-if)# switchport mode trunk
デバイス(config-if)# no ip address
デバイス(config-if)# exit
デバイス(config)# interface gigabitethernet1/0/8
デバイス(config-if)# switchport access vlan 208
デバイス(config-if)# no ip address
デバイス(config-if)# exit
デバイス(config)# interface gigabitethernet1/0/11
デバイス(config-if)# switchport trunk encapsulation dot1q
デバイス(config-if)# switchport mode trunk
デバイス(config-if)# no ip address
デバイス(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。

```
デバイス(config)# interface vlan10
デバイス(config-if)# ip vrf forwarding v11
デバイス(config-if)# ip address 38.0.0.8 255.255.255.0
デバイス(config-if)# exit
デバイス(config)# interface vlan20
デバイス(config-if)# ip vrf forwarding v12
デバイス(config-if)# ip address 83.0.0.8 255.255.255.0
デバイス(config-if)# exit
デバイス(config)# interface vlan118
デバイス(config-if)# ip vrf forwarding v12
デバイス(config-if)# ip address 118.0.0.8 255.255.255.0
デバイス(config-if)# exit
デバイス(config)# interface vlan208
デバイス(config-if)# ip vrf forwarding v11
デバイス(config-if)# ip address 208.0.0.8 255.255.255.0
デバイス(config-if)# exit
```

VPN1 と VPN2 で OSPF ルーティングを設定します。

```
デバイス(config)# router ospf 1 vrf vl1
デバイス(config-router)# redistribute bgp 800 subnets
デバイス(config-router)# network 208.0.0.0 0.0.0.255 area 0
デバイス(config-router)# exit
デバイス(config)# router ospf 2 vrf vl2
デバイス(config-router)# redistribute bgp 800 subnets
デバイス(config-router)# network 118.0.0.0 0.0.0.255 area 0
デバイス(config-router)# exit
```

CE/PE ルーティングに BGP を設定します。

```
デバイス(config)# router bgp 800
デバイス(config-router)# address-family ipv4 vrf vl2
デバイス(config-router-af)# redistribute ospf 2 match internal
デバイス(config-router-af)# neighbor 83.0.0.3 remote-as 100
デバイス(config-router-af)# neighbor 83.0.0.3 activate
デバイス(config-router-af)# network 8.8.2.0 mask 255.255.255.0
デバイス(config-router-af)# exit
デバイス(config-router)# address-family ipv4 vrf vl1
デバイス(config-router-af)# redistribute ospf 1 match internal
デバイス(config-router-af)# neighbor 38.0.0.3 remote-as 100
デバイス(config-router-af)# neighbor 38.0.0.3 activate
デバイス(config-router-af)# network 8.8.1.0 mask 255.255.255.0
デバイス(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。

```
デバイス# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
デバイス(config)# ip routing
デバイス(config)# interface gigabitethernet1/0/2
デバイス(config-if)# no switchport
デバイス(config-if)# ip address 208.0.0.20 255.255.255.0
デバイス(config-if)# exit

デバイス(config)# router ospf 101
デバイス(config-router)# network 208.0.0.0 0.0.0.255 area 0
デバイス(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。

```
デバイス# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
デバイス(config)# ip routing
デバイス(config)# interface gigabitethernet1/0/1
デバイス(config-if)# switchport trunk encapsulation dot1q
デバイス(config-if)# switchport mode trunk
デバイス(config-if)# no ip address
デバイス(config-if)# exit
```

```
デバイス(config)# interface vlan118
デバイス(config-if)# ip address 118.0.0.11 255.255.255.0
デバイス(config-if)# exit

デバイス(config)# router ospf 101
デバイス(config-router)# network 118.0.0.0 0.0.0.255 area 0
デバイス(config-router)# end
```

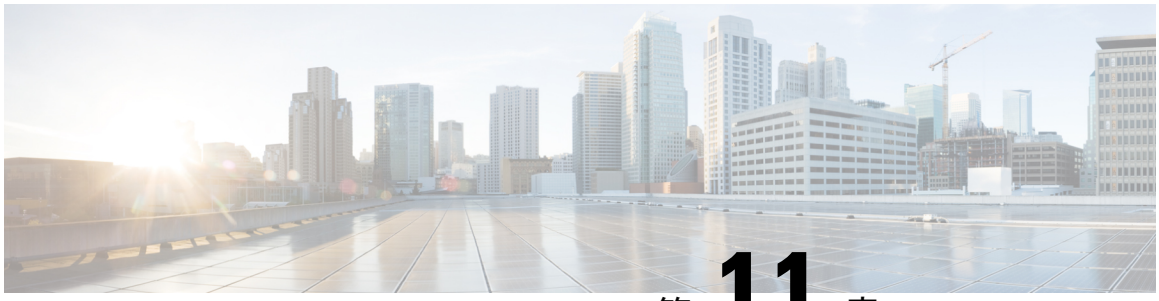
このコマンドをスイッチB（PEルータ）で使用すると、CEデバイス、スイッチAに対する接続だけが設定されます。

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip vrf v1
Router(config-vrf)# rd 100:1
Router(config-vrf)# route-target export 100:1
Router(config-vrf)# route-target import 100:1
Router(config-vrf)# exit

Router(config)# ip vrf v2
Router(config-vrf)# rd 100:2
Router(config-vrf)# route-target export 100:2
Router(config-vrf)# route-target import 100:2
Router(config-vrf)# exit
Router(config)# ip cef
Router(config)# interface Loopback1
Router(config-if)# ip vrf forwarding v1
Router(config-if)# ip address 3.3.1.3 255.255.255.0
Router(config-if)# exit

Router(config)# interface Loopback2
Router(config-if)# ip vrf forwarding v2
Router(config-if)# ip address 3.3.2.3 255.255.255.0
Router(config-if)# exit

Router(config)# interface gigabitethernet1/1/0.10
Router(config-if)# encapsulation dot1q 10
Router(config-if)# ip vrf forwarding v1
Router(config-if)# ip address 38.0.0.3 255.255.255.0
Router(config-if)# exit

Router(config)# interface gigabitethernet1/1/0.20
Router(config-if)# encapsulation dot1q 20
Router(config-if)# ip vrf forwarding v2
Router(config-if)# ip address 83.0.0.3 255.255.255.0
Router(config-if)# exit

Router(config)# router bgp 100
Router(config-router)# address-family ipv4 vrf v2
Router(config-router-af)# neighbor 83.0.0.8 remote-as 800
Router(config-router-af)# neighbor 83.0.0.8 activate
Router(config-router-af)# network 3.3.2.0 mask 255.255.255.0
Router(config-router-af)# exit
Router(config-router)# address-family ipv4 vrf vl
Router(config-router-af)# neighbor 38.0.0.8 remote-as 800
Router(config-router-af)# neighbor 38.0.0.8 activate
Router(config-router-af)# network 3.3.1.0 mask 255.255.255.0
Router(config-router-af)# end
```

マルチ VRF CE の機能情報

表 *27 :* 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 11 章

ユニキャスト リバース パス転送の設定

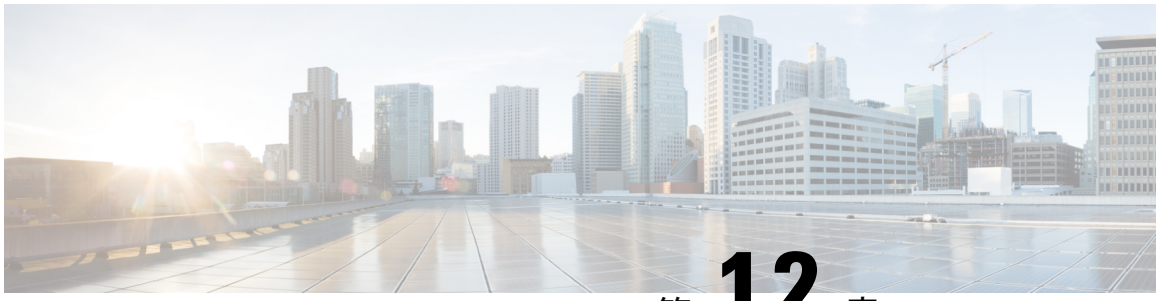
ユニキャスト リバース パス転送の設定

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）

- uRPF は、 でサポートされますNetwork Essentials。
- 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、uRPF を設定しないでください。たとえば、Catalyst 3750-X、Catalyst 3750-E、Catalyst 3750 スイッチです。

ユニキャスト リバース パス転送の設定



第 12 章

プロトコル独立機能

プロトコル独立機能

この項では、IP ルーティング プロトコルに依存しない機能について説明します。これらの機能は、Network Essentials フィーチャ セットが稼働するスイッチ上で使用できます。

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。

デフォルト設定では、すべてのレイヤ3インターフェイスでCEFまたはdCEFがイネーブルです。**no ip route-cache cef** 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と、ソフトウェアが転送するトラフィックに対してCEFが無効になります。このコマンドは、ハードウェア転送パスには影響しません。CEFを無効にして **debug ip packet detail** 特権 EXEC コマンドを使用すると、ソフトウェア転送トラフィックをデバッグするのに便利です。ソフトウェア転送パス用のインターフェイスでCEFを有効にするには、**ip route-cache cef** 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。

⚠

注意 CLIには、インターフェイス上でCEFを無効にする **no ip route-cache cef** インターフェイス コンフィギュレーション コマンドが表示されますが、デバッグ以外の目的でインターフェイス上でCEFまたはdCEFを無効にしないようにしてください。

ディセーブルであるCEFまたはdCEFをグローバルにイネーブルにしたり、ソフトウェア転送トラフィックのインターフェイス上でイネーブルにするには、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	**configure terminal** **例：** `デバイス# 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：** `デバイス(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。

	コマンドまたはアクション	目的
ステップ 3	**ip cef distributed** 例： `デバイス(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： `デバイス(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： `デバイス(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。
ステップ 6	**end** 例： `デバイス(config-if)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show ip cef** 例： `デバイス# 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： `デバイス# 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： `デバイス# 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： `デバイス# 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。

	コマンドまたはアクション	目的
ステップ 11	**show adjacency** 例： `デバイス# 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： `デバイス# 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：	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	**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。

	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 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： `Device(config)# end`	特権 EXEC モードに戻ります。

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	**configure terminal** 例： `デバイス# 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： `デバイス(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 3	**maximum-paths** *maximum* 例： `デバイス(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 4	**end** 例： `デバイス(config-router)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show ip protocols** 例：	*Maximum path* フィールドの設定を確認します。

	コマンドまたはアクション	目的
	デバイス# show ip protocols	
ステップ 6	**copy running-config startup-config** 例： デバイス# 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 *28*：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： `デバイス> 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： `デバイス# 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： `デバイス(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： `デバイス(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 5	**show ip route** 例： デバイス# **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： デバイス# **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device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

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `デバイス# 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 default-network** *network number* 例： `デバイス(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： `デバイス(config)# end`	特権 EXEC モードに戻ります。
ステップ 4	**show ip route** 例： `デバイス# 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： `デバイス# 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： `デバイス# 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： `デバイス(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。

	コマンドまたはアクション	目的
		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： `デバイス(config-route-map)#match as-path 10`	BGP AS パス アクセス リストと照合します。
ステップ 4	**match community-list** *community-list-number* [**exact**] 例： `デバイス(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： `デバイス(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： `デバイス(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： `デバイス(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。

	コマンドまたはアクション	目的
ステップ 8	**match tag** *tag value* [*...tag-value*] 例： `デバイス(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： `デバイス(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： `デバイス(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**]} 例： `デバイス(config-route-map)# match route-type local`	指定された **route-type** と一致させます。 • **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： `デバイス(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： `デバイス(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： `デバイス(config-route-map)#set origin igp`	BGP 送信元コードを設定します。

	コマンドまたはアクション	目的
ステップ 15	**set as-path** {**tag** | **prepend** *as-path-string*} 例： `デバイス(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： `デバイス(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： `デバイス(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： `デバイス(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）。 • *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： `デバイス(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： `デバイス(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。

	コマンドまたはアクション	目的
ステップ 21	**set weight** *number* 例： `デバイス(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： `デバイス(config-route-map)# end`	特権 EXEC モードに戻ります。
ステップ 23	**show route-map** 例： `デバイス# 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： `デバイス# 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： `デバイス# 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： `デバイス(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： `デバイス(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。
ステップ 4	**default-metric** *number* 例： `デバイス(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、OSPF）。
ステップ 5	**default-metric** bandwidth delay reliability loading mtu 例： `デバイス(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： `デバイス(config-router)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show route-map** 例： `デバイス# 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： `デバイス# 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）。

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。

PBR の設定方法

- PBR を使用するには、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。

- VRF と PBR は、スイッチ インターフェイス上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、VRF を有効にはできません。その反対の場合も同じで、VRF 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。
- 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パケットがローカル PBR の影響を受けます。ローカル PBR は、デフォルトで無効に設定されています。

手順の概要

1. **configure terminal**
2. **route-map** *map-tag* [**permit**] [*sequence number*]
3. **match ip address** {*access-list-number* | *access-list-name*} [*access-list-number* |*...access-list-name*]
4. **match length min max**
5. **set ip next-hop** *ip-address* [*...ip-address*]
6. **exit**
7. **interface** *interface-id*
8. **ip policy route-map** *map-tag*
9. **ip route-cache policy**
10. **exit**
11. **ip local policy route-map** *map-tag*
12. **end**
13. **show route-map** [*map-name*]

14. **show ip policy**
15. **show ip local policy**

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： `デバイス# configure terminal`	グローバル設定モードを開始します。
ステップ 2	**route-map** *map-tag* [**permit**] [*sequence number*] 例： `デバイス(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。 •（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。 （注） ハードウェアでの間違ったトラフィック転送の原因となるため、シーケンスに定義された match または set アクションを指定せずに **route-map** *map-tag* [*sequence number*] コマンドを設定しないでください。
ステップ 3	**match ip address** {*access-list-number* | *access-list-name*} [*access-list-number* |*...access-list-name*] 例： `デバイス(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 4	**match length min max** 例：	パケット長と照合します。

	コマンドまたはアクション	目的
	デバイス(config-route-map)# match length 64 1500	
ステップ 5	**set ip next-hop** *ip-address* [*...ip-address*] 例： デバイス(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 6	**exit** 例： デバイス(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 7	**interface** *interface-id* 例： デバイス(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 8	**ip policy route-map** *map-tag* 例： デバイス(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 9	**ip route-cache policy** 例： デバイス(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。
ステップ 10	**exit** 例： デバイス(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 local policy route-map** *map-tag* 例： デバイス(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 12	**end** 例： デバイス(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 route-map** [*map-name*] 例： `デバイス# 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 14	**show ip policy** 例： `デバイス# 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 15	**show ip local policy** 例： `デバイス# 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：	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
	デバイス# configure terminal	
ステップ 2	**router** { **rip** | **ospf** | **eigrp**} 例： デバイス(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： デバイス(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： デバイス(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： デバイス(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： デバイス(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： デバイス(config-router)# end	特権 EXEC モードに戻ります。
ステップ 8	**copy running-config startup-config** 例： デバイス# 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	**configure terminal** 例： `デバイス# 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** | **eigrp**} 例： `デバイス(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	**distribute-list** {*access-list-number* | *access-list-name*} **out** [*interface-name* | *routing process* | *autonomous-system-number*] 例： `デバイス(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。
ステップ 4	**distribute-list** {*access-list-number* | *access-list-name*} **in** [*type-number*] 例： `デバイス(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 5	**end** 例： `デバイス(config-router)# end`	特権 EXEC モードに戻ります。
ステップ 6	**copy running-config startup-config** 例： `デバイス# 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	**configure terminal** **例**： `デバイス# 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**： `デバイス(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	**distance** *weight* {*ip-address* {*ip-address mask*}} [*ip access list*] **例**： `デバイス(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 4	**end** **例**： `デバイス(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 5	**show ip protocols** 例： `デバイス# 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 6	**copy running-config startup-config** 例： `デバイス# 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。

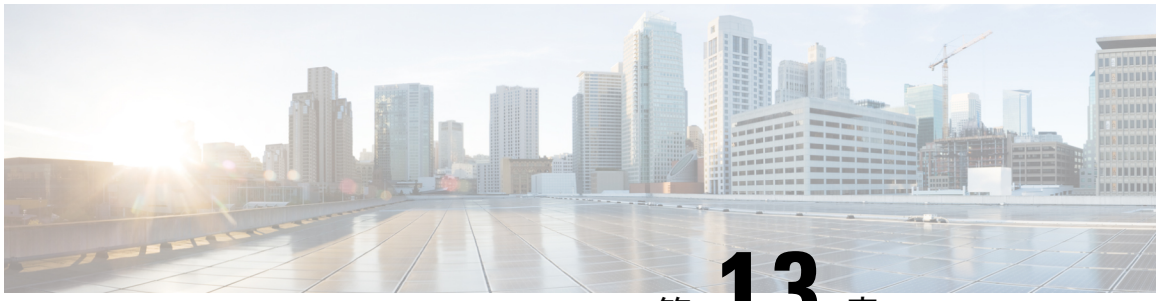
認証キーの設定方法

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： `デバイス# configure terminal`	グローバル設定モードを開始します。
ステップ 2	**key chain** *name-of-chain* 例：	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。

	コマンドまたはアクション	目的
	デバイス(config)# key chain key10	
ステップ 3	**key** *number* 例： デバイス(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： デバイス(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： デバイス(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*} 例： デバイス(config-keychain)# 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： デバイス(config-keychain)# end	特権 EXEC モードに戻ります。
ステップ 8	**show key chain** 例： デバイス# 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： デバイス# 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 13 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。

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： デバイス> **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： `デバイス# 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： `デバイス(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。 • 番号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： `デバイス(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 に割り当てる名前です。
ステップ 5	**ip address***ip-address subnet-mask* 例： `デバイス(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 には、インターフェイスの IP アドレスを指定します。 • • subnet-mask には、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： `デバイス(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 には、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 には、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 引数には、ポート、コネクタ、またはインターフェイス カード番号を指定します。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： `デバイス(config-if)# tunnel destination 10.5.5.5`	トンネルの宛先を指定します。 • hostname には、ホストの宛先の名前を指定します。

	コマンドまたはアクション	目的
		• ip-address には、ホストの宛先の IP アドレスを指定します。
ステップ 8	**tunnel vrf***vrf-name* 例： デバイス(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 *29 :* 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

関連項目	マニュアル タイトル
トンネル	『Cisco IOS Interface Configuration Guide, Release 12.2』

表 *30 :* 標準

標準	タイトル
この機能によってサポートされる新しい規格または変更された規格はありません。またこの機能による既存規格のサポートに変更はありません。	--

表 *31 : RFC*

RFC	タイトル
この機能によりサポートされた新規 RFC または改訂 RFC はありません。またこの機能による既存 RFC のサポートに変更はありません。	--

表 *32 :* 関連 *DoTechnical Assistancecuments*

説明	リンク
シスコのテクニカル サポートおよびドキュメンテーション Web サイトでは、製品、テクノロジー、ソリューション、テクニカル ティップス、ツールへのリンクなど、技術的なコンテンツを検索可能な形で大量に提供しています。Cisco.com に登録済みのユーザーは、このページから詳細情報にアクセスできます。	https://www.cisco.com/c/ja_jp/support/index.html

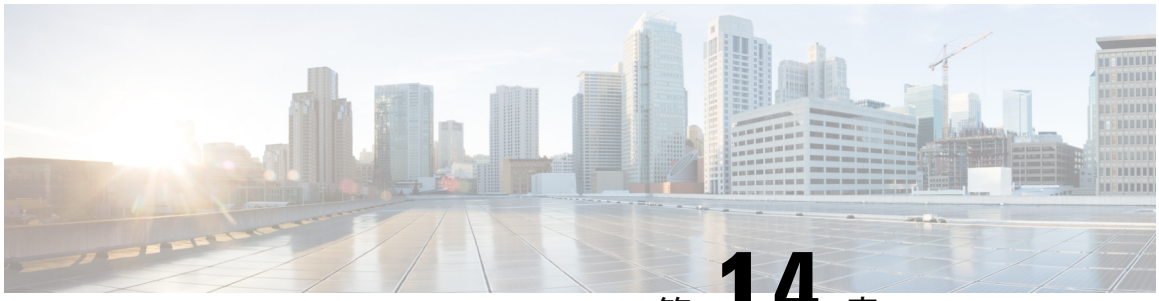
Generic Routing Encapsulation（GRE）トンネル IP 送信元および宛先 VRF メンバーシップの機能履歴

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。

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 : 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。



第 14 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。

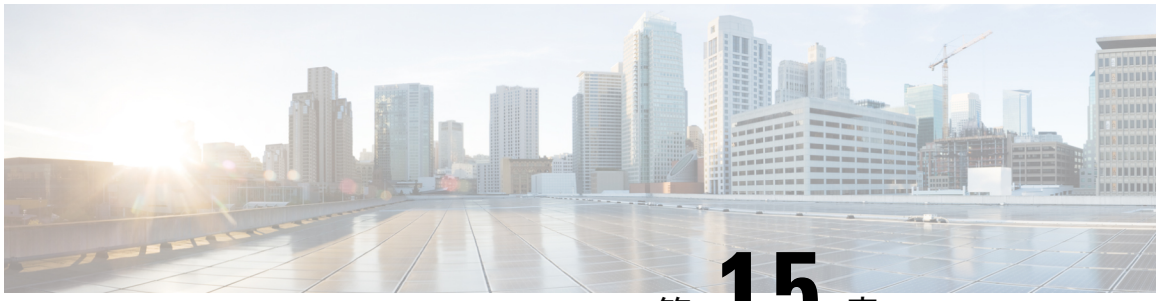
```
Device> enable
Device# show ipv6 ospf interface

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 *34 : 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 15 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 *35 : 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。

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。