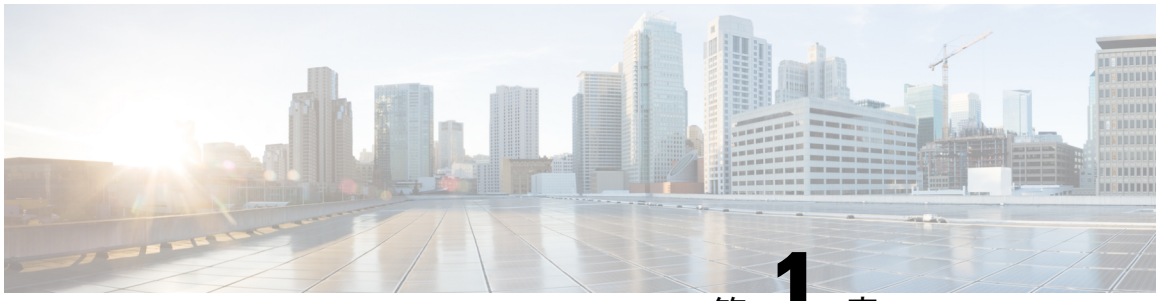
Cisco IOS XE Bengaluru 17.4.x（Catalyst 9200 スイッチ）IP ルーティング コンフィギュレーション ガイド

初版：2020 年 11 月 30 日

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

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



第 1 章

双方向フォワーディング検出の設定

双方向フォワーディング検出

このマニュアルでは、双方向フォワーディング検出（BFD）プロトコルをイネーブル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回数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、さまざまなルーティングプロトコルのhelloメカニズムで、変動速度ではなく一定速度で転送パスの障害を検出できるため、ネットワークプロファイリングおよびプランニングが容易になります。また、再コンバージェンス時間の整合性が保たれ、予測可能になります。

双方向フォワーディング検出の前提条件

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

双方向フォワーディング検出の制約事項

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

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

双方向フォワーディング検出について

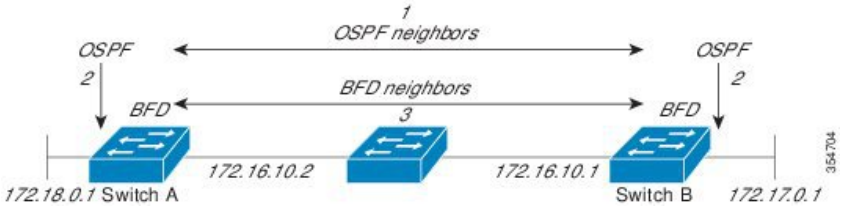
BFD の動作

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

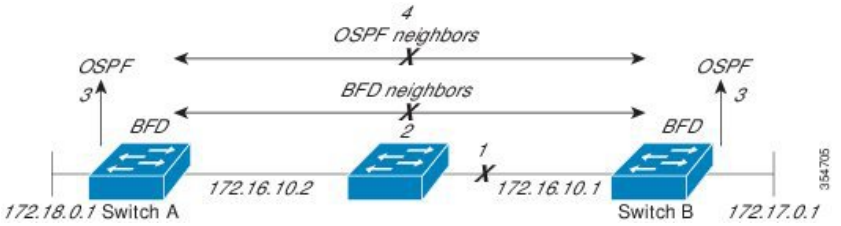
BFD はインターフェイス レベルおよびルーティング プロトコル レベルでイネーブルにする検出プロトコルです。シスコでは、BFD 非同期モードをサポートしています。BFD 非同期モードは、デバイス間の BFD ネイバー セッションをアクティブにして維持するための、2 台のシステム間の BFD 制御パケットの送信に依存します。したがって、BFD セッションを作成するには、両方のシステム（または BFD ピア）で BFD を設定する必要があります。BFD が適切なルーティングプロトコルに対してインターフェイスおよびデバイスレベルで有効になると、BFD セッションが作成されます。BFD タイマーがネ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

ネイバー関係

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



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



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

次のとき、OSPF では、BFD を使用して登録が行われます。

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

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

BFD の障害検出

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

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

BFD バージョンの相互運用性

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

BFD バージョンの検出の例については、エコーモードがデフォルトで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 セッションの制限

作成できる BFD セッションの最大数は 128 です。

非ブロードキャスト メディア インターフェイスに対する BFD サポート

Cisco IOS XE Denali 16.3.1 以降、BFD 機能は、ルーティングされた 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）が有効になっているデバイスのルーティングフラップを抑制するのに役立ち、そのためネットワークの不安定さが減少します。

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

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

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。

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。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。

ルーティング プロトコルの減少したタイマー メカニズムで BFD を実装すると、いくつかの利点があります。

- 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、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 IS-IS、および OSPF タイマーよりも CPU の負荷を軽くすることができます。

双方向フォワーディング検出の設定方法

インターフェイスでの BFD セッション パラメータの設定

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

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	次のいずれかの手順を実行します。 • **ip address** *ipv4-address mask*	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

	• **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`	

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 をイネーブルにします。 BFD interval 設定は、それを設定したサブインターフェイスが削除されたときに削除されます。 BFD interval 設定は次のような場合には削除されません。 • IPv4 アドレスがインターフェイスから削除された場合 • IPv6 アドレスがインターフェイスから削除された場合 • IPv6 がインターフェイスで無効にされた場合 • インターフェイスがシャットダウンされた場合 • インターフェイスで IPv4 CEF がグローバルまたはローカルで無効にされた場合 • インターフェイスで IPv6 CEF がグローバルまたはローカルで無効にされた場合
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

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

	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 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。

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

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

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

手順

	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： `Device(config-if)#ip router isis tag1`	インターフェイスで IPv4 ルーティングのサポートをイネーブルにします。

	コマンドまたはアクション	目的
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、OSPF がルーティングしているすべてのインターフェイスに対して BFD を有効にできます。インターフェイス コンフィギュレーション モードで **ip ospf bfd** [**disable**] コマンドを使用して、個々のインターフェイスで BFD サポートを無効にできます。

- インターフェイス コンフィギュレーション モードで **ip ospf bfd** コマンドを使用すると、OSPF がルーティングしているインターフェイスのサブセットに対して BFD を有効にできます。

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

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

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

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

始める前に

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router ospf** *process-id* 例： `Device(config)#router ospf 4`	OSPF プロセスを指定し、ルータ コンフィギュレーション モードを開始します。
ステップ 4	**bfd all-interfaces** 例： `Device(config-router)#bfd all-interfaces`	OSPF ルーティング プロセスに関連付けられたすべてのインターフェイスで、BFD をグローバルにイネーブルにします。
ステップ 5	**exit** 例：	（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

	コマンドまたはアクション	目的
	`Device(config-router)#exit`	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 をディセーブルにする場合にだけ、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 をディセーブルにする場合にだけ、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 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 *BFD over IPv4* に対する *OSPF* サポートの設定

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

手順

	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 をイネーブルまたはディセーブルにします。 （注） ルータ コンフィギュレーション モードで **bfd all-interfaces** コマンドを使用して OSPF が関連付けられたすべてのインターフェイスで BFD を有効にした場合にのみ、**disable** キーワードを使用します。
ステップ 5	**end** 例： `Device(config-if)#end`	インターフェイス コンフィギュレーション モードを終了して、デバイスが特権 EXEC モードに戻ります。
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。

	コマンドまたはアクション	目的
		（注） ハードウェア オフロードされた BFD セッションが、50 ms の倍数でない Tx および Rx 間隔で設定されると、ハードウェア間隔が変更されます。ただし、**show bfd neighbors details** コマンドの出力には、変更された間隔ではなく、設定された間隔値のみが表示されます。
ステップ 7	**show ip ospf** 例： `Device#show ip ospf`	（任意）OSPF に対して BFD サポートが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。

HSRP に対する BFD サポートの設定

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

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

始める前に

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。

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

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

スタティック ルーティングに対する BFD サポートの設定

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *type number* 例： `Device(config)#interface serial 2/0`	インターフェイスを設定し、インターフェイス コンフィギュレーション モードを開始します。

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

	コマンドまたはアクション	目的
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をイネーブルにします。 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 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*]	スタティック ルートの BFD ネイバーを指定します。

	コマンドまたはアクション	目的
	例： `Device(config)#ip route 10.0.0.0 255.0.0.0`	
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 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 モードがイネーブルになっていますが、方向ごとに個別に実行できるように、ディセーブル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、パケット内遅延が向上する可能性があり、それによってBFDバージョン0をBFDセッションのBFD制御パケットで使用する場合に、障害検出時間を短縮できます。

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

前提条件

- BFD は、参加しているすべてのデバイスで実行されている必要があります。

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

機能制限

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

非対称性のない BFD エコー モードのディセーブル化

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

各 BFD デバイスに対してこの手順を繰り返します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**no bfd echo** 例： `Device(config)#no bfd echo`	BFD エコー モードをディセーブルにします。 • **no** 形式を使用すると、BFD エコーモードを無効にできます。
ステップ 4	**end** 例： `Device(config)#end`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

BFD テンプレートの作成と設定

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

（注） 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

	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

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**show bfd neighbors** [**details**] 例： `Device#show bfd neighbors details`	（任意）BFD 隣接関係データベースを表示します。 • **details** キーワードを指定すると、すべての BFD プロトコル パラメータとネイバーごとにタイマーが表示されます。
ステップ 3	**debug bfd** [**packet** | **event**] 例： `Device#debug bfd packet`	（任意）BFD パケットのデバッグ情報を表示します。

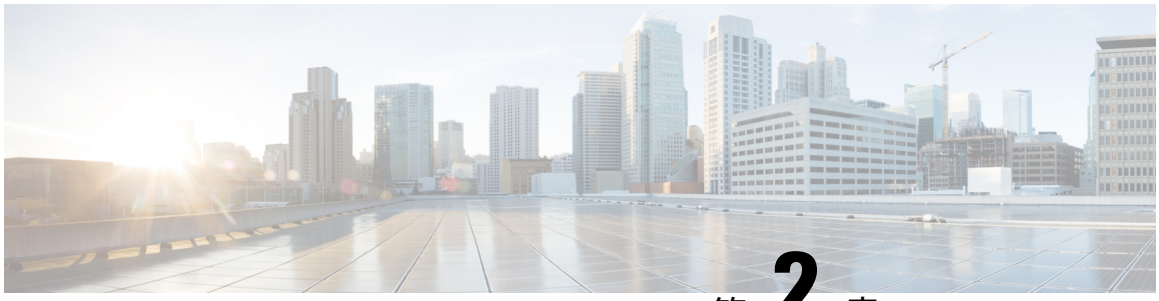
双方向フォワーディング検出の設定例

ここでは、双方向フォワーディング検出の設定例を示します。

双方向フォワーディング検出に関する機能情報

表 *1 :* 双方向フォワーディング検出に関する機能情報

リリース	機能情報
Cisco IOS XE Fuji 16.9.2	この機能が導入されました



第 2 章

BFD-EIGRP サポートの設定

BFD-EIGRP サポート

BFD-EIGRP サポート機能により、Enhanced Interior Gateway Routing Protocol（EIGRP）を Bidirectional Forwarding Detection（BFD）に登録し、BFD からすべての転送パス検出エラーメッセージを受信するように、BFD で EIGRP を設定できます。

BFD-EIGRP サポートの前提条件

- Enhanced Interior Gateway Routing Protocol（EIGRP）は、関連するすべての参加ルータで実行する必要があります。
- Bidirectional Forwarding Detection（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** コマンドを使用して 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-EIGRP サポートに関する情報

BFD-EIGRP サポートの概要

BFD-EIGRP サポート機能により、ルーティングインターフェイスで Enhanced Interior Gateway Routing Protocol（EIGRP）を Bidirectional Forwarding Detection（BFD）セッションに登録し、BFD から転送パス検出エラーメッセージを受信するように、EIGRP 用の BFD 機能を設定できます。

任意のインターフェイスで BFD を有効にするには、**bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier* コマンドを使用します。EIGRP ルーティングが有効になっているすべてのインターフェイスに対して BFD を有効にするには、ルータ コンフィギュレーション モードで **bfd all-interfaces** コマンドを使用します。EIGRP ルーティングが有効になっているイ

ンターフェイスのサブセットに対して BFD を有効にするには、ルータ コンフィギュレーション モードで **bfd interface** *type number* コマンドを使用します。

BFD-EIGRP サポートの設定方法

BFD-EIGRP サポートの設定方法

BFD-EIGRP サポートの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router eigrp** *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 FastEthernet 6/0`
		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。

	コマンドまたはアクション	目的
ステップ 6	**show bfd neighbors** [**details**] 例： `Device#show bfd neighbors details`	（任意）BFD ネイバーがアクティブで、BFD が登録したルーティング プロトコルが表示されることを確認します。
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。

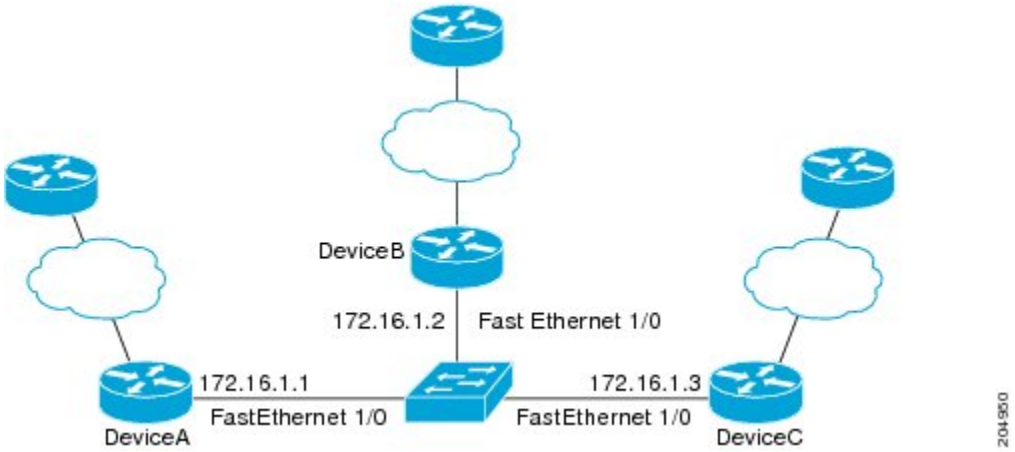
BFD-EIGRP サポートの設定例

例：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

次の例では、EIGRP ネットワークにデバイス A、デバイス B およびデバイス C が含まれています。デバイス A のファスト イーサネット インターフェイス 1/0 がデバイス B のファスト イーサネット インターフェイス 1/0 と同じネットワークに接続されています。デバイス B のファスト イーサネット 1/0 が デバイス C のファスト イーサネット インターフェイス 1/0 と同じネットワークに接続されています。

デバイス A とデバイス B はエコー モードをサポートする BFD バージョン 1 を実行しており、デバイス C はエコー モードをサポートしない BFD バージョン 0 を実行しています。エコー モードはデバイス A とデバイス B の転送パスで動作するため、デバイス C とその BFD ネイバーの間の BFD セッションは非対称のエコー モードで実行されます。BFD セッションおよび障害検出のため、エコー パケットは同じパスで返されます。また、BFD ネイバー デバイス C は BFD バージョン 0 を実行し、BFD セッションおよび障害検出のために BFD 制御パケットを使用します。

下の図に、複数のデバイスがある大規模な EIGRP ネットワークを示します。その中の 3 台は、ルーティング プロトコルとして EIGRP を実行している BFD ネイバーです。



この例は、グローバル コンフィギュレーション モードから開始し、BFD の設定を示します。

デバイス A の設定

```
interface Fast Ethernet0/0
no shutdown
ip address 10.4.9.14 255.255.255.0
duplex auto
speed auto
!
interface Fast Ethernet1/0
ip address 172.16.1.1 255.255.255.0
bfd interval 50 min_rx 50 multiplier 3
no shutdown
duplex auto
speed auto
!
router eigrp 11
network 172.16.0.0
bfd all-interfaces
auto-summary
!
ip default-gateway 10.4.9.1
ip default-network 0.0.0.0
ip route 0.0.0.0 0.0.0.0 10.4.9.1
ip route 172.16.1.129 255.255.255.255 10.4.9.1
!
no ip http server
!
logging alarm informational
!
control-plane
!
line con 0
exec-timeout 30 0
stopbits 1
line aux 0
stopbits 1
line vty 0 4
login
!
!
end
```

デバイス B の設定

```
!
interface Fast Ethernet0/0
no shutdown
ip address 10.4.9.34 255.255.255.0
duplex auto
speed auto
!
interface Fast Ethernet1/0
ip address 172.16.1.2 255.255.255.0
bfd interval 50 min_rx 50 multiplier 3
no shtdown
duplex auto
speed auto
!
router eigrp 11
network 172.16.0.0
bfd all-interfaces
auto-summary
!
ip default-gateway 10.4.9.1
ip default-network 0.0.0.0
ip route 0.0.0.0 0.0.0.0 10.4.9.1
ip route 172.16.1.129 255.255.255.255 10.4.9.1
!
no ip http server
!
logging alarm informational
!
control-plane
!
line con 0
exec-timeout 30 0
stopbits 1
line aux 0
stopbits 1
line vty 0 4
login
!
!
end
```

デバイス C の設定

```
!
!
interface Fast Ethernet0/0
no shutdown
ip address 10.4.9.34 255.255.255.0
duplex auto
speed auto
!
interface Fast Ethernet1/0
ip address 172.16.1.2 255.255.255.0
bfd interval 50 min_rx 50 multiplier 3
no shutdown
duplex auto
speed auto
!
router eigrp 11
network 172.16.0.0
```

```
bfd all-interfaces
auto-summary
!
ip default-gateway 10.4.9.1
ip default-network 0.0.0.0
ip route 0.0.0.0 0.0.0.0 10.4.9.1
ip route 172.16.1.129 255.255.255.255 10.4.9.1
!
no ip http server
!
logging alarm informational
!
control-plane
!
line con 0
exec-timeout 30 0
stopbits 1
line aux 0
stopbits 1
line vty 0 4
login
!
!
end
```

デバイス A からの **show bfd neighbors details** コマンドの出力で、3 台のすべてのデバイス間に BFD セッションが作成され、EIGRP が BFD サポートに登録されることを確認できます。出力の最初のグループは、IP アドレスが 172.16.1.3 のデバイス C が BFD バージョン 0 を実行しているため、エコー モードを使用しないことを示します。出力の 2 番目のグループは、IP アドレスが 172.16.1.2 のデバイス B が BFD バージョン 1 を実行していて、50 ミリ秒の BFD interval パラメータが使用されていることを示します。この出力では、対応するコマンド出力が太字で表示されています。

```
DeviceA# show bfd neighbors details

OurAddr
       NeighAddr
      LD/RD  RH/RS    Holdown(mult)  State     Int
172.16.1.1    172.16.1.3
     5/3    1(RH)    150 (3 )       Up   Fa1/0
Session state is UP and not using echo function.
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 50000, MinRxInt: 50000, Multiplier: 3
Received MinRxInt: 50000, Received Multiplier: 3
Holdown (hits): 150(0), Hello (hits): 50(1364284)
Rx Count: 1351813, Rx Interval (ms) min/max/avg: 28/64/49 last: 4 ms ago
Tx Count: 1364289, Tx Interval (ms) min/max/avg: 40/68/49 last: 32 ms ago
Registered protocols: EIGRP
Uptime: 18:42:45
Last packet: Version: 0
            - Diagnostic: 0
             I Hear You bit: 1     - Demand bit: 0
             Poll bit: 0           - Final bit: 0
             Multiplier: 3         - Length: 24
             My Discr.: 3          - Your Discr.: 5
             Min tx interval: 50000    - Min rx interval: 50000
             Min Echo interval: 0
OurAddr       NeighAddr
     LD/RD  RH/RS   Holdown(mult)  State     Int
172.16.1.1    172.16.1.2
```

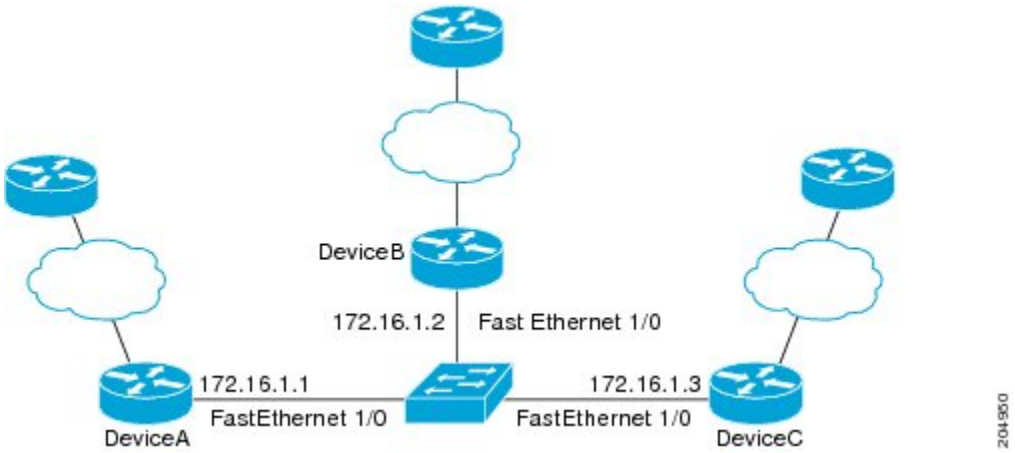
```
    6/1    Up        0    (3 )   Up        Fa1/0
Session state is UP and using echo function with 50 ms interval.
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 1000000, MinRxInt: 1000000, Multiplier: 3
Received MinRxInt: 1000000, Received Multiplier: 3
Holdown (hits): 3000(0), Hello (hits): 1000(317)
Rx Count: 305, Rx Interval (ms) min/max/avg: 1/1016/887 last: 448 ms ago
Tx Count: 319, Tx Interval (ms) min/max/avg: 1/1008/880 last: 532 ms ago
Registered protocols: EIGRP
Uptime: 00:04:30
Last packet: Version: 1

         - Diagnostic: 0
             State bit: Up         - Demand bit: 0
             Poll bit: 0           - Final bit: 0
             Multiplier: 3         - Length: 24
             My Discr.: 1          - Your Discr.: 6
             Min tx interval: 1000000    - Min rx interval: 1000000
             Min Echo interval: 50000
```

デバイス B の **show bfd neighbors details** コマンドによる出力で、BFD セッションが作成され、EIGRP が BFD サポートに対して登録されていることを確認できます。前述のように、デバイス A は BFD バージョン 1 を実行するため、エコー モードを実行しており、デバイス C は BFD バージョン 0 を実行するため、エコーモードを実行しません。この出力では、対応するコマンド出力が太字で表示されています。

```
DeviceB# show bfd neighbors details

OurAddr       NeighAddr
     LD/RD  RH/RS   Holdown(mult)  State     Int
172.16.1.2    172.16.1.1
    1/6    Up        0    (3 )   Up        Fa1/0
Session state is UP and using echo function with 50 ms interval.
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 1000000, MinRxInt: 1000000, Multiplier: 3
Received MinRxInt: 1000000, Received Multiplier: 3
Holdown (hits): 3000(0), Hello (hits): 1000(337)
Rx Count: 341, Rx Interval (ms) min/max/avg: 1/1008/882 last: 364 ms ago
Tx Count: 339, Tx Interval (ms) min/max/avg: 1/1016/886 last: 632 ms ago
Registered protocols: EIGRP
Uptime: 00:05:00
Last packet: Version: 1
           - Diagnostic: 0
             State bit: Up         - Demand bit: 0
             Poll bit: 0           - Final bit: 0
             Multiplier: 3         - Length: 24
             My Discr.: 6          - Your Discr.: 1
             Min tx interval: 1000000    - Min rx interval: 1000000
             Min Echo interval: 50000
OurAddr       NeighAddr

 LD/RD  RH/RS   Holdown(mult)  State     Int
172.16.1.2    172.16.1.3
     3/6    1(RH)     118  (3 )   Up        Fa1/0
Session state is UP and not using echo function.
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 50000, MinRxInt: 50000, Multiplier: 3
Received MinRxInt: 50000, Received Multiplier: 3
Holdown (hits): 150(0), Hello (hits): 50(5735)
```

```
Rx Count: 5731, Rx Interval (ms) min/max/avg: 32/72/49 last: 32 ms ago
Tx Count: 5740, Tx Interval (ms) min/max/avg: 40/64/50 last: 44 ms ago
Registered protocols: EIGRP
Uptime: 00:04:45
Last packet: Version: 0
            - Diagnostic: 0
             I Hear You bit: 1     - Demand bit: 0
             Poll bit: 0           - Final bit: 0
             Multiplier: 3         - Length: 24
             My Discr.: 6          - Your Discr.: 3
             Min tx interval: 50000    - Min rx interval: 50000
             Min Echo interval: 0
```

下の図は、デバイス B のファスト イーサネット インターフェイス 1/0 に障害が発生したことを示しています。デバイス B でファスト イーサネット インターフェイス 1/0 をシャット ダウンした場合、デバイス A とデバイス B の対応する BFD セッションの BFD 統計情報が少なくなります。



デバイス B のファスト イーサネット インターフェイス 1/0 に障害が発生すると、BFD はデバイス A またはデバイス C の BFD ネイバーとしてデバイス B を検出しなくなります。この例では、デバイス B でファスト イーサネット インターフェイス 1/0 が管理的上の理由でシャットダウンされています。

デバイス A での **show bfd neighbors** コマンドによる次の出力では、EIGRP ネットワークのデバイス A の唯一の BFD ネイバーが表示されます。この出力では、対応するコマンド出力が太字で表示されています。

```
DeviceA# show bfd neighbors
OurAddr       NeighAddr

    LD/RD  RH/RS   Holdown(mult)  State     Int
172.16.1.1    172.16.1.3

    5/3    1(RH)     134  (3 )   Up        Fa1/0
```

デバイス C での **show bfd neighbors** コマンドによる次の出力でも、EIGRP ネットワークのデバイス C の唯一の BFD ネイバーが表示されます。この出力では、対応するコマンド出力が太字で表示されています。

```
DeviceC# show bfd neighbors
```

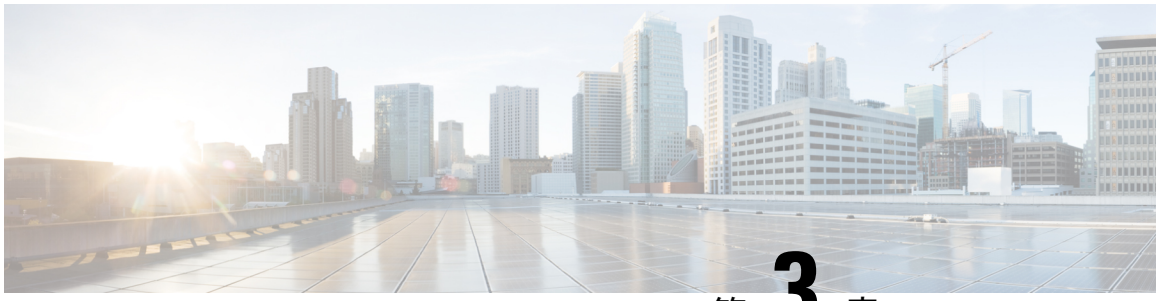
```
OurAddr        NeighAddr

   LD/RD RH   Holdown(mult)  State      Int
172.16.1.3     172.16.1.1

    3/5  1    114   (3 )       Up         Fa1/0
```

BFD-EIGRP サポートの機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

表 *2 : BFD-EIGRP* サポートの機能情報

機能名	リリース	機能情報
BFD-EIGRP サポート	Cisco IOS XE Everest 16.6.2	BFD-EIGRP サポート機能により、Enhanced Interior Gateway Routing Protocol（EIGRP）を Bidirectional Forwarding Detection（BFD）に登録し、BFD からすべての転送パス検出エラーメッセージを受信するように、BFD で EIGRP を設定できます。 この機能は、Cisco IOS XE Everest 16.6.2 で、Cisco Catalyst 9400 シリーズ スイッチに実装されました。



第 3 章

EIGRP IPv6 に対する BFD サポートの設定

EIGRP IPv6 に対する BFD サポートの前提条件

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

EIGRP IPv6 に対する BFD サポートに関する制約事項

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

EIGRP IPv6 に対する BFD サポートに関する情報

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

EIGRP IPv6 に対する BFD サポートの設定方法

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

すべてのインターフェイスでの BFD サポートの設定

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** **例：** `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ipv6 unicast-routing** **例：** `Device(config)# ipv6 unicast-routing`	IPv6 ユニキャスト データグラムの転送をイネーブルにします。
ステップ 4	**interface***type number* **例：** `Device(config)# interface ethernet0/0`	インターフェイスのタイプと番号を指定し、インターフェイス コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 5	**ipv6 address** *ipv6-address/prefix-length* 例： `Device(config-if)# ipv6 address 2001:DB8:A:B::1/64`	IPv6 アドレスを設定します。
ステップ 6	**bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier* 例： `Device(config-if)# bfd interval 50 min_rx 50 multiplier 3`	インターフェイスのベースライン BFD セッションパラメータを設定します。
ステップ 7	**exit** 例： `Device(config-if)# exit`	インターフェイス コンフィギュレーションモードを終了し、グローバル コンフィギュレーションモードに戻ります。
ステップ 8	**router eigrp** *virtual-name* 例： `Device(config)# router eigrp name`	EIGRP ルーティングプロセスを指定し、ルータ コンフィギュレーション モードを開始します。
ステップ 9	**address-family ipv6 autonomous-system***as-number* 例： `Device(config-router)# address-family ipv6 autonomous-system 3`	IPv6 のアドレス ファミリ コンフィギュレーション モードを開始して、EIGRP ルーティングインスタンスを設定します。
ステップ 10	**eigrp router-id***ip-address* 例： `Device(config-router-af)# eigrp router-id 172.16.1.3`	EIGRP ピアがネイバーと通信する際に EIGRP がこのアドレスファミリに関して使用するデバイス ID を設定します。
ステップ 11	**af-interface default** 例： `Device(config-router-af)# af-interface default`	EIGRP 名前付きモード設定においてアドレスファミリに属するすべてのインターフェイスでインターフェイス固有のコマンドを設定します。アドレス ファミリ インターフェイス コンフィギュレーションモードを開始します。
ステップ 12	**bfd** 例： `Device(config-router-af-interface)# bfd`	すべてのインターフェイスで BFD を有効にします。

	コマンドまたはアクション	目的
ステップ 13	**End** 例： `Device(config-router-af-interface)# end`	アドレスファミリ インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 14	**show eigrp address-family ipv6 neighbors detail** 例： `Device# show eigrp address-family ipv6 neighbors detail`	（任意）インターフェイスでBFDが有効になっているEIGRPによって検出されたネイバーに関する詳細情報を表示します。
ステップ 15	**show bfd neighbors** 例： `Device# show bfd neighbors`	（任意）BFD情報をネイバーに表示します。

インターフェイスでの BFD サポートの設定

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ipv6 unicast-routing** 例： `Device(config)# ipv6 unicast-routing`	IPv6ユニキャストデータグラムの転送をイネーブルにします。
ステップ 4	**interface***type number* 例： `Device(config)# interface ethernet0/0`	インターフェイスのタイプと番号を指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 5	**ipv6 address** *ipv6-address /prefix-length* 例： `Device(config-if)# ipv6 address 2001:DB8:A:B::1/64`	IPv6 アドレスを設定します。

	コマンドまたはアクション	目的
ステップ 6	**bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier* 例： `Device(config-if)# bfd interval 50 min_rx 50 multiplier 3`	インターフェイスのベースライン BFD セッション パラメータを設定します。
ステップ 7	**exit** 例： `Device(config-if)# exit`	インターフェイス コンフィギュレーション モードを終了し、グローバル コンフィギュレーション モードに戻ります。
ステップ 8	**router eigrp** *virtual-name* 例： `Device(config)# router eigrp name`	EIGRP ルーティング プロセスを指定し、ルータ コンフィギュレーション モードを開始します。
ステップ 9	**address-family ipv6 autonomous-system***as-number* 例： `Device(config-router)# address-family ipv6 autonomous-system 3`	IPv6 のアドレス ファミリ コンフィギュレーション モードを開始して、EIGRP ルーティング インスタンスを設定します。
ステップ 10	**eigrp router-id***ip-address* 例： `Device(config-router-af)# eigrp router-id 172.16.1.3`	EIGRP ピアがネイバーと通信する際に EIGRP がこのアドレス ファミリに関して使用するデバイス ID を設定します。
ステップ 11	**af-interface** *interface-type interface-number* 例： `Device(config-router-af)# af-interface ethernet0/0`	EIGRP 名前付きモード設定においてアドレス ファミリに属するインターフェイスでインターフェイス固有のコマンドを設定します。アドレス ファミリ インターフェイス コンフィギュレーション モードを開始します。
ステップ 12	**bfd** 例： `Device(config-router-af-interface)# bfd`	指定されたインターフェイス上で BFD をイネーブルにします。
ステップ 13	**end** 例： `Device(config-router-af-interface)# end`	アドレス ファミリ インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
ステップ 14	**show eigrp address-family ipv6 neighbors** 例： `Device# show eigrp address-family ipv6 neighbors`	（任意）BFD が有効になっているネイバーを表示します。
ステップ 15	**show bfd neighbors** 例： `Device# show bfd neighbors`	（任意）BFD 情報をネイバーに表示します。

EIGRP IPv6 に対する BFD サポートの設定例

ここでは、EIGRP に対する BFD サポートの設定例を示します。

例：すべてのインターフェイスでの BFD サポートの設定

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

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

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

Max Nbrs: 0, Current Nbrs: 0

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

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

```
Device# show bfd neighbors

IPv6 Sessions
```

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

例：インターフェイスでの BFD サポートの設定

次に、インターフェイスで BFD サポートを設定する例を示します。

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

EIGRP IPv6 に対する BFD サポートの機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

表 *3 : EIGRP IPv6* に対する *BFD* サポートの機能情報

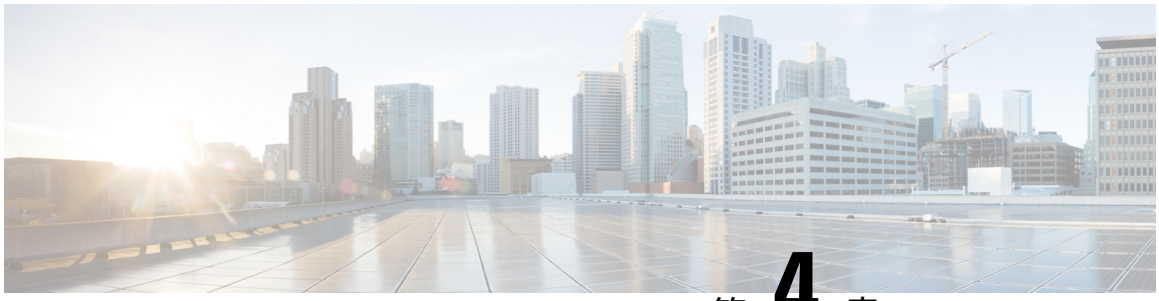
機能名	リリース	機能情報
EIGRP IPv6 に対する BFD サポート	Cisco IOS XE Gibraltar 16.11.x	この機能が導入されました。

その他の参考資料

関連資料

関連項目	マニュアル タイトル
BFD コマンド：コマンド構文、コマンドモード、コマンド履歴、デフォルト、使用に関する注意事項、および例。	次のドキュメントの IP ルーティングに関する項を参照してください：*Command Reference (Catalyst 9200 Series Switches)*

関連項目	マニュアル タイトル
EIGRP コマンド：コマンド構文の詳細、コマンドモード、コマンド履歴、デフォルト設定、使用に関する注意事項、および例	次のドキュメントの IP ルーティングに関する項を参照してください：*Command Reference (Catalyst 9200 Series Switches)*
EIGRP の設定	次のドキュメントのルーティングに関する項を参照してください：*Software Configuration Guide (Catalyst 9200 Switches)*



第 4 章

IP ユニキャスト ルーティングの設定

IP ユニキャスト ルーティングの設定に関する情報

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

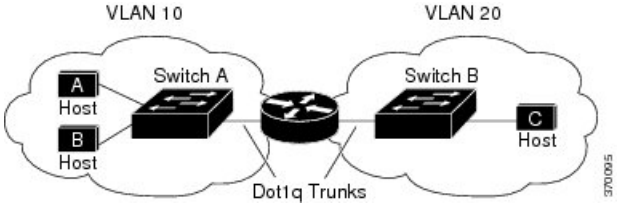
（注） IPv4 トラフィックに加えて、I6（IPv6）ユニキャストルーティングをイネーブルにし、IPv6 トラフィックを転送するようにインターフェイスを設定できます。

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 *1:* ルーティング トポロジの例

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



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

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

ルーティング タイプ

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

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

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

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

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

- ディスタンスベクトル プロトコルを使用するルータでは、ネットワーク リソースの距離の値を使用してルーティング テーブルを保持し、これらのテーブルをネイバーに定期的に渡します。ディスタンスベクトルプロトコルは1つまたは複数のメトリックを使用し、最適なルートを計算します。これらのプロトコルは、簡単に設定、使用できます。
- リンクステート プロトコルを使用するルータでは、ルータ間のリンクステート アドバタイズメント（LSA）の交換に基づき、ネットワーク トポロジに関する複雑なデータベースを保持します。LSA はネットワークのイベントによって起動され、コンバージェンス時

間、またはこれらの変更への対応時間を短縮します。リンクステートプロトコルはトポロジの変更にすばやく対応しますが、ディスタンスベクトルプロトコルよりも多くの帯域幅およびリソースが必要になります。

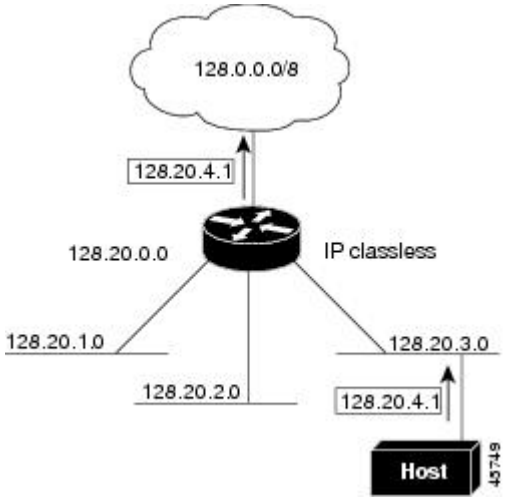
スイッチでサポートされているディスタンスベクトルプロトコルは、ルーティング情報プロトコル（RIP）です。RIP は最適パスを決定するために単一の距離メトリック（コスト）を使用します。また、Open Shortest Path First（OSPF）リンクステート プロトコル、および従来の Interior Gateway Routing Protocol（IGRP）にリンクステート ルーティング機能の一部を追加して効率化を図った Enhanced IGRP（EIGRP）もサポートされています。

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。

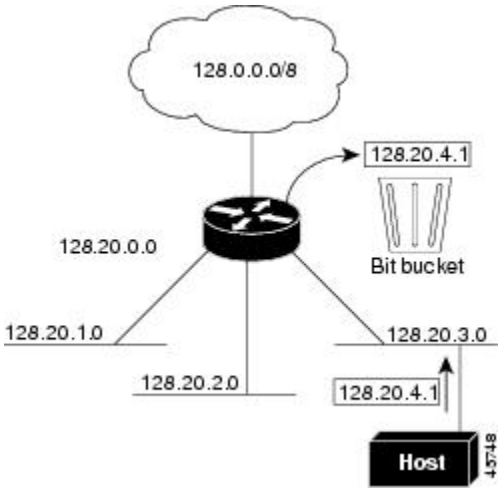
図では、クラスレスルーティングがイネーブルとなっています。ホストがパケットを128.20.4.1に送信すると、ルータはパケットを廃棄せずに、最適なスーパーネットルートに転送します。クラスレス ルーティングがディセーブルの場合、デフォルト ルートがないネットワークのサブネット宛てパケットを受信したルータは、パケットを廃棄します。

図 *2 : 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 *3: 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。

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。イーサネット以外の IEEE 802 ネットワークにおける IP データグラムのカプセル化、および ARP 要求や応答については、サブネットワーク アクセス プロトコル（SNAP）で規定されています。
- 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 サーバを設置する必要があります。サーバを識別するには、**ip rarp-server** *address* インターフェイス コンフィギュレーション コマンドを使用します。

プロシキ ARP

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

ICMP Router Discovery Protocol

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

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

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

UDP ブロードキャスト パケットおよびプロトコル

ユーザ 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、サーバを含まないネットワーク セグメント上にある場合、通常 UDP ブロードキャストは転送されません。この状況を改善するには、特定のクラスのブロードキャストをヘルパーアドレスに転送するように、ルータのインターフェイスを設定します。インターフェイスごとに、複数のヘルパー アドレスを使用できます。

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

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

ブロードキャスト パケットの処理

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

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

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

ルータはローカル ケーブルまでの範囲を制限して、ブロードキャスト ストームを防ぎます。ブリッジ（インテリジェントなブリッジを含む）はレイヤ 2 デバイスであるため、ブロードキャストはすべてのネットワーク セグメントに転送され、ブロードキャスト ストームを伝播します。ブロードキャスト ストーム問題を解決する最善の方法は、ネットワーク上で単一のブロードキャスト アドレス方式を使用することです。最新の IP 実装機能ではほとんどの場合、アドレスをブロードキャストアドレスとして使用するように設定できます。デバイスの場合も含めて、多くの実装機能では、ブロードキャストメッセージを転送するためのアドレス方式が複数サポートされています。

IP ブロードキャストのフラッディング

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

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

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

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

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

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

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

IP ルーティングの設定方法

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

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

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

（注） 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。

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

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

ルーティングを設定するための主な手順は次のとおりです。

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

IP アドレッシングの設定方法

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

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

- アドレス指定のデフォルト設定
- ネットワーク インターフェイスへの IP アドレスの割り当て
- アドレス解決方法の設定
- IP ルーティングがディセーブルの場合のルーティング支援機能
- ブロードキャスト パケットの処理方法の設定
- IP アドレスのモニタ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 *4:* アドレス指定のデフォルト設定

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

機能	デフォルト設定
IP 転送プロトコル	ヘルパー アドレスが定義されているか、またはユーザ データグラム プロトコル（UDP）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転送がイネーブルとなります 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 プロバイダー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： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config)#interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 4	**no switchport** 例： `Device(config-if)#no switchport`	レイヤ 2 コンフィギュレーション モードからインターフェイスを削除します（物理インターフェイスの場合）。
ステップ 5	**ip address** *ip-address subnet-mask* 例： `Device(config-if)#ip address 10.1.5.1 255.255.255.0`	IP アドレスおよび IP サブネット マスクを設定します。
ステップ 6	**no shutdown** 例： `Device(config-if)#no shutdown`	物理インターフェイスをイネーブルにします。
ステップ 7	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 8	**show ip route** 例： `Device#show ip route`	入力を確認します。

	コマンドまたはアクション	目的
ステップ 9	**show ip interface** [*interface-id*] 例： `Device#show ip interface gigabitethernet 1/0/1`	入力を確認します。
ステップ 10	**show running-config** 例： `Device#show running-config`	入力を確認します。
ステップ 11	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーションファイルに設定を保存します。

サブネット ゼロの使用

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

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

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	**ip subnet-zero** 例： `Device(config)#ip subnet-zero`	インターフェイス アドレスおよびルーティングのアップデート時にサブネットゼロの使用をイネーブルにします。
ステップ 4	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `Device#show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

クラスレス ルーティングのディセーブル化

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**no ip classless** 例：	クラスレスルーティング動作をディセーブルにします。

	コマンドまたはアクション	目的
	`Device(config)#no ip classless`	
ステップ 4	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `Device#show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

アドレス解決方法の設定

アドレス解決を設定するために必要な作業は次のとおりです。

スタティック ARP キャッシュの定義

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

手順

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

	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	**arp** *ip-address hardware-address type* 例： `Device(config)#ip 10.1.5.1 c2f3.220a.12f4 arpa`	ARP キャッシュ内で IP アドレスを MAC（ハードウェア）アドレスに関連付け、次に示すカプセル化タイプのいずれかを指定します。 • **arpa**：ARP カプセル化（イーサネット インターフェイス用） • **snap**：Subnetwork Address Protocol カプセル化（トークン リングおよび FDDI インターフェイス用） • **sap**：HP の ARP タイプ
ステップ 4	**arp** *ip-address hardware-address type* [**alias**] 例： `Device(config)#ip 10.1.5.3 d7f3.220d.12f5 arpa alias`	（任意）指定された IP アドレスがスイッチに属する場合と同じ方法で、スイッチが ARP 要求に応答するように指定します。
ステップ 5	**interface** *interface-id* 例： `Device(config)#interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するインターフェイスを指定します。
ステップ 6	**arp** *timeout seconds* 例： `Device(config-if)#arp 20000`	（任意）ARP キャッシュ エントリがキャッシュに保持される期間を設定します。デフォルト値は 14400 秒（4 時間）です。指定できる範囲は 0 ～ 2147483 秒です。
ステップ 7	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 8	**show interfaces** [*interface-id*] 例：	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

	コマンドまたはアクション	目的
	`Device#show interfaces gigabitethernet 1/0/1`	のタイプおよびタイムアウト値を確認します。
ステップ 9	**show arp** 例： `Device#show arp`	ARP キャッシュの内容を表示します。
ステップ 10	**show ip arp** 例： `Device#show ip arp`	ARP キャッシュの内容を表示します。
ステップ 11	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーションファイルに設定を保存します。

ARP のカプセル化の設定

IP インターフェイスでは、イーサネット ARP カプセル化（**arpa** キーワードで表される）がデフォルトで有効に設定されています。ネットワークの必要性に応じて、カプセル化方法を SNAP に変更できます。

カプセル化タイプを無効にするには、**no arp arpa** または **no arp snap** 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： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	**interface** *interface-id* 例： `Device(config)#interface gigabitethernet 1/0/2`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 4	**arp** {**arpa** | **snap**} 例： `Device(config-if)#arp arpa`	ARP カプセル化方法を指定します。 • **arpa**：Address Resolution Protocol • **snap**：Subnetwork Address Protocol
ステップ 5	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 6	**show interfaces** [*interface-id*] 例： `Device#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： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

プロキシ ARP のイネーブル化

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例：	グローバル 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)#interface gigabitethernet 1/0/2`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 4	**ip proxy-arp** 例： `Device(config-if)#ip proxy-arp`	インターフェイス上でプロキシ ARP をイネーブルにします。
ステップ 5	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 6	**show ip interface** [*interface-id*] 例： `Device#show ip interface gigabitethernet 1/0/2`	指定されたインターフェイスまたはすべてのインターフェイスの設定を確認します。
ステップ 7	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

IP ルーティングがディセーブルの場合のルーティング支援機能

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

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

プロシキ ARP

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

デフォルト ゲートウェイ

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `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 default-gateway** *ip-address* 例： `Device(config)#ip default gateway 10.1.5.1`	デフォルト ゲートウェイ（ルータ）を設定します。
ステップ 4	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 5	**show ip redirects** 例： `Device#show ip redirects`	設定を確認するため、デフォルト ゲートウェイ ルータのアドレスを表示します。

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

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： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config)#interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 4	**ip irdp** 例： `Device(config-if)#ip irdp`	インターフェイスでIRDP処理をイネーブルにします。
ステップ 5	**ip irdp multicast** 例：	（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

	コマンドまたはアクション	目的
	`Device(config-if)#ip irdp multicast`	（224.0.0.1）にIRDPアドバタイズを送信します。 （注） このコマンドを使用すると、IRDP パケットをマルチキャストとして送信するサンマイクロシステムズ社の Solaris との互換性を維持できます。実装機能の中には、これらのマルチキャストを受信できないものも多くあります。このコマンドを使用する前に、エンドホストがこの機能に対応していることを確認してください。
ステップ 6	**ip irdp holdtime** *seconds* 例： `Device(config-if)#ip irdp holdtime 1000`	（任意）アドバタイズが有効であるIRDP期間を設定します。デフォルトは **maxadvertinterval** 値の 3 倍です。**maxadvertinterval** 値よりも大きな値（9000秒以下）を指定する必要があります。**maxadvertinterval**値を変更すると、この値も変更されます。
ステップ 7	**ip irdp maxadvertinterval** *seconds* 例： `Device(config-if)#ip irdp maxadvertinterval 650`	（任意）アドバタイズメントの IRDP 最大間隔を設定します。デフォルトは 600 秒です。
ステップ 8	**ip irdp minadvertinterval** *seconds* 例： `Device(config-if)#ip irdp minadvertinterval 500`	（任意）アドバタイズ間の IRDP の最小インターバルを設定します。デフォルト値は **maxadvertinterval** 値の 0.75 倍です。**maxadvertinterval** を変更すると、この値も新しいデフォルト値（**maxadvertinterval** の 0.75 倍）に変更されます。
ステップ 9	**ip irdp preference** *number* 例： `Device(config-if)#ip irdp preference 2`	（任意）デバイスの IRDP プリファレンス レベルを設定します。指定できる範囲は -231 ～ 231 です。デフォルトは 0 です。大きな値を設定すると、ルータのプリファレンス レベルも高くなります。

	コマンドまたはアクション	目的
ステップ 10	**ip irdp address** *address* [*number*] 例： `Device(config-if)#ip irdp address 10.1.10.10`	（任意）プロキシアドバタイズを行うための IRDP アドレスとプリファレンスを設定します。
ステップ 11	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 12	**show ip irdp** 例： `Device#show ip irdp`	IRDP 値を表示し、設定を確認します。
ステップ 13	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーションファイルに設定を保存します。

ブロードキャスト パケットの処理方法の設定

これらの方式をイネーブルにするには、次に示す作業を実行します。

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

ダイレクト ブロードキャストから物理ブロードキャストへの変換のイネーブル化

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

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

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

（注） 出力インターフェイスで **ip directed-broadcast** コマンドを設定する前に、入力インターフェイスで **ip network-broadcast** コマンドを設定する必要があります。これにより、確実に、IP ダイレクトブロードキャストが正しく機能し、アップグレード後の停止の発生が防止されるようになります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config)#interface gigabitethernet 1/0/2`	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 network-broadcast** 例： `Device(config-if)#ip network-broadcast`	入力インターフェイスがネットワーク プレフィックス ダイレクト ブロードキャスト パケットを受信して受け入れることを可能にします。
ステップ 5	**exit** 例： `Device(config-if)#exit`	グローバル コンフィギュレーション モードに戻ります。

	コマンドまたはアクション	目的
ステップ 6	**interface** *interface-id* 例： `Device(config)#interface gigabitethernet 1/0/3`	インターフェイス コンフィギュレーション モードを開始し、設定するインターフェイスを指定します。
ステップ 7	**ip directed-broadcast** [*access-list-number*] 例： `Device(config-if)#ip directed-broadcast 103`	インターフェイス上で、ダイレクトブロードキャストから物理ブロードキャストへの変換をイネーブルにします。転送するブロードキャストを制御するアクセスリストを指定できます。アクセスリストを指定すると、アクセスリストで許可されているIPパケットだけが変換可能になります。
ステップ 8	**exit** 例： `Device(config-if)#exit`	グローバル コンフィギュレーション モードに戻ります。
ステップ 9	**ip forward-protocol** {**udp** [*port*] | **nd** | **sdns**} 例： `Device(config)#ip forward-protocol nd`	ブロードキャストパケットを転送するとき、ルータによって転送されるプロトコルおよびポートを指定します。 • **udp**：UPD データグラムを転送します。 port：（任意）転送される UDP サービスを制御する宛先ポートです。 • **nd**：ND データグラムを転送します。 • **sdns**：SDNS データグラムを転送します。
ステップ 10	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 11	**show ip interface** [*interface-id*] 例： `Device#show ip interface`	指定されたインターフェイスまたはすべてのインターフェイスの設定を確認します。

	コマンドまたはアクション	目的
ステップ 12	**show running-config** 例： `Device#show running-config`	入力を確認します。
ステップ 13	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

UDP ブロードキャスト パケットおよびプロトコルの転送

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config)#interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 4	**ip helper-address** *address* 例： `Device(config-if)#ip helper address 10.1.10.1`	転送をイネーブルにし、BOOTPなどのUDPブロードキャスト パケットを転送するための宛先アドレスを指定します。

	コマンドまたはアクション	目的
ステップ 5	**exit** 例： `Device(config-if)#exit`	グローバル コンフィギュレーション モードに戻ります。
ステップ 6	**ip forward-protocol** {**udp** [*port*] | **nd** | **sdns**} 例： `Device(config)#ip forward-protocol sdns`	ブロードキャストパケットを転送するときに、ルータによって転送されるプロトコルを指定します。
ステップ 7	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 8	**show ip interface** [*interface-id*] 例： `Device#show ip interface gigabitethernet 1/0/1`	指定されたインターフェイスまたはすべてのインターフェイスの設定を確認します。
ステップ 9	**show running-config** 例： `Device#show running-config`	入力を確認します。
ステップ 10	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーションファイルに設定を保存します。

IP ブロードキャスト アドレスの確立

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config)#interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するインターフェイスを指定します。
ステップ 4	**ip broadcast-address** *ip-address* 例： `Device(config-if)#ip broadcast-address 128.1.255.255`	デフォルト値と異なるブロードキャスト アドレス（128.1.255.255 など）を入力します。
ステップ 5	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 6	**show ip interface** [*interface-id*] 例： `Device#show ip interface`	指定されたインターフェイスまたはすべてのインターフェイスのブロードキャスト アドレスを確認します。
ステップ 7	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

IP ブロードキャストのフラッディング

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip forward-protocol spanning-tree** 例： `Device(config)#ip forward-protocol spanning-tree`	ブリッジング スパニングツリー データベースを使用し、UDP データグラムをフラッディングします。
ステップ 4	**ip forward-protocol turbo-flood** 例： `Device(config)#ip forward-protocol turbo-flood`	スパニングツリー データベースを使用し、UDP データグラムのフラッディングを高速化します。
ステップ 5	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 6	**show running-config** 例： `Device#show running-config`	入力を確認します。
ステップ 7	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

IP アドレスのモニタリングおよびメンテナンス

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

表 *5:* キャッシュ、テーブル、データベースをクリアするコマンド

コマンド	目的
clear arp-cache	IP ARP キャッシュおよび高速スイッチング キャッシュをクリアします。
clear host {*name* | *****}	ホスト名およびアドレス キャッシュから 1 つまたはすべてのエントリを削除します。
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削除します。

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 *6:* キャッシュ、テーブル、データベースを表示するコマンド

コマンド	目的
show arp	ARP テーブル内のエントリを表示します。
show hosts	デフォルトのドメイン名、検索サービスの方式、サーバ ホスト名、お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トを表示します。
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（エイリアス）。
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マスクを使用するサブネット番号を表示します。

コマンド	目的
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表示します。

IP ユニキャスト ルーティングの設定方法

IP ユニキャスト ルーティングのイネーブル化

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip routing** 例： `Device(config)# ip routing`	IPルーティングをイネーブルにします。
ステップ 4	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。

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

IP ルーティングの有効化の例

次に、IP ルーティングをイネーブルにする例を示します。

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

次の作業

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

- RIP
- OSPF
- EIGRP
- 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。

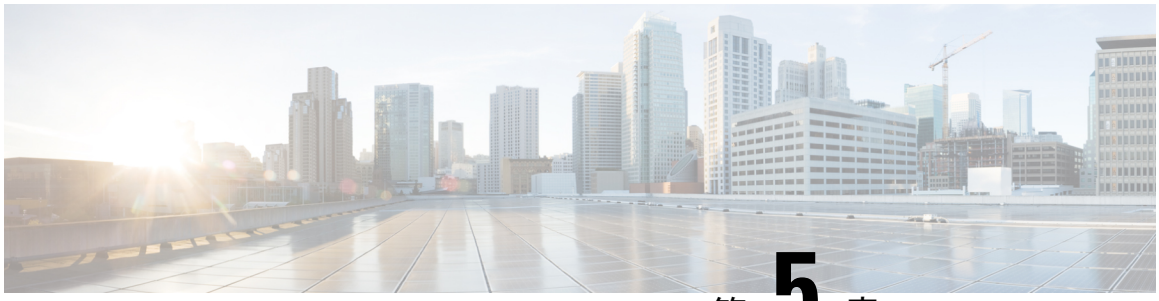
表 *7 : IP* ルートの削除またはルート ステータスの表示を行うコマンド

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ータスを表示します。

IP ユニキャスト ルーティングの機能情報

表 *8 : IP* ユニキャスト ルーティングの機能情報

リリース	機能	機能情報
Cisco IOS XE Fuji 16.9.2	IP ユニキャスト ルーティング	IP ユニキャストルーティングは、トラフィックをユニキャストアドレスに転送するルーティングプロセスです。ルータとレイヤ 3 スイッチは、事前にプログラムされたスタティックルートまたはデフォルトルートのいずれかを介してパケットをルーティングします。
Cisco IOS XE Amsterdam 17.3.1	新しいコマンドの **ip network-broadcast**	**ip network-broadcast** コマンドは、ネットワークプレフィックスダイレクト ブロードキャスト パケットを受信して受け入れるために導入されました。



第 5 章

IPv6 ユニキャスト ルーティングの設定

IPv6 ユニキャスト ルーティングの設定について

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

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

IPv6 の概要

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

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

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

IPv6 およびこの章のその他の機能については、

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

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

IPv6 のスタティック ルート

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

IPv6 のスタティック ルーティングの設定（CLI）

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

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

IPv6 ユニキャストのパス MTU ディスカバリ

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

ICMPv6

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

ネイバー探索

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

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

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

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

デフォルト ルータ プリファレンス

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

IPv6 ホストは、オフリンク宛先へのトラフィック用にルータを選択する、デフォルト ルータ リストを維持します。次に、宛先用に選択されたルータは、宛先キャッシュに格納されます。IPv6 NDP では、到達可能であるルータまたは到達可能性の高いルータが、到達可能性が不明または低いルータよりも優先されます。NDP は、到達可能または到達可能の可能性があるルータとして、常に同じルータを選択するか、またはルータ リストから繰り返し使用できます。DRP を使用することにより、IPv6 ホストが、両方ともが到達可能または到達可能の可能性がある 2 台のルータを差別化するように設定できます。

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

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

宛先ガード

IPv6 宛先ガード機能は、IPv6 ネイバー探索とともに動作して、リンク上でアクティブであると認識されているアドレスについてのみ、デバイスがアドレスを解決するようにします。アドレス グリーニング機能に依存して、リンク上でアクティブなすべての宛先をバインディング テーブルに挿入した後に、バインディング テーブルで宛先が見つからなかったときに実行される解決をブロックします。

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

MTU パスディスカバリ

IPv6 MTU パス ディスカバリを使用すると、ホストは指定されたデータ パスを通るすべてのリンクの最大伝送ユニット（MTU）サイズを動的に検出して、サイズに合わせて調整できます。

IPv4 の場合と同様に、IPv6 のパス MTU ディスカバリを使用すると、特定のデータ パス上のすべてのリンクの MTU サイズの差をホストが動的に検出し、調整できます。ただし、IPv6 では、特定のデータ パス上の 1 つのリンクのパス MTU がパケットのサイズに十分に対応できる大きさでない場合に、フラグメンテーションはパケットの送信元によって処理されます。IPv6 ホストにパケット フラグメンテーションを処理させると、IPv6 デバイスの処理リソースが節約され、IPv6 ネットワークの効率が向上します。

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

IPv6 のポリシーベース ルーティング

ポリシーベース ルーティング（PBR）は、トラフィック フローに定義ポリシーを設定し、ルートにおけるルーティング プロトコルへの依存度を軽くして、パケットのルーティングを柔軟に行えるようにします。したがって、PBR は、ルーティング プロトコルで提供される既存のメカニズムを拡張および補完することにより、ルーティングの制御を強化します。PBR を使用すると、IPv6 precedence を設定できます。単純なポリシーでは、これらのタスクのいずれかを使用し、複雑なポリシーでは、これらすべてのタスクを使用できます。高コストリンク上のプライオリティ トラフィックなど、特定のトラフィックのパスを指定することもできます。

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

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

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

PBR を使用すると、次の作業を実行できます。

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

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

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

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

サポートされていない IPv6 ユニキャスト ルーティング機能

スイッチは、次の IPv6 機能をサポートしません。

- サイトローカルなアドレス宛ての IPv6 パケット
- IPv4/IPv6 や IPv6/IPv4 などのトンネリング プロトコル

- IPv4/IPv6 または IPv6/IPv4 トンネリング プロトコルをサポートするトンネル エンドポイントとしてのスイッチ
- IPv6 Web Cache Communication Protocol（WCCP）

IPv6 機能の制限

スイッチでは IPv6 はハードウェアに実装されるため、ハードウェア メモリ内の IPv6 圧縮アドレスによる制限がいくつか発生します。これらのハードウェア制限により、機能の一部が失われて、制限されます。

機能の制限は次のとおりです。

- スイッチはハードウェアで SNAP カプセル化 IPv6 パケットを転送できません。これらはソフトウェアで転送されます。
- スイッチはソースルート IPv6 パケットに関する QoS 分類をハードウェアで適用できません。

IPv6 とスイッチ スタック

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

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

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

IPv6 アクティブスイッチおよびメンバーの機能は次のとおりです。

- アクティブスイッチ：
  - IPv6 ルーティングプロトコルの実行
  - ルーティング テーブルの生成
  - IPv6 用の分散型シスコ エクスプレス フォワーディングを使用するメンバースイッチにルーティングテーブルを配布します。
  - IPv6 ホスト機能および IPv6 アプリケーションの実行

- メンバースイッチ：
  - アクティブスイッチから IPv6 用のシスコ エクスプレス フォワーディングのルーティングテーブルを受信します。
  - ハードウェアへのルートのプログラミング

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

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

IPv6 のデフォルト設定

表 *9 : IPv6* のデフォルト設定

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

IPv6 ユニキャストルーティングの設定方法

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

IPv6 アドレッシングの設定と IPv6 ルーティングのイネーブル化

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

スイッチ上の IPv6 を設定する前に、次の注意事項に従ってください。

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

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

- インターフェイスに割り当てられた各ユニキャストアドレスの送信要求ノードマルチキャスト グループ FF02:0:0:0:0:1:ff00::/104（このアドレスはネイバー探索プロセスで使用される）
- すべてのノードを含む、ルータリンクに対してローカルなマルチキャスト グループ FF02::1
- すべてのルータを含む、リンクに対してローカルなマルチキャスト グループ FF02::2

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

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

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**sdm prefer** {**advanced** |**vlan**} 例： `(config)# sdm prefer vlan`	SDM テンプレートを選択します。 • **advanced**：スイッチをアドバンスドテンプレートに設定します。 • **vlan**：ハードウェアでのルーティングをサポートしないスイッチでの VLAN 設定を最適化します。
ステップ 4	**end** 例： `(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**reload** 例： `# reload`	オペレーティング システムをリロードします。
ステップ 6	**configure terminal** 例： `# configure terminal`	スイッチのリロード後、グローバル コンフィギュレーション モードを開始します。
ステップ 7	**interface** *interface-id* 例： `(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。インターフェイスは物理インターフェイス、スイッチ仮想インターフェイス（SVI）、またはレイヤ 3 EtherChannel に設定できます。

	コマンドまたはアクション	目的
ステップ 8	**no switchport** 例： `(config-if)# no switchport`	レイヤ 2 コンフィギュレーション モードからインターフェイスを削除します（物理インターフェイスの場合）。

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

ステップ 10	**exit** 例： `(config-if)# exit`	グローバル コンフィギュレーション モードに戻ります。
ステップ 11	**ip routing** 例：	スイッチ上で IP 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
	`(config)# ip routing`	
ステップ 12	**ipv6 unicast-routing** 例： `(config)# ipv6 unicast-routing`	IPv6 ユニキャスト データ パケットの転送をイネーブルにします。
ステップ 13	**end** 例： `(config)# end`	特権 EXEC モードに戻ります。
ステップ 14	**show ipv6 interface** *interface-id* 例： `# show ipv6 interface gigabitethernet 1/0/1`	入力を確認します。
ステップ 15	**copy running-config startup-config** 例： `# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

IPv4 および IPv6 プロトコル スタックの設定

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

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

手順

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

	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 routing** 例： `Device(config)# ip routing`	スイッチ上でルーティングをイネーブルにします。
ステップ 4	**ipv6 unicast-routing** 例： `Device(config)# ipv6 unicast-routing`	スイッチ上で IPv6 データ パケットの転送をイネーブルにします。
ステップ 5	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 6	**no switchport** 例： `Device(config-if)# no switchport`	レイヤ 2 コンフィギュレーション モードからインターフェイスを削除します（物理インターフェイスの場合）。
ステップ 7	**ip address** *ip-address mask* [**secondary**] 例： `Device(config-if)# ip address 10.1.2.3 255.255.255`	インターフェイスのプライマリまたはセカンダリ IPv4 アドレスを指定します。
ステップ 8	次のいずれかを使用します。 • **ipv6 address** *ipv6-prefix/prefix length* **eui-64** • **ipv6 address** *ipv6-address/prefix length* • **ipv6 address** *ipv6-address* **link-local** • **ipv6 enable** • **ipv6 address***WORD* • **ipv6 address***autoconfig* • **ipv6 address***dhcp*	• グローバル IPv6 アドレスを指定します。ネットワーク プレフィックスだけを指定します。最終の 64 ビットは、スイッチの MAC アドレスから自動的に計算されます。 • インターフェイスで IPv6 がイネーブルな場合に自動設定されるリンクローカルなアドレスでなく、インターフェイス上のリンクローカルなアドレスを使用するように指定します。 • インターフェイスに IPv6 リンクローカルなアドレスを自動設定し、インターフェイスでの IPv6 処理をイネーブルにします。リンクに対してローカルなアドレスを使

	コマンドまたはアクション	目的
		用できるのは、同じリンク上のノードと通信する場合だけです。 （注） インターフェイスから手動で設定したすべての IPv6 アドレスを削除するには、**no ipv6 address** インターフェイス コンフィギュレーション コマンドを引数なしで使用します。
ステップ 9	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 10	次のいずれかを使用します。 • **show interface** *interface-id* • **show ip interface** *interface-id* • **show ipv6 interface** *interface-id*	入力を確認します。
ステップ 11	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

デフォルト ルータ プリファレンス（DRP）の設定

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

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

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

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始して、DRP を指定するレイヤ 3 インターフェイスを特定します。
ステップ 4	**ipv6 nd router-preference** {**high** | **medium** | **low**} 例： `Device(config-if)# ipv6 nd router-preference medium`	スイッチ インターフェイス上のルータに DRP を指定します。
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show ipv6 interface** 例： `Device# show ipv6 interface`	設定を確認します。
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

IPv6 ICMP レート制限の設定

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

ICMP のレート制限パラメータを変更するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ipv6 icmp error-interval** *interval* [*bucketsize*] 例： `Device(config)# ipv6 icmp error-interval 50 20`	IPv6 ICMP エラー メッセージの間隔とバケット サイズを設定します。 • *interval*：バケットに追加されるトークンの間隔（ミリ秒）。指定できる範囲は 0 ～ 2147483647 ミリ秒です。 • *bucketsize*：（任意）バケットに格納される最大トークン数。指定できる範囲は 1 ～ 200 です。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show ipv6 interface** [*interface-id*] 例： `Device# show ipv6 interface gigabitethernet0/1`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

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

シスコ エクスプレス フォワーディングは、ネットワークパフォーマンスを最適化するためのレイヤ 3 IP スイッチングテクノロジーです。シスコ エクスプレス フォワーディングには高度な IP 検索および転送アルゴリズムが実装されているため、レイヤ 3 スイッチングのパフォーマンスを最大化できます。高速スイッチング ルート キャッシュよりも CPU にかかる負担が少

ないため、CEF はより多くの CPU 処理能力をパケット転送に振り分けることができます。IPv4 用のシスコ エクスプレス フォワーディングおよび分散型シスコ エクスプレス フォワーディングはデフォルトで有効になっています。IPv6 用のシスコ エクスプレス フォワーディングおよび分散型シスコ エクスプレス フォワーディングはデフォルトでは無効になっていますが、IPv6 ルーティングを設定すると自動的に有効になります。

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

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

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

IPv6 のスタティック ルーティングの設定

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

スタティック IPv6 ルーティングを設定するには、次の手順を実行します。

始める前に

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	**ipv6 route** *ipv6-prefix/prefix length* {*ipv6-address* | *interface-id* [*ipv6-address*]} [*administrative distance*] 例： `Device(config)# ipv6 route 2001:0DB8::/32 gigabitethernet2/0/1 130`	スタティック IPv6 ルートを設定します。 • *ipv6-prefix*：スタティック ルートの宛先となる IPv6 ネットワーク。スタティック ホスト ルートを設定する場合は、ホスト名も設定できます。 • */prefix length*：IPv6 プレフィックスの長さ。プレフィックス（アドレスのネットワーク部分）を構成するアドレスの上位連続ビット数を示す 10 進値です。10 進数値の前にスラッシュ記号が必要です。 • *ipv6-address*：指定したネットワークに到達するために使用可能なネクスト ホップの IPv6 アドレス。ネクスト ホップの IPv6 アドレスを直接接続する必要はありません。再帰処理が実行されて、直接接続されたネクスト ホップの IPv6 アドレスが検出されます。このアドレスは RFC 2373 に記載された形式（16 ビット値を使用したコロン区切りの 16 進表記で指定）で設定する必要があります。 • *interface-id*：Point-To-Point（ポイントツーポイント）インターフェイスおよびブロードキャスト インターフェイスからのダイレクト スタティック ルートを指定します。ポイントツーポイント インターフェイスの場合、ネクスト ホップの IPv6 アドレスを指定する必要はありません。ブロードキャスト インターフェイスの場合は、常にネクスト ホップの IPv6 アドレスを指定するか、または指定したプレフィックスをリンクに割り当てて、リンクに対してローカルなアドレスをネクスト ホップとして指定する必要があります。パケットの送信先となるネ

	コマンドまたはアクション	目的
		クスト ホップの IPv6 アドレスを指定することもできます。 （注） リンクに対してローカルなアドレスをネクスト ホップとして使用する場合は、*interface-id* を指定する必要があります（リンクに対してローカルなネクスト ホップを隣接ルータに設定する必要もあります）。 • *administrative distance*：（任意）アドミニストレーティブ ディスタンス。指定できる範囲は 1 ～ 254 です。デフォルト値は 1 で、この場合、接続されたルートを除くその他のどのルート タイプよりも、スタティック ルートが優先します。フローティング スタティック ルートを設定する場合は、ダイナミック ルーティング プロトコルよりも大きなアドミニストレーティブ ディスタンスを使用します。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

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

	コマンドまたはアクション	目的
		• **detail**：（任意）次に示す追加情報を表示します。 • 有効な再帰ルートの場合、出力パス セットおよび最大分解深度 • 無効なルートの場合、ルートが無効な理由
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

インターフェイスでの IPv6 PBR の有効化

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

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

PBR for IPv6 を有効にするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] 例： `Device(config)# route-map rip-to-ospf permit`	ルーティング プロトコル間でルートを再配布する条件を定義するか、ポリシー ルーティングを有効にしてルート マップ コンフィギュレーション モードを開始します。

ステップ 4	次のいずれかを実行します。
	• **match length** *minimum-length maximum-length*
	• **match ipv6 address** {**prefix-list** *prefix-list-name* | *access-list-name*}
	例：
	`Device(config-route-map)# match length 3 200`
	例：
	`Device(config-route-map)# match ipv6 address marketing`
		一致基準を指定します。
		• 次のうちの任意の項目またはすべてを指定できます。
		• レベル 3 のパケット長とのマッチング。
		• 指定された IPv6 アクセス リストとのマッチング。
		• **match** コマンドを指定しない場合、ルートマップはすべてのパケットに適用されます。

ステップ 5	次のいずれかを実行します。
	• **set ipv6 next-hop** *global-ipv6-address* [*global-ipv6-address...*]
	• **set ipv6 default next-hop** *global-ipv6-address* [*global-ipv6-address...*]
	例：
	`Device(config-route-map)# set ipv6 next-hop 2001:DB8:2003:1::95`
	例：
	`Device(config-route-map)# set ipv6 default next-hop 2001:DB8:2003:1::95`
		基準に一致したパケットに適用するアクション（1 つまたは複数）を指定します。
		• 次のうちの任意の項目またはすべてを指定できます。
		• パケットのルーティング先となるネクスト ホップを設定します（ネクスト ホップは隣接している必要があります）。
		• 宛先への明示的なルートがない場合に、パケットのルーティング先となるネクスト ホップを設定します。

ステップ 6	**exit** 例： `Device(config-route-map)# exit`	ルート マップ インターフェイス コンフィギュレーションモードを終了して、グローバル コンフィギュレーション モードに戻ります。
ステップ 7	**interface** *type number* 例： `Device(config)# interface FastEthernet 1/0`	インターフェイスのタイプと番号を指定し、ルータをインターフェイス コンフィギュレーション モードにします。

	コマンドまたはアクション	目的
ステップ 8	**ipv6 policy route-map** *route-map-name* 例： `Device(config-if)# ipv6 policy-route-map interactive`	インターフェイスで IPv6 PBR に使用するルート マップを特定します。
ステップ 9	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

ローカル PBR for IPv6 の有効化

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

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ipv6 local policy route-map** *route-map-name* 例： `Device(config)# ipv6 local policy route-map pbr-src-90`	デバイスによって生成されるパケットに対する IPv6 PBR を設定します。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

IPv6 の表示

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

表 *10 : IPv6* をモニタリングするコマンド

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

IPv6 ユニキャスト ルーティングの設定例

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

例：IPv4 および IPv6 プロトコルスタックの設定

次に、インターフェイス上で IPv4 および IPv6 ルーティングをイネーブルにする例を示します。

```
Device> enable
Device# configure terminal
```

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

例：デフォルト ルータ プリファレンスの設定

次に、インターフェイス上のルータに高い DRP を設定する例を示します。

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

例：IPv6 ICMP レート制限の設定

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

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

例：IPv6 のスタティックルーティングの設定

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

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

例：インターフェイスでの PBR のイネーブル化

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

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

例：ローカル PBR for IPv6 の有効化

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

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

例：IPv6 の表示

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

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

その他の参考資料

標準および RFC

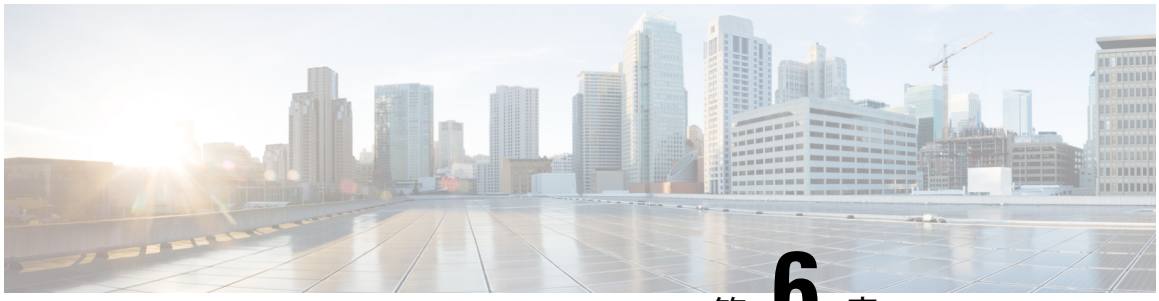
標準/RFC	タイトル
RFC 5453	予約済み *IPv6* インターフェイス識別子

機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

表 *11 : IPv6* ユニキャストおよびルーティングの機能情報

機能名	リリース	機能情報
IPv6 ユニキャストおよびルーティング	Cisco IOS XE Fuji 16.9.2	ユニキャストおよびルーティング機能が IPv6 に対してサポートされました。
RFC 5453	Cisco IOS XE Gibraltar 16.11.1	RFC 5453 がサポートされています。



第 6 章

RIP の設定

RIP 情報

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

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

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

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

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

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

RIP for IPv6

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

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

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

サマリー アドレスおよびスプリット ホライズン

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

RIP の設定方法

RIP のデフォルト設定

表 *12 : 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	ディセーブル
IP スプリット ホライズン	メディアにより異なる

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

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • プロンプトが表示されたらパスワードを入力します。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	**ip routing** 例： `Device(config)# ip routing`	IP ルーティングをイネーブルにします。（IP ルーティングがディセーブルになっている場合だけ、必須です）。
ステップ 4	**router rip** 例： `Device(config)# router rip`	RIP 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。
ステップ 5	**network** *network number* 例： `Device(config-router)# network 12.0.0.0`	ネットワークを RIP ルーティング プロセスと関連付けます。複数の **network** コマンドを指定できます。RIP ルーティング アップデートの送受信は、これらのネットワークのインターフェイスを経由する場合だけ可能です。 （注） RIP コマンドを有効にするには、ネットワーク番号を設定する必要があります。
ステップ 6	**neighbor** *ip-address* 例： `Device(config-router)# neighbor 10.2.5.1`	（任意）ルーティング情報を交換する隣接ルータを定義します。このステップを使用すると、RIP（通常はブロードキャスト プロトコル）からのルーティング アップデートが非ブロードキャスト ネットワークに到達するようになります。
ステップ 7	**offset-list** [*access-list number* | *name*] {**in** | **out**} *offset* [*type number*] 例： `Device(config-router)# offset-list 103 in 10`	（任意）オフセット リストをルーティング メトリックに適用し、RIP によって取得したルートへの着信および発信メトリックを増加します。アクセス リストまたはインターフェイスを使用し、オフセット リストを制限できます。
ステップ 8	**timers basic** *update invalid holddown flush* 例： `Device(config-router)# timers basic 45 360 400 300`	（任意）ルーティング プロトコル タイマーを調整します。すべてのタイマーの有効範囲は 0 ～ 4294967295 秒です。 • *update*：ルーティング アップデートの送信間隔。デフォルトは 30 秒です。

	コマンドまたはアクション	目的
		• *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： `Device(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、インターフェイスでの送受信に使用するバージョンを制御することもできます。
ステップ 10	**no auto summary** 例： `Device(config-router)# no auto summary`	（任意）自動要約をディセーブルにします。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。サマライズをディセーブルにし（RIP バージョン 2 だけ）、クラスフル ネットワーク境界にサブネットおよびホストルーティング情報をアドバタイズします。
ステップ 11	**output-delay** *delay* 例： `Device(config-router)# output-delay 8`	（任意）送信する RIP アップデートにパケット間遅延を追加します。デフォルトでは、複数のパケットからなる RIP アップデートのパケットに、パケット間遅延が追加されません。パケットを低速なデバイスに送信する場合は、8 ～ 50 ミリ秒のパケット間遅延を追加できます。
ステップ 12	**end** 例：	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
	`Device(config-router)# end`	
ステップ 13	**show ip protocols** 例： `Device# show ip protocols`	入力を確認します。
ステップ 14	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーションファイルに設定を保存します。

RIP 認証の設定

RIP バージョン 1 は認証をサポートしていません。RIP バージョン 2 のパケットを送受信する場合は、インターフェイスでRIP認証をイネーブルにできます。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 認証がイネーブルであるインターフェイスでは、プレーン テキストと MD5 という 2 つの認証モードがサポートされています。デフォルトはプレーン テキストです。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するインターフェイスを指定します。

	コマンドまたはアクション	目的
ステップ 4	**ip rip authentication key-chain** *name-of-chain* 例： `Device(config-if)# ip rip authentication key-chain trees`	RIP 認証をイネーブルにします。
ステップ 5	**ip rip authentication mode** {**text** | **md5**} 例： `Device(config-if)# ip rip authentication mode md5`	プレーン テキスト認証（デフォルト）または MD5 ダイジェスト認証を使用するように、インターフェイスを設定します。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

IPv6 RIP の設定

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

IPv6 の RIP ルーティングを設定するには、次の手順を実行します。

始める前に

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ipv6 router rip** *name* 例： `Device(config)# ipv6 router rip cisco`	IPv6 RIP ルーティング プロセスを設定し、このプロセスに対してルータ コンフィギュレーション モードを開始します。
ステップ 4	**maximum-paths** *number-paths* 例： `Device(config-router)# maximum-paths 6`	（任意）IPv6 RIP がサポートできる等コスト ルートの最大数を定義します。指定できる範囲は 1 ～ 32 で、デフォルトは 16 ルートです。
ステップ 5	**exit** 例： `Device(config-router)# exit`	グローバル コンフィギュレーション モードに戻ります。
ステップ 6	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 7	**ipv6 rip** *name* **enable** 例： `Device(config-if)# ipv6 rip cisco enable`	指定された IPv6 RIP ルーティング プロセスをインターフェイス上でイネーブルにします。
ステップ 8	**ipv6 rip** *name* **default-information** {**only** | **originate**} 例： `Device(config-if)# ipv6 rip cisco default-information only`	（任意）IPv6 デフォルト ルート（::/0）を RIP ルーティング プロセス アップデートに格納して、指定インターフェイスから送信します。

	コマンドまたはアクション	目的
		（注） 任意のインターフェイスから IPv6 デフォルト ルート（::/0）を送信したあとに、ルーティングループが発生しないようにするために、ルーティングプロセスは任意のインターフェイスで受信したすべてのデフォルトルートを無視します。 • **only**：このインターフェイスから送信するアップデートに、デフォルトルートを格納し、その他のすべてのルートを含めない場合に選択します。 • **originate**：このインターフェイスから送信するアップデートに、デフォルトルートおよびその他のすべてのルートを格納する場合に選択します。
ステップ 9	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

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

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

サマリー アドレスおよびスプリット ホライズンの設定

（注） ルートを適切にアドバタイズするため、アプリケーションがスプリット ホライズンをディセーブルにする必要がある場合を除き、通常はこの機能をディセーブル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 ホライズンがイネーブルの場合、自動サマリーとインターフェイス IP サマリー アドレスはともにアドバタイズされません。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 4	**ip address** *ip-address subnet-mask* 例： `Device(config-if)# ip address 10.1.1.10 255.255.255.0`	IP アドレスおよび IP サブネットを設定します。
ステップ 5	**ip summary-address rip ip address** *ip-network mask* 例：	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。

	コマンドまたはアクション	目的
	`Device(config-if)# ip summary-address rip ip address 10.1.1.30 255.255.255.0`	
ステップ 6	**no ip split horizon** 例： `Device(config-if)# no ip split horizon`	インターフェイスでスプリット ホライズンをディセーブルにします。
ステップ 7	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 8	**show ip interface** *interface-id* 例： `Device# show ip interface gigabitethernet 1/0/1`	入力を確認します。
ステップ 9	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

スプリット ホライズンの設定

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

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するインターフェイスを指定します。
ステップ 4	**ip address** *ip-address subnet-mask* 例： `Device(config-if)# ip address 10.1.1.10 255.255.255.0`	IP アドレスおよび IP サブネットを設定します。
ステップ 5	**no ip split-horizon** 例： `Device(config-if)# no ip split-horizon`	インターフェイスでスプリット ホライズンをディセーブルにします。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show ip interface** *interface-id* 例： `Device# show ip interface gigabitethernet 1/0/1`	入力を確認します。
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config`	（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
	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例

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

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

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

例：IPv6 用の RIP の設定

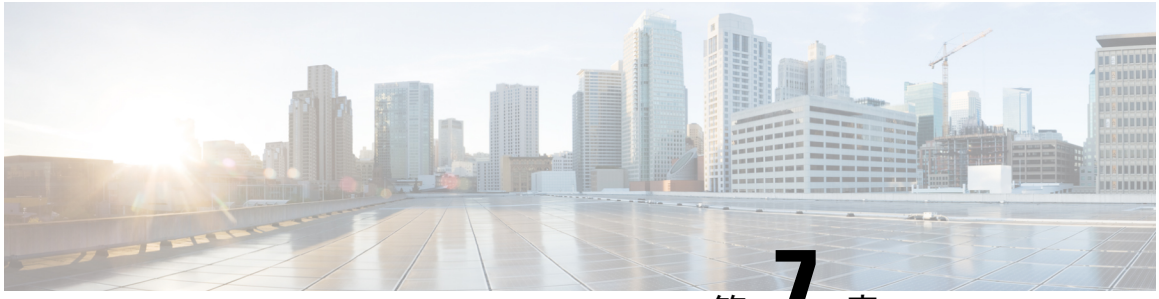
次に、最大 8 の等コスト ルートにより RIP ルーティング プロセス *cisco* をイネーブルにし、インターフェイス上でこれをイネーブルにする例を示します。

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

Routing Information Protocol に関する機能情報

表 *13 : Routing Information Protocol* に関する機能情報

リリース	機能情報
Cisco IOS XE Fuji 16.9.2	この機能が導入されました。



第 7 章

OSPF の設定

OSPF に関する情報

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

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

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

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

『OSPF for IPv6』

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

（注） Network Essentials ライセンスでは、1000 のルートの設定のみが許可されます。1000 を超えるルートを設定するには、Network Advantage ライセンスが必要です。

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

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

OSPF NSF

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

• OSPF NSF 認識 （112 ページ）
• OSPF NSF 対応 （112 ページ）

OSPF NSF 認識

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

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

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

OSPF エリア パラメータ

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

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

その他の OSPF パラメータ

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

- ルート集約：他のプロトコルからルートを再配信すると、各ルートは外部 LSA 内で個別にアドバタイズされます。OSPF リンクステートデータベースのサイズを小さくするには、**summary-address** ルータ コンフィギュレーション コマンドを使用し、指定されたネットワークアドレスおよびマスクに含まれる、再配信されたすべてのルートを単一のルータにアドバタイズします。

- 仮想リンク：OSPF では、すべてのエリアがバックボーン エリアに接続されている必要があります。バックボーンが不連続である場合に仮想リンクを確立するには、2 つの ABR を仮想リンクのエンドポイントとして設定します。設定情報には、他の仮想エンドポイント（他の ABR）の ID、および 2 つのルータに共通する非バックボーン リンク（通過エリア）などがあります。仮想リンクをスタブ エリアから設定できません。

- デフォルト ルート：OSPF ルーティング ドメイン内へのルート再配信を設定すると、ルータは自動的に自律システム境界ルータ（ASBR）になります。ASBR を設定し、強制的に OSPF ルーティング ドメインにデフォルト ルートを生成できます。

- すべての OSPF **show** 特権 EXEC コマンドでの表示にドメインネームサーバ（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 *14 : 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は10で、外部ルートタイプのデフォルトはタイプ 2 です。
デフォルト メトリック	各ルーティング プロトコルに適切な、組み込みの自動メトリック変換
距離 OSPF	dist1（エリア内のすべてのルート）：110。 dist2（エリア間のすべてのルート）：110。および dist3（他のルーティング ドメインからのルート）：110。

機能	デフォルト設定
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グされます。
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す。
ネットワーク エリア	ディセーブル
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。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router ospf** process-id 例： `Device(config)#router ospf 15`	OSPF ルーティングをイネーブルにし、ルータ コンフィギュレーション モードを開始します。プロセス ID はローカルに割り当てられ、内部で使用される識別パラメータで、任意の正の整数を指定できます。各 OSPF ルーティング プロセスには一意の値があります。 （注） OSPF for Routed Access は、OSPFv2 インスタンスと OSPFv3 インスタンスをそれぞれ 1 つずつと、最大 1000 のダイナミックに学習されるルートをサポートします。
ステップ 4	**network** address wildcard-mask **area** area-id 例： `Device(config-router)#network 10.1.1.1 255.240.0.0 area 20`	OSPF が動作するインターフェイス、およびそのインターフェイスのエリア ID を定義します。単一のコマンドにワイルドカードマスクを指定し、特定の OSPF エリアに関連付けるインターフェイスを 1 つまたは複数定義できます。エリア ID には 10 進数または IP アドレスを指定できます。
ステップ 5	**end** 例： `Device(config-router)#end`	特権 EXEC モードに戻ります。
ステップ 6	**show ip protocols** 例：	入力を確認します。

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

IPv6 OSPF の設定

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

IPv6 の OSPF ルーティングを設定するには、次の手順を実行します。

始める前に

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

次の注意事項に従ってください。

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ipv6 router ospf** *process-id* 例：	プロセスに対して OSPF ルータ コンフィギュレーション モードをイネーブ

	コマンドまたはアクション	目的
	`Device(config)# ipv6 router ospf 21`	ルにします。プロセス ID は、IPv6 OSPF ルーティング プロセスをイネーブルにする場合に管理上割り当てられる番号です。この ID はローカルに割り当てられ、1 ～ 65535 の正の整数を指定できます。
ステップ 4	**area** *area-id* **range** {*ipv6-prefix/prefix length*} [**advertise** | **not-advertise**] [**cost** *cost*] 例： `Device(config)# area .3 range 2001:0DB8::/32 not-advertise`	（任意）エリア境界でルートを統合および集約します。 • *area-id*：ルートをサマライズするエリアの ID。10 進数または IPv6 プレフィックスのどちらかを指定できます。 • *ipv6-prefix/prefix length*：宛先 IPv6 ネットワーク、およびプレフィックス（アドレスのネットワーク部分）を構成するアドレスの上位連続ビット数を示す 10 進数。10 進値の前にスラッシュ（/）を付加する必要があります。 • **advertise**：（任意）アドバタイズするアドレス範囲ステータスを設定し、タイプ 3 のサマリー リンク ステート アドバタイズメント（LSA）を生成します。 • **not-advertise**：（任意）アドレス範囲ステータスを DoNotAdvertise に設定します。Type3 サマリー LSA は抑制され、コンポーネント ネットワークは他のネットワークから隠された状態のままです。 • **cost** *cost*：（任意）現在のサマリー ルートのメトリックまたはコストを設定します。宛先への最短パスを判別する場合に、OSPF SPF 計算で使用します。指定できる値は 0 ～ 16777215 です。
ステップ 5	**maximum paths** *number-paths* 例： `Device(config)# maximum paths 16`	（任意）IPv6 OSPF がルーティング テーブルに入力する必要がある、同じ宛先への等コスト ルートの最大数を定

	コマンドまたはアクション	目的
		義します。指定できる範囲は 1 ～ 32 で、デフォルトは 16 です。
ステップ 6	**exit** 例： `Device(config-if)# exit`	グローバル コンフィギュレーション モードに戻ります。
ステップ 7	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 8	**ipv6 ospf** *process-id* **area** *area-id* [**instance** *instance-id*] 例： `Device(config-if)# ipv6 ospf 21 area .3`	インターフェイスで IPv6 の OSPF をイネーブルにします。 • **instance** *instance-id*：（任意）インスタンス ID
ステップ 9	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。

ステップ 10	次のいずれかを使用します。
	• **show ipv6 ospf** [*process-id*] [*area-id*] **interface** [*interface-id*]
	• **show ipv6 ospf** [*process-id*] [*area-id*]
	例：
	`Device# show ipv6 ospf 21 interface gigabitethernet2/0/1`
	または
	`Device# show ipv6 ospf 21`
		• OSPF インターフェイスに関する情報を表示します。
		• OSPF ルーティング プロセスに関する一般情報を表示します。

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

OSPF インターフェイスの設定

ip ospf インターフェイス コンフィギュレーション コマンドを使用すると、インターフェイス固有の OSPF パラメータを変更できます。これらのパラメータを変更する必要はありません

が、一部のインターフェイスパラメータ（helloインターバル、デッドインターバル、認証キーなど）については、接続されたネットワーク内のすべてのルータで統一性を維持する必要があります。これらのパラメータを変更した場合は、ネットワーク内のすべてのルータの値も同様に変更してください。

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** interface-id 例： `Device(config)#interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 4	**ip ospf** cost 例： `Device(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 5	**ip ospf retransmit-interval** seconds 例： `Device(config-if)#ip ospf transmit-interval 10`	（任意）LSA 送信間隔を秒数で指定します。指定できる範囲は 1 ～ 65535 秒です。デフォルト値は 5 秒です。
ステップ 6	**ip ospf transmit-delay** seconds 例： `Device(config-if)#ip ospf transmit-delay 2`	（任意）リンク ステートアップデート パケットを送信するまでの予測待機時間を秒数で設定します。指定できる範囲は 1 ～ 65535 秒です。デフォルト値は 1 秒です。

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

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

OSPF エリア パラメータの設定

始める前に

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router ospf** process-id 例： `Device(config)#router ospf 109`	OSPF ルーティングをイネーブルにし、ルータ コンフィギュレーション モードを開始します。
ステップ 4	**area** *area-id* **authentication** 例： `Device(config-router)#area 1 authentication`	（任意）特定のエリアへの無許可アクセスに対して、パスワードベースの保護を可能にします。ID には 10 進数または IP アドレスのいずれかを指定できます。
ステップ 5	**area** *area-id* **authentication message-digest** 例： `Device(config-router)#area 1 authentication message-digest`	（任意）エリアに関して MD5 認証を有効にします。
ステップ 6	**area** *area-id* **stub** [**no-summary**] 例： `Device(config-router)#area 1 stub`	（任意）エリアをスタブエリアとして定義します。**no-summary** キーワードを指定すると、ABR はサマリー リンク アドバタイズメントをスタブエリアに送信できなくなります。
ステップ 7	**area** *area-id* **nssa** [**no-redistribution**] [**default-information-originate**] [**no-summary**] 例： `Device(config-router)#area 1 nssa default-information-originate`	（任意）エリアを NSSA として定義します。同じエリア内のすべてのルータは、エリアが NSSA であることを認識する必要があります。次のキーワードのいずれかを選択します。 • **no-redistribution**：ルータが NSSA ABR の場合、**redistribute** コマンドを使用して、ルートを NSSA エリ

	コマンドまたはアクション	目的
		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。
ステップ 8	**area** *area-id* **range** *address mask* 例： `Device(config-router)#area 1 range 255.240.0.0`	（任意）単一のルートをアドバタイズするアドレス範囲を指定します。このコマンドは、ABR に対してだけ使用します。
ステップ 9	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 10	**show ip ospf** [*process-id*] 例： `Device#show ip ospf`	設定を確認するため、一般的な OSPF ルーティングプロセスまたは特定のプロセス ID に関する情報を表示します。
ステップ 11	**show ip ospf** [*process-id* [*area-id*]] **database** 例： `Device#show ip osfp database`	特定のルータの OSPF データベースに関連する情報のリストを表示します。
ステップ 12	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

その他の OSPF パラメータの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router ospf** *process-id* 例： `Device(config)#router ospf 10`	OSPF ルーティングをイネーブルにし、ルータ コンフィギュレーション モードを開始します。
ステップ 4	**summary-address** *address mask* 例： `Device(config)#summary-address 10.1.1.1 255.255.255.0`	（任意）1 つのサマリー ルートだけがアドバタイズされるように、再配信されたルートのアドレスおよび IP サブネット マスクを指定します。
ステップ 5	**area** *area-id* **virtual-link** *router-id* [**hello-interval** *seconds*] [**retransmit-interval** *seconds*] [**trans**] [[**authentication-key** *key*] | **message-digest-key** *keyid* **md5** *key*]] 例： `Device(config)#area 2 virtual-link 192.168.255.1 hello-interval 5`	（任意）仮想リンクを確立し、パラメータを設定します。
ステップ 6	**default-information originate** [**always**] [**metric** *metric-value*] [**metric-type** *type-value*] [**route-map** *map-name*] 例： `Device(config)#default-information originate metric 100 metric-type 1`	（任意）強制的に OSPF ルーティング ドメインにデフォルト ルートを生成するように ASBR を設定します。パラメータはすべて任意です。
ステップ 7	**ip ospf name-lookup** 例： `Device(config)#ip ospf name-lookup`	（任意）DNS 名検索を設定します。デフォルトでは無効になっています。

	コマンドまたはアクション	目的
ステップ 8	**ip auto-cost reference-bandwidth** *ref-bw* 例： `Device(config)#ip auto-cost reference-bandwidth 5`	（任意）単一のルートをアドバタイズするアドレス範囲を指定します。このコマンドは、ABR に対してだけ使用します。
ステップ 9	**distance ospf** {[**inter-area** *dist1*] [**inter-area** *dist2*] [**external** *dist3*]} 例： `Device(config)#distance ospf inter-area 150`	（任意）OSPF の距離の値を変更します。各タイプのルートのデフォルト距離は 110 です。有効値は 1 ～ 255 です。
ステップ 10	**passive-interface** *type number* 例： `Device(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。
ステップ 11	**timers throttle spf** *spf-delay spf-holdtime spf-wait* 例： `Device(config)#timers throttle spf 200 100 100`	（任意）ルート計算タイマーを設定します。 • *spf-delay*：SPF 計算の変更を受信する間の遅延。指定できる範囲は 1 ～ 600000 ミリ秒です。 • *spf-holdtime*：最初と 2 番目の SPF 計算の間の遅延。指定できる範囲は 1 ～ 600000 ミリ秒です。 • *spf-wait*：SPF 計算の最大待機時間（ミリ秒）。指定できる範囲は 1 ～ 600000 ミリ秒です。
ステップ 12	**ospf log-adj-changes** 例： `Device(config)#ospf log-adj-changes`	（任意）ネイバーステートが変更されたとき、syslog メッセージを送信します。
ステップ 13	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
ステップ 14	**show ip ospf** [*process-id* [*area-id*]] **database** 例： `Device#show ip ospf database`	特定のルータの OSPF データベースに関連する情報のリストを表示します。
ステップ 15	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーションファイルに設定を保存します。

LSA グループ ペーシングの変更

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `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 25`	OSPF ルーティングをイネーブルにし、ルータ コンフィギュレーション モードを開始します。
ステップ 4	**timers lsa-group-pacing** *seconds* 例： `Device(config-router)#timers lsa-group-pacing 15`	LSA の グループ ペーシングを変更します。
ステップ 5	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。

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

ループバック インターフェイスの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `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 loopback 0** 例： `Device(config)#interface loopback 0`	ループバック インターフェイスを作成し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**ip address** address mask 例： `Device(config-if)#ip address 10.1.1.5 255.255.240.0`	このインターフェイスに IP アドレスを割り当てます。
ステップ 5	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。

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

OSPF のモニタリング

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

表 *15 : IP OSPF* 統計情報の表示コマンド

コマンド	目的
show ip ospf [*process-id*]	OSPF ルーティング プロセスに関する一般情報を表示します。
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データベースに関連する情報のリストを表示します。
show ip ospf border-routes	内部の OSPF ルーティング ABR および ASBR テーブル エントリを表示します。

コマンド	目的
show ip ospf interface [*interface-name*]	OSPF に関連するインターフェイス情報を表示します。
show ip ospf neighbor [*interface-name*] [*neighbor-id*] **detail**	OSPF インターフェイス ネイバー情報を表示します。
show ip ospf virtual-links	OSPF に関連する仮想リンク情報を表示します。

OSPF の設定例

OSPF の設定例

例：基本的な OSPF パラメータの設定

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

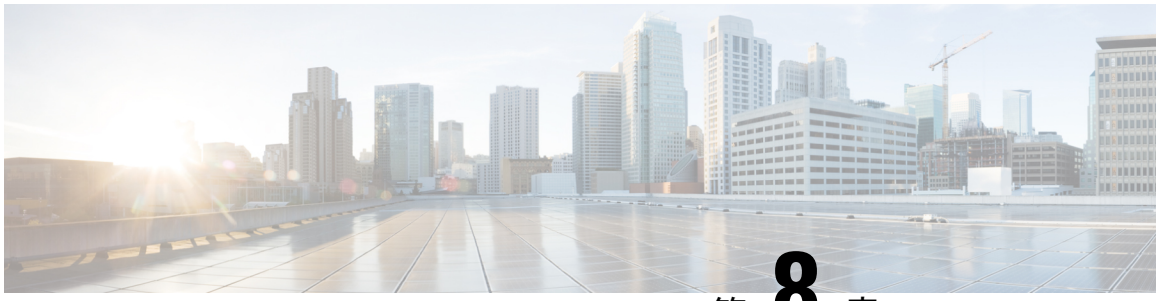
```
Device(config)#router ospf 109
Device(config-router)#network 131.108.0.0 255.255.255.0 area 24
```

OSPF の機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

表 *16 : OSPF* の機能情報

リリース	機能情報
Cisco IOS XE Fuji 16.9.2	この機能が導入されました



第 8 章

OSPFv3 のルート再配布数制限の設定

OSPFv3 のルート再配布数の制限に関する制約事項

この機能は、IPv6 アドレスファミリについてのみサポートされています。

OSPFv3 のルート再配布数制限の前提条件

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

OSPFv3 のルート再配布数制限について

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

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

OSPFv3 のルート再配布数制限を設定する方法

ここでは、OSPFv3 のルート再配布数制限の設定について説明します。

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

OSPFv3 のルート再配布数の制限

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router ospfv3** *process-id* 例： `Device(config)# router ospfv3 1`	OSPFv3 ルーティング プロセスを設定します。
ステップ 4	**address-family ipv6** [**unicast**] 例： `Device(config-router)# address-family ipv6 unicast`	IPv6 アドレス ファミリ コンフィギュレーション モードを開始します。
ステップ 5	**redistribute** *protocol* [*process-id*] [*as-number*] [**include-connected** {**level-1** | **level-1-2** | **level-2**} [**metric** *metric-value*] [**metric-type** *type-value*] [**nssa-only**] [**tag** *tag-value*] [**route-map** *map-tag*] 例： `Device(config-router-af)# redistribute eigrp 10`	ルートを 1 つのルーティング ドメインから他のルーティング ドメインに再配布します。

	コマンドまたはアクション	目的
ステップ 6	**redistribute maximum-prefix** *maximum* [*threshold*] 例： `Device(config-router-af)# redistribute maximum-prefix 100 80`	OSPFv3 への再配布が許可される IPv6 プレフィックスの最大数を設定します。 • 引数 *maximum* のデフォルト値はありません。 • *threshold* 値はデフォルトで 75% に設定されています。 （注） **warning-only** キーワードをこのコマンドで設定すると、再配布数の制限は設定されず、警告メッセージがログに記録されるようになります。
ステップ 7	**exit-address-family** 例： `Device(config-router-af)# exit-address-family`	IPv6 アドレス ファミリ コンフィギュレーション モードを終了します。
ステップ 8	**end** 例： `Device(config-router)# end`	ルータ コンフィギュレーション モードを終了します。

OSPFv3 へのルートの再配布数に関する警告メッセージの要求

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router ospfv3** *process-id* 例： `Device(config)# router ospfv3 1`	OSPFv3 ルーティング プロセスを設定します。

	コマンドまたはアクション	目的
ステップ 4	**address-family ipv6** [**unicast**] 例： `Device(config-router)# address-family ipv6 unicast`	IPv6 アドレス ファミリ コンフィギュレーション モードを開始します。
ステップ 5	**redistribute** *protocol* [*process-id*] [*as-number*] [**include-connected** {**level-1** | **level-1-2** | **level-2**} [**metric** *metric-value*] [**metric-type** *type-value*] [**nssa-only**] [**tag** *tag-value*] [**route-map** *map-tag*] 例： `Device(config-router-af)# redistribute eigrp 10`	ルートを 1 つのルーティング ドメインから他のルーティング ドメインに再配布します。
ステップ 6	**redistribute maximum-prefix** *maximum* [*threshold*] [**warning-only**] 例： `Device(config-router-af)# redistribute maximum-prefix 100 80 warning-only`	IP プレフィックスの最大数が OSPFv3 内に再配布されたときに警告メッセージのログが記録されます。 • **warning-only** キーワードが含まれているため、OSPFv3 へのプレフィックスの再配布数に制限は設定されません。 • 引数 *maximum* のデフォルト値はありません。 • *threshold* 値はデフォルトで 75% に設定されています。 • ここでは、1000 の 80%（800 個のルート再配布）で警告する場合と、1000 個のルート再配布で警告する場合の、2 つの例について説明します。
ステップ 7	**end** 例： `Device(config-router)# end`	ルータ コンフィギュレーション モードを終了します。

OSPFv3 のルート再配布数制限の設定例

ここでは、OSPFv3 のルート再配布数制限の設定例を示します。

例：OSPFv3 のルート再配布数の制限

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

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

例：ルートの再配布数に関する警告メッセージの要求

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

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

OSPFv3 のルート再配布数制限のモニタリング

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

表 *17 : OSPFv3* のルート再配布数制限をモニタするためのコマンド

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

その他の参考資料

関連資料

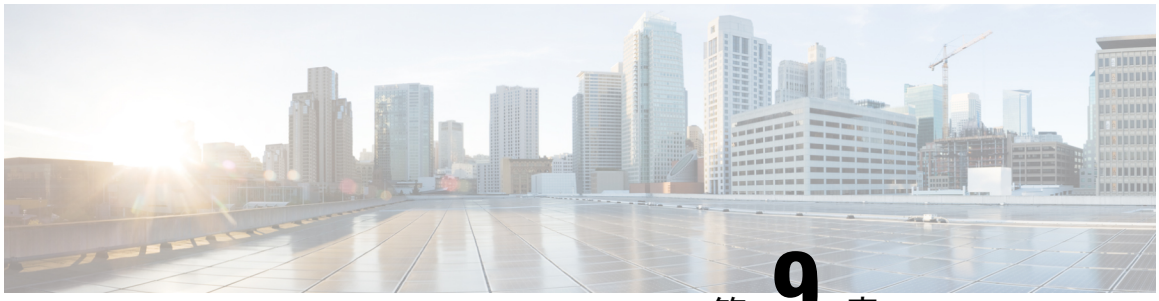
関連項目	マニュアル タイトル
この章で使用するコマンドの完全な構文および使用方法の詳細。	次のドキュメントのルーティングに関する項を参照してください：*Command Reference (Catalyst 9200 Series Switches)*

OSPFv3 のルート再配布数制限の機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

表 *18 : OSPFv3* のルート再配布数制限の機能情報

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



第 9 章

EIGRP の設定

EIGRP に関する情報

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

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

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

EIGRP IPv6

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

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

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

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

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

EIGRP の機能

EIGRP には次の機能があります。

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

EIGRP コンポーネント

EIGRP には次に示す 4 つの基本コンポーネントがあります。

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

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

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

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

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

EIGRP NSF対応のアクティブスイッチが再起動したとき、または新しいアクティブスイッチが起動して NSF が再起動したとき、このデバイスにはネイバーが存在せず、トポロジテーブルは空の状態です。デバイスは、デバイススタックに対するトラフィックを中断することなく、インターフェイスの起動、ネイバーの再取得、およびトポロジテーブルとルーティングテーブ

ルの再構築を行う必要があります。EIGRP ピアルータは新しいアクティブスイッチから学習したルートを維持し、NSF の再起動処理の間トラフィックの転送を継続します。

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

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

EIGRP スタブ ルーティング

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

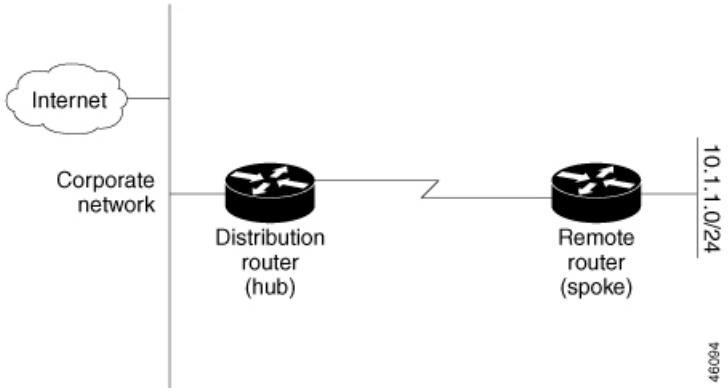
スタブルーティングは一般にハブアンドスポーク型のネットワークトポロジで使用されます。ハブアンドスポーク型ネットワークでは、1 つ以上のエンド（スタブ）ネットワークが 1 台のリモートデバイス（スポーク）に接続され、そのリモートデバイスは 1 つ以上のディストリビューションデバイス（ハブ）に接続されています。リモートデバイスは、1つ以上のディストリビューション デバイスに隣接しています。IP トラフィックがリモートデバイスに到達するための唯一のルートは、ディストリビューションデバイスを経由するものです。このタイプの設定は、一般的に、ディストリビューション デバイスが WAN に直接接続されている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

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

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

次の図は、単純なハブアンドスポーク型ネットワークを示しています。

図 *4*: 単純なハブアンドスポーク型ネットワーク



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

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

EIGRP スタブルーティング機能がない場合、ディストリビューション デバイスからリモートデバイスに送信されたルートがフィルタリングまたは集約された後でも、問題が発生することがあります。企業ネットワーク内でルートが失われると、EIGRPはクエリーをディストリビューションデバイスに送信できます。ルートがサマライズされている場合でも、ディストリビューションデバイスが代わりにリモートデバイスにクエリーを送信します。ディストリビューショ

ンデバイスとリモートデバイスの間の通信（WANリンクを介した）に問題がある場合、EIGRP Stuck In Active（SIA）状態が発生し、ネットワークのどこかで不安定になる可能性があります。EIGRPスタブルーティング機能を使用することにより、ネットワーク管理者はリモートデバイスへクエリーが送信されないようにできます。

EIGRPv6 スタブ ルーティング

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

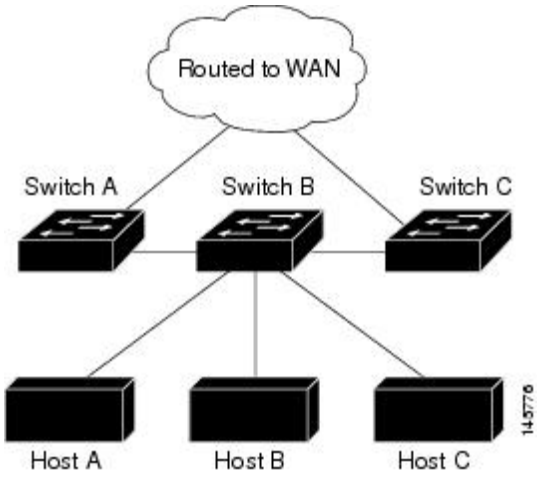
EIGRPv6 スタブ ルーティングを使用するネットワークでは、ユーザに対する IPv6 トラフィックの唯一の許容ルートは、EIGRPv6 スタブ ルーティングを設定しているスイッチ経由です。スイッチは、ユーザインターフェイスとして設定されているインターフェイスまたは他のデバイスに接続されているインターフェイスにルーテッド トラフィックを送信します。

EIGRPv6 スタブ ルーティングを使用しているときは、EIGRPv6 を使用してスイッチだけをスタブとして設定するように、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スイッチから伝播されます。スイッチは、サマリー、接続ルート、およびルーティング アップデートに対するすべてのクエリーに応答します。

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

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

図 *5: EIGRP* スタブ ルータ設定



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

EIGRP の設定方法

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

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

EIGRP のデフォルト設定

表 *19 : 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 プロセス間でデフォルト情報が渡されます。
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、接続されたルートおよびインターフェイスのスタティック ルートだけです。デフォルト メト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 ナノ秒の倍数である任意の正の数値 • 信頼性：0 ～ 255 の任意の数値（255 の場合は信頼性が 100%） • 負荷：0 ～ 255 の数値で表される有効帯域幅（255 の場合は 100% の負荷） • MTU：バイトで表されたルートの MTU サイズ（0 または任意の正の整数）

機能	デフォルト設定
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）ネットワークの場合：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、それ以外のネットワークの場合：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	レイヤ 3 スイッチでは、ハードウェアやソフ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トを転送し続けることができます。
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 IPv4 に対してサポートします。
オフセットリスト	ディセーブル
ルータ EIGRP	ディセーブル
メトリック設定	ルート マップにはメトリック設定なし
トラフィック共有	メトリックの比率に応じて配分
バリアンス	1（等コスト ロード バランシング）

基本的な EIGRP パラメータの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router eigrp** autonomous-system 例： `Device(config)#router eigrp 10`	EIGRP ルーティング プロセスをイネーブルにし、ルータ コンフィギュレーション モードを開始します。AS 番号によって他の EIGRP ルータへのルートを特定し、ルーティング情報をタグ付けします。
ステップ 4	**nsf** 例： `Device(config-router)#nsf`	（任意）EIGRP NSF をイネーブルにします。アクティブスイッチとそのすべてのピアでこのコマンドを入力します。
ステップ 5	**network** network-number 例： `Device(config-router)#network 192.168.0.0`	ネットワークを EIGRP ルーティング プロセスに関連付けます。EIGRP は指定されたネットワーク内のインターフェイスにアップデートを送信します。
ステップ 6	**eigrp log-neighbor-changes** 例： `Device(config-router)#eigrp log-neighbor-changes`	（任意）EIGRP 隣接関係変更のロギングをイネーブルにし、ルーティング システムの安定性をモニタします。
ステップ 7	**metric weights** *tos k1 k2 k3 k4 k5* 例： `Device(config-router)#metric weights 0 2 0 2 0 0`	（任意）EIGRP メトリックを調整します。デフォルト値はほとんどのネットワークで適切に動作するよう入念に設定されていますが、調整することも可能です。

	コマンドまたはアクション	目的
		注意 メトリックを設定する作業は複雑です。熟練したネットワーク設計者の指導がない場合は、行わないでください。
ステップ 8	**offset-list** [*access-list number* | *name*] {**in** | **out**} *offset* [*type number*] 例： `Device(config-router)#offset-list 21 out 10`	（任意）オフセットリストをルーティング メトリックに適用し、EIGRP によって取得したルートへの着信および発信メトリックを増加します。アクセスリストまたはインターフェイスを使用し、オフセットリストを制限できます。
ステップ 9	**auto-summary** 例： `Device(config-router)#auto-summary`	（任意）ネットワークレベルルートへのサブネットルートの自動サマライズをイネーブルにします。
ステップ 10	**interface***interface-id* 例： `Device(config-router)#interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーションモードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 11	**ip summary-address eigrp** *autonomous-system-number address mask* 例： `Device(config-if)#ip summary-address eigrp 1 192.168.0.0 255.255.0.0`	（任意）サマリー集約を設定します。
ステップ 12	**end** 例： `Device(config-if)#end`	特権 EXEC モードに戻ります。
ステップ 13	**show ip protocols** 例： `Device#show ip protocols`	入力を確認します。 NSF 認識の場合、出力に次のように表示されます。 *** IP Routing is NSF aware *** EIGRP NSF enabled
ステップ 14	**copy running-config startup-config** 例：	（任意）コンフィギュレーションファイルに設定を保存します。

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

EIGRP インターフェイスの設定

インターフェイスごとに、他の EIGRP パラメータを任意で設定できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface***interface-id* 例： `Device(config)#interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 4	**ip bandwidth-percent eigrp** *percent* 例： `Device(config-if)#ip bandwidth-percent eigrp 60`	（任意）インターフェイスで EIGRP が使用できる帯域幅の割合を設定します。デフォルト値は 50% です。
ステップ 5	**ip summary-address eigrp** *autonomous-system-number address mask* 例： `Device(config-if)#ip summary-address eigrp 109 192.161.0.0 255.255.0.0`	（任意）指定されたインターフェイスのサマリー集約アドレスを設定します（auto-summary がイネーブルの場合は、通常設定する必要はありません）。
ステップ 6	**ip hello-interval eigrp** *autonomous-system-number seconds* 例：	（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

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

IPv6 の EIGRP の設定

IPv6 EIGRP を実行するようにスイッチを設定する前に、 **ip routing global configuration** グローバル コンフィギュレーション コマンドを入力してルーティングを有効にし、**ipv6 unicast-routing**

global グローバル コンフィギュレーション コマンドを入力して IPv6 パケットの転送を有効にし、IPv6 EIGRP を有効にするレイヤ 3 インターフェイス上で IPv6 を有効にします。

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

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

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

EIGRP ルート認証の設定

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

手順

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

	コマンドまたはアクション	目的
ステップ 5	**ip authentication key-chain eigrp** *autonomous-system key-chain* 例： `Device(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 6	**exit** 例： `Device(config-if)#exit`	グローバル コンフィギュレーション モードに戻ります。
ステップ 7	**key chain** *name-of-chain* 例： `Device(config)#key chain chain1`	キーチェーンを識別し、キーチェーン コンフィギュレーション モードを開始します。ステップ 4 で設定した名前を指定します。
ステップ 8	**key** *number* 例： `Device(config-keychain)#key 1`	キーチェーン コンフィギュレーション モードで、キー番号を識別します。
ステップ 9	**key-string** *text* 例： `Device(config-keychain-key)#key-string key1`	キーチェーン コンフィギュレーション モードで、キー ストリングを識別します。
ステップ 10	**accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} 例： `Device(config-keychain-key)#accept-lifetime 13:30:00 Jan 25 2011 duration 7200`	（任意）キーを受信できる期間を指定します。 *start-time* および *end-time* 構文には、*hh:mm:ss Month date year* または *hh:mm:ss date Month year* のいずれかを使用できます。デフォルトは、デフォルトの *start-time* 以降、無制限です。指定できる最初の日付は 1993 年 1 月 1 日です。デフォルトの *end-time* および **duration** は **infinite** です。
ステップ 11	**send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} 例： `Device(config-keychain-key)#send-lifetime 14:00:00 Jan 25 2011 duration 3600`	（任意）キーを送信できる期間を指定します。 *start-time* および *end-time* 構文には、*hh:mm:ss Month date year* または *hh:mm:ss date Month year* のいずれかを使用できます。デフォルトは、デフォ

	コマンドまたはアクション	目的
		ルトの *start-time* 以降、無制限です。指定できる最初の日付は 1993 年 1 月 1 日です。デフォルトの *end-time* および **duration** は **infinite** です。
ステップ 12	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 13	**show key chain** 例： `Device#show key chain`	認証キーの情報を表示します。
ステップ 14	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーションファイルに設定を保存します。

EIGRP のモニタリングおよびメンテナンス

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

表 *20 : IP EIGRP* の *clear* および *show* コマンド

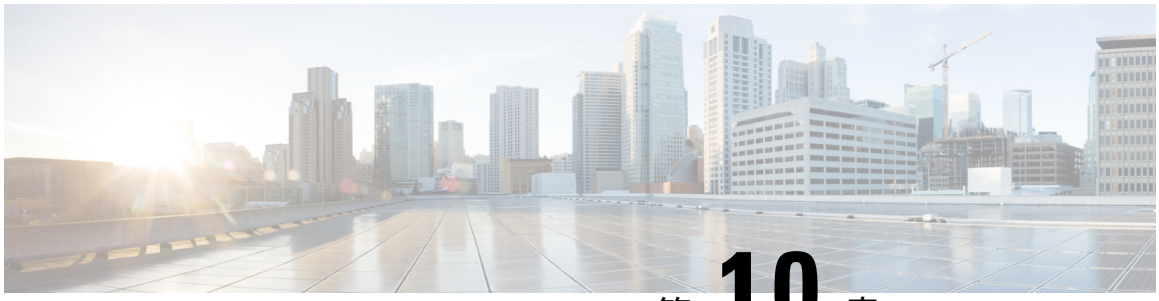
コマンド	目的
clear ip eigrp neighbors [*if-address* | *interface*]	ネイバーテーブルからネイバーを削除します。
show ip eigrp interface [*interface*] [*as number*]	EIGRP に設定されているインターフェイスに関する情報を表示します。
show ip eigrp neighbors [*type-number*]	EIGRP によって検出されたネイバーを表示します。
show ip eigrp topology [*autonomous-system-number*] | [[*ip-address*] *mask*]]	指定されたプロセスの EIGRP トポロジテーブルを表示します。
show ip eigrp traffic [*autonomous-system-number*]	すべてまたは指定された EIGRP プロセスの送受信パケット数を表示します。

EIGRP の機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

表 *21 : EIGRP* 機能の機能情報

リリース	機能情報
Cisco IOS XE Fuji 16.9.2	この機能が導入されました。



第 10 章

IS-IS ルーティングの設定

IS-IS ルーティングに関する情報

Integrated Intermediate System-to-Intermediate System（IS-IS）は、ISO ダイナミック ルーティング プロトコルの一つです（ISO 105890 を参照）。IS-IS をイネーブルするには、IS-IS ルーティングプロセスを作成し、それをネットワークではなく特定のインターフェイスに割り当てる必要があります。マルチエリア IS-IS コンフィギュレーション シンタックスを使用することで、レイヤ 3 デバイスごとに複数の IS-IS ルーティングプロトコルを指定できます。その後、IS-IS ルーティングプロセスのインスタンスごとにパラメータを設定する必要があります。

小規模の IS-IS ネットワークは、ネットワーク内にすべてのデバイスが含まれる単一のエリアとして構築されます。このネットワークは、その規模が大きくなるにしたがって、ローカルエリアに接続されたままの、接続済みのレベル 2 デバイスのセットで構成されるバックボーンエリア内に再編成されます。ローカル エリアの内部では、デバイスがすべてのシステム ID に到達する方法を認識しています。エリア間では、デバイスはバックボーンへの到達方法を認識しており、バックボーン デバイスは他のエリアに到達する方法を認識しています。

デバイスは、ローカルエリア内でルーティングを実行するために、レベル 1 の隣接関係を確立します（ステーションルーティング）。デバイスは、レベル 2 隣接関係を確立して、レベル 1 エリア間でルーティングを実行します（エリアルーティング）。

1 つの Cisco デバイスは、最大 29 エリアのルーティングに参加でき、バックボーンでレベル 2 ルーティングを実行できます。一般に、ルーティング プロセスごとに 1 つのエリアに対応します。デフォルトでは、設定されているルーティングプロセスの最初のインスタンスが、レベル 1 ルーティングと レベル 2 ルーティングの両方を実行します。追加のデバイスインスタンスを設定できます。このインスタンスは、自動的にレベル 1 エリアとして扱われます。IS-IS ルーティング プロセスの各インスタンスごとに個別にパラメータを設定する必要があります。

IS-IS マルチエリア ルーティングでは、シスコの各装置に対して最大 29 個の レベル 1 エリアを定義できますが、レベル 2 ルーティングを実行するプロセスは 1 つだけ設定できます。レベ

ル2ルーティングが任意のプロセス上に設定されている場合、追加のプロセスは、すべて自動的にレベル1に設定されます。同時に、このプロセスがレベル1ルーティングを実行するように設定することもできます。デバイスインスタンスにレベル2ルーティングが必要でない場合は、グローバル コンフィギュレーション モードで **is-type** コマンドを使用してレベル2の機能を削除します。別のデバイスインスタンスをレベル2デバイスとして設定する場合にも **is-type** コマンドを使用します。

NSF 認識

統合型 IS-IS ノンストップ フォワーディング（NSF）認識機能は IPv4G でサポートされています。この機能により、NSF を認識する顧客宅内機器（CPE）デバイスが、NFS 対応デバイスによるパケットのノンストップフォワーディングを実現します。ローカルデバイスでは、必ずしも NSF を実行している必要はありませんが、その NSF を認識機能により、スイッチオーバープロセス時にルーティングデータベースの完全性と精度、および隣接 NSF 対応デバイス上のリンクステートデータベースが保持できます。

統合型 IS-IS ノンストップ フォワーディング（NSF）認識機能は自動的に有効になり、設定は不要です。

IS-IS グローバル パラメータ

次に、設定可能なオプションの IS-IS グローバルパラメータを示します。

- ルートマップによって制御されるデフォルトルートを設定することで、デフォルトルートを IS-IS ルーティングドメイン内に強制的に設定できます。ルートマップで設定可能な、その他のフィルタリングオプションも指定できます。
- 内部チェックサムエラーとともに受信された IS-IS リンクステートパケット（LSP）を無視したり、破損したLSPを消去するようにデバイスを設定できます。これにより、LSPの発信側は、LSP を再生成します。
- エリアおよびドメインにパスワードを割り当てられます。
- ルーティングテーブルでサマリーアドレスによって表される（経路集約に基づいた）集約アドレスを作成できます。他のルーティングプロトコルから学習したルートも集約できます。サマリーをアドバタイズするのに使用されるメトリックは、すべての個別ルートにおける最小のメトリックです。
- 過負荷ビットを設定できます。
- LSPリフレッシュインターバルおよびLSPがリフレッシュなしでデバイスデータベース内にとどまることができる最大時間を設定できます。
- LSP生成に対するスロットリングタイマー、最短パス優先計算、および部分ルート計算を設定できます。
- IS-IS 隣接関係（アジャセンシー）がステートを変更（アップまたはダウン）する際に、デバイスがログメッセージを生成するように設定できます。

- ネットワーク内のリンクが、1500バイト未満の最大伝送ユニット（MTU）サイズの場合、それでもルーティングが行われるように LSP MTU の値を低くできます。
- **partition avoidance** コマンドを使用して、レベル 1-2 境界デバイス、隣接レベル 1 デバイス、およびエンドホスト間で完全な接続が失われた場合に、エリアがパーティション化されるのを防ぐことができます。

IS-IS インターフェイス パラメータ

任意で、特定のインターフェイス固有の IS-IS パラメータを、付加されている他のデバイスとは別に設定できます。ただし、デフォルト値（乗数およびタイムインターバルなど）を変更する場合、複数のデバイスおよびインターフェイス上でもこれを変更する必要があります。ほとんどのインターフェイスパラメータは、レベル1、レベル2、またはその両方で設定できます。

設定可能なインターフェイスレベルのパラメータは次のとおりです。

- インターフェイスのデフォルトメトリック：Quality of Service（QoS）ルーティングが実行されない場合に、IS-IS メトリックの値として使用され、割り当てられます。
- helloインターバル（インターフェイスから送信されるhelloパケットの間隔）またはデフォルトのhelloパケット乗数：インターフェイス上で使用されて、IS-IS helloパケットで送信されるホールド タイムを決定します。ホールド タイムは、ネイバーがダウンしていると宣言するまでに、別の hello パケットを待機する時間を決定します。これにより、障害リンクまたはネイバーが検出される速さも決定し、ルートを再計算できるようになります。helloパケットが頻繁に失われ、IS-IS 隣接に無用な障害が発生する場合は、hello 乗数を変更してください。hello 乗数を大きくし、それに対応して hello インターバルを小さくすると、リンク障害を検出するのに必要な時間を増やすことなく、helloプロトコルの信頼性を高めることができます。
- その他のタイム インターバル：
  - Complete Sequence Number PDU（CSNP）インターバル：CSNP は、データベースの同期を維持するために指定デバイスによって送信されます。
  - 再送信インターバル：これは、ポイントツーポイントリンクの IS-IS LSP の再送信間隔です。
  - IS-IS LSP 再送信スロットルインターバル：これは、IS-IS LSP がポイントツーポイントリンク上で再送信される最大レート（パケット間のミリ秒数）です。この間隔は、同じ LSP の連続した再送信の間隔である再送信インターバルとは異なります。
- 指定デバイスの選択の優先順位：マルチアクセスネットワークで必要な隣接数を削減し、その代わりに、ルーティング プロトコル トラフィックの量およびトポロジデータベースのサイズを削減できます。
- インターフェイス回線タイプ：指定されたインターフェイス上のネイバーに必要な隣接タイプです。
- インターフェイスのパスワード認証。

IS-IS の設定方法

ここでは、インターフェイスで IS-IS を有効にする方法、IS-IS グローバルパラメータを設定する方法、および IS-IS インターフェイスパラメータを設定する方法について説明します。

IS-IS のデフォルト設定

表 *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）とレベル2（エリア）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スタンスがレベル 1-2 ルータです。残りのインスタンスは、レベル 1 ルータです。
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 ミリ秒 初期 LSP 生成遅延：50 ミリ秒 最初と 2 番目の LSP 生成の間のホールド時間：200 ミリ秒
LSP 最大ライフ タイム（リフレッシュなし）	LSP パケットが削除されるまで 1200 秒（20 分）
LSP リフレッシュ インターバル	900 秒（15 分）ごと
最大 LSP パケット サイズ	1497 バイト
NSF 認識	イネーブルレイヤ3デバイスでは、ハードウェアやソフトウェアの変更中に、隣接するノンストップ フォワーディング対応ルータからのパケットを転送し続けることができます。

機能	デフォルト設定
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、認証はディセーブルになっています。
過負荷ビットの設定	ディセーブル。有効の際に引数が入力されない場合、過負荷ビットがただちに設定され、**no set-overload-bit** コマンドが入力されるまで設定されたままになります。
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。インターフェイス上でIS-ISルーティングをイネーブルにし、ルーティングプロセスの各インスタンスに対してエリアを指定します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	**configure terminal** 例：	グローバル 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	**clns routing** 例： `Device(config)#clns routing`	デバイス上で ISO コネクションレス型ルーティングをイネーブルに設定します。
ステップ 4	**router isis** [*area tag*] 例： `Device(config)#router isis tag1`	指定したルーティングプロセスに対して IS-IS ルーティングをイネーブルにし、IS-IS ルーティング コンフィギュレーション モードを開始します。 （任意）*area tag* 引数を使用して、IS-IS ルータが割り当てられているエリアを特定します。複数の IS-IS エリアを設定する場合は、値を入力します。 最初に設定された IS-IS インスタンスは、デフォルトでレベル 1-2 です。後のインスタンスは、自動的にレベル 1 に設定されます。グローバル コンフィギュレーション モードで **is-type** コマンドを使用してルーティングのレベルを変更できます。
ステップ 5	**net** *network-entity-title* 例： `Device(config-router)#net 47.0004.004d.0001.0001.0c11.1111.00`	ルーティング プロセスに NET を設定します。マルチエリア IS-IS を設定する場合は、各ルーティングプロセスに NET を指定します。NET およびアドレスの名前を指定します。
ステップ 6	**is-type** {**level-1** | **level-1-2** | **level-2-only**} 例： `Device(config-router)#is-type level-2-only`	（任意）レベル 1（ステーション）ルータ、マルチエリアルーティング用のレベル 2（エリア）ルータ、または両方（デフォルト）として機能するようにルータを設定します。 • **level 1**：ステーションルータとしてだけ機能します。 • **level 1-2**：ステーションルータおよびエリアルータの両方として機能します。 • **level 2**：エリアルータとしてだけ機能します。

	コマンドまたはアクション	目的
ステップ 7	**exit** 例： `Device(config-router)#end`	グローバル コンフィギュレーション モードに戻ります。
ステップ 8	**interface** *interface-id* 例： `Device(config)#interface gigabitethernet 1/0/1`	IS-IS をルーティングするインターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。インターフェイスがまだレイヤ 3 インターフェイスとして設定されていない場合は、**no switchport** コマンドを入力してインターフェイスをレイヤ 3 モードに設定します。
ステップ 9	**ip router isis** [*area tag*] 例： `Device(config-if)#ip router isis tag1`	インターフェイスに IS-IS ルーティング プロセスを設定し、エリア指示子をルーティング プロセスに割り当てます。
ステップ 10	**ip address** *ip-address-mask* 例： `Device(config-if)#ip address 10.0.0.5 255.255.255.0`	インターフェイスの IP アドレスを定義します。インターフェイスのいずれかで IS-IS ルーティングが設定されている場合は、IS-IS がイネーブルになっているエリアに含まれるすべてのインターフェイスに IP アドレスが必要です。
ステップ 11	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 12	**show isis** [*area tag*] **database detail** 例： `Device#show isis database detail`	入力を確認します。

IS-IS グローバル パラメータの設定

グローバル IS-IS パラメータを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router isis** 例： `Device(config)#router isis`	IS-IS ルーティング プロトコルを指定し、ルータ コンフィギュレーション モードを開始します。
ステップ 4	**default-information originate** [**route-map** *map-name*] 例： `Device(config-router)#default-information originate route-map map1`	（任意）デフォルト ルートを IS-IS ルーティング ドメインに強制的に設定します。**route-map** *map-name* コマンドを入力すると、にルーティングロセスによって有効なルートマップのデフォルトルートが生成されます。
ステップ 5	**ignore-lsp-errors** 例： `Device(config-router)#ignore-lsp-errors`	（任意）LSP を消去する代わりに、内部チェックサムにエラーがある LSP を無視するようにデバイスを設定します。このコマンドは、デフォルトでイネーブルになっています（破損した LSP はドロップされます）。破損した LSP を消去するには、ルータ コンフィギュレーション モードで **no ignore-lsp-errors** コマンドを入力します。
ステップ 6	**area-password** *password* 例： `Device(config-router)#area-password 1password`	（任意）レベル 1（ステーション ルータ レベル）LSP に挿入されるエリア認証パスワードを設定します。
ステップ 7	**domain-password** *password* 例： `Device(config-router)#domain-password 2password`	（任意）レベル 2（エリア ルータ レベル）LSP に挿入されるルーティング ドメイン認証パスワードを設定します。

	コマンドまたはアクション	目的
ステップ 8	**summary-address** *address mask* [**level-1** | **level-1-2** | **level-2**] 例： `Device(config-router)#summary-address 10.1.0.0 255.255.0.0 level-2`	（任意）所定のレベルのアドレスのサマリーを作成します。
ステップ 9	**set-overload-bit** [**on-startup** {*seconds* | **wait-for-bgp**}] 例： `Device(config-router)#set-overload-bit on-startup wait-for-bgp`	（任意）デバイスに問題がある場合に、他のデバイスが最短パス優先（SPF）計算でこのデバイスを無視するように過負荷ビットを設定します。 • （任意）**on-startup**：スタートアップ時だけ過負荷ビットを設定します。**on-startup** が指定されない場合、過負荷ビットが即座に設定され、**no set-overload-bit** コマンドを入力するまで設定されたままになります。**on-startup** が指定されている場合は、秒数または **wait-for-bgp** のどちらかを入力する必要があります。 • *seconds*：**on-startup** キーワードが設定されている場合、システム起動時に過負荷ビットが設定されて、指定した秒数の間設定されたままになります。指定できる範囲は 5 ～ 86400 秒です。 • **wait-for-bgp**：**on-startup** キーワードが設定されている場合、過負荷ビットがシステム起動時に設定され、BGP が収束するまで設定されたままになります。BGP が収束されたことが IS-IS に通知されない場合、IS-IS は 10 分後に過負荷ビットをオフにします。
ステップ 10	**lsp-refresh-interval** *seconds* 例： `Device(config-router)#lsp-refresh-interval 1080`	（任意）LSP リフレッシュインターバル（秒）を設定します。範囲は 1 ～ 65535 秒です。デフォルトでは、LSP リフレッシュを 900 秒（15 分）ごとに送信します。

	コマンドまたはアクション	目的
ステップ 11	**max-lsp-lifetime** *seconds* 例： `Device(config-router)#max-lsp-lifetime 1000`	（任意）LSP パケットがリフレッシュされずにルータ データベース内に存続する最大時間を設定します。範囲は 1 ～ 65535 秒です。デフォルト値は 1200 秒（20 分）です。指定された時間間隔のあと、LSP パケットは削除されます。
ステップ 12	**lsp-gen-interval** [**level-1** | **level-2**] *lsp-max-wait* [*lsp-initial-wait lsp-second-wait*] 例： `Device(config-router)#lsp-gen-interval level-2 2 50 100`	（任意）IS-IS 生成スロットリング タイマーを設定します。 • *lsp-max-wait*：生成される LAP の連続した 2 つのオカレンス間の最大インターバル（ミリ秒）。指定できる範囲は 1 ～ 120 ミリ秒です。デフォルト値は 5000 ミリ秒です。 • *lsp-initial-wait*：最初の LSP 生成遅延（ミリ秒）。指定できる範囲は 1 ～ 10000 ミリ秒です。デフォルト値は 50 ミリ秒です。 • *lsp-second-wait*：最初と 2 番目の LSP 生成間（ミリ秒）のホールド時間。指定できる範囲は 1 ～ 10000 ミリ秒です。デフォルト値は 200 ミリ秒です。
ステップ 13	**spf-interval** [**level-1** | **level-2**] *spf-max-wait* [*spf-initial-wait spf-second-wait*] 例： `Device(config-router)#spf-interval level-2 5 10 20`	（任意）IS-IS SPF スロットリングタイマーを設定します。 • *spf-max-wait*：連続する SFP 間（ミリ秒）の最大インターバル。指定できる範囲は 1 ～ 120 ミリ秒です。デフォルト値は 5000 ミリ秒です。 • *spf-initial-wait*：トポロジ変更後の最初の SFP 計算（ミリ秒）。指定できる範囲は 1 ～ 10000 ミリ秒です。デフォルト値は 50 ミリ秒です。 • *spf-second-wait*：最初と 2 番目の SFP 計算間（ミリ秒）のホールド時間。指定できる範囲は 1 ～ 10000

	コマンドまたはアクション	目的
		ミリ秒です。デフォルト値は 200 ミリ秒です。
ステップ 14	**prc-interval** *prc-max-wait* [*prc-initial-wait prc-second-wait*] 例： `Device(config-router)#prc-interval 5 10 20`	（任意）IS-IS PRC スロットリングタイマーを設定します。 • *prc-max-wait*：2 つの連続する PRC 計算間の最大インターバル（ミリ秒）。指定できる範囲は 1 ～ 120 ミリ秒です。デフォルト値は 5000 ミリ秒です。 • *prc-initial-wait*：トポロジ変更後の最初の PRC 計算遅延（ミリ秒）。指定できる範囲は 1 ～ 10,000 ミリ秒です。デフォルト値は 50 ミリ秒です。 • *prc-second-wait*：最初と 2 番目の PRC 計算間（ミリ秒）のホールドタイム。指定できる範囲は 1 ～ 10,000 ミリ秒です。デフォルト値は 200 ミリ秒です。
ステップ 15	**log-adjacency-changes** [**all**] 例： `Device(config-router)#log-adjacency-changes all`	（任意）IS-IS 隣接ステート変更をログするようルータを設定します。End System-to-Intermediate System PDU および LSP など、IS-IS hello に関連しないイベントにより生成されたすべての変更をログに含めるには、**all** を入力します。
ステップ 16	**lsp-mtu** *size* 例： `Device(config-router)#lsp mtu 1560`	（任意）最大 LSP パケットサイズ（バイト）を指定します。指定できる範囲は 128 ～ 4352 バイトです。デフォルト値は 1497 バイトです。 （注） ネットワーク内のリンクで MTU サイズが縮小された場合、ネットワーク内のすべてのデバイスで LSP MTU サイズを変更する必要があります。

	コマンドまたはアクション	目的
ステップ 17	**partition avoidance** 例： `Device(config-router)#partition avoidance`	（任意）境界ルータ、すべての隣接レベル 1 ルータ、およびエンド ホスト間で、フル接続が切断された場合、IS-IS レベル 1-2 境界ルータがレベル 1 エリア プレフィックスをレベル 2 バックボーンにアドバタイズしないようにします。
ステップ 18	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。

IS-IS インターフェイス パラメータの設定

IS-IS インターフェイス固有のパラメータを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config)#interface gigabitethernet 1/0/1`	設定するインターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。インターフェイスがまだレイヤ 3 インターフェイスとして設定されていない場合は、**no switchport** コマンドを入力してインターフェイスをレイヤ 3 モードに設定します。
ステップ 4	**isis metric** *default-metric* [**level-1** | **level-2**] 例： `Device(config-if)#isis metric 15`	（任意）指定したインターフェイスにメトリック（またはコスト）を設定します。指定できる範囲は 0 ～ 63 です。デフォルトは 10 です。レベルが入力さ

	コマンドまたはアクション	目的
		れない場合は、レベル 1 ルータとレベル 2 ルータの両方にデフォルト値が適用されます。
ステップ 5	**isis hello-interval** {*seconds* | **minimal**} [**level-1** | **level-2**] 例： `Device(config-if)#isis hello-interval minimal`	（任意）デバイスが hello パケットを送信する間隔を指定します。デフォルトでは、hello インターバル *seconds* の 3 倍の値が、送信される hello パケットの *holdtime* としてアドバタイズされます。hello インターバルが狭まると、トポロジ変更の検出も速くなりますが、ルーティング トラフィック量は増大します。 • **minimal**：結果として得られるホールドタイムが 1 秒になるように、hello 乗数に基づいて hello 間隔が計算されます。 • *seconds*：指定できる範囲は 1 ～ 65535 です。デフォルトは 10 秒です。
ステップ 6	**isis hello-multiplier** *multiplier* [**level-1** | **level-2**] 例： `Device(config-if)#isis hello-multiplier 5`	（任意）ネイバーが見落とすことができる IS-IS hello パケット数の最大値を指定します。見落とされたパケット数がこの値を超えると、デバイスは隣接がダウンしていると宣言します。指定できる範囲は 3 ～ 1000 です。デフォルトは 3 です。 （注） hello 乗数を小さくすると、高速コンバージェンスとなりますが、ルーティングが不安定になる場合があります。
ステップ 7	**isis csnp-interval** *seconds* [**level-1** | **level-2**] 例： `Device(config-if)#isis csnp-interval 15`	（任意）インターフェイスに IS-IS CSNP を設定します。指定できる範囲は 0 ～ 65535 です。デフォルトは 10 秒です。
ステップ 8	**isis retransmit-interval** *seconds* 例：	（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

	コマンドまたはアクション	目的
	`Device(config-if)#isis retransmit-interval 7`	つのルータ間で予測されるラウンドトリップ遅延よりも大きい値を指定してください。指定できる範囲は0～65535です。デフォルトは 5 秒です。
ステップ 9	**isis retransmit-throttle-interval** *milliseconds* 例： `Device(config-if)#isis retransmit-throttle-interval 4000`	（任意）IS-IS LSP 再送信スロットルインターバルを設定します。これは、IS-IS LSP がポイントツーポイントリンク上で再送信される最大レート（パケット間のミリ秒数）です。指定できる範囲は 0 ー 65535 です。デフォルトは **isis lsp-interval** コマンドによって決定されます。
ステップ 10	**isis priority** *value* [**level-1** | **level-2**] 例： `Device(config-if)#isis priority 50`	（任意）指定ルータの優先順位を設定します。指定できる範囲は 0 ～ 127 です。デフォルトは 64 です。
ステップ 11	**isis circuit-type** {**level-1** | **level-1-2** | **level-2-only**} 例： `Device(config-if)#isis circuit-type level-1-2`	（任意）指定されたインターフェイス上のネイバーで必要な隣接タイプを設定します（インターフェイスの回線タイプを指定します）。 • **level-1**：このノードとネイバーの両方に共通のエリアアドレスが少なくとも 1 つある場合、レベル 1 隣接関係が確立されます。 • **level-1-2**：ネイバーもレベル 1 およびレベル 2 の両方として設定されていて、少なくとも 1 つの共通のエリアがある場合、レベル 1 およびレベル 2 隣接関係が確立されます。共通のエリアがない場合は、レベル 2 隣接関係が確立されます。これはデフォルト設定です。これがデフォルトのオプションです。 • **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 12	**isis password** *password* [**level-1** | **level-2**] 例： `Device(config-if)#isis password secret`	（任意）インターフェイスの認証パスワードを設定します。デフォルトでは、認証はディセーブルに設定されています。レベル 1 またはレベル 2 を指定すると、それぞれレベル 1 またはレベル 2 ルーティング用のパスワードだけがイネーブルになります。レベルを指定しない場合、デフォルトはレベル 1 およびレベル 2 です。
ステップ 13	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。

IS-IS 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	目的
show ip route isis	IS-IS IP ルーティングテーブルの現在のステートを表示します。
show isis database	IS-IS リンクステート データベースを表示します。
show isis routes	IS-IS レベル 1 ルーティング テーブルを表示します。
show isis spf-log	IS-IS の SPF 計算の履歴を表示します。
show isis topology	すべてのエリア内の接続されたルータすべてのリストを表示します。
show route-map	設定済みのすべてのルートマップを表示するか、指定した 1 つのルートマップだけを表示します。

コマンド	目的
trace clns *[接続先（Destination）]*	ネットワークのパケットが指定された宛先までに経由するパスをトレースします。

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 Fuji 16.9.2	この機能が導入されました。

第 11 章

VRF-Lite の設定

VRF-Lite について

VRF-Lite の機能によって、サービス プロバイダーは、VPN 間で重複した IP アドレスを使用できる複数の VPN をサポートできます。VRF-Lite は入力インターフェイスを使用して異なる VPN のルートを区別し、各 VRF に 1 つまたは複数のレイヤ 3 インターフェイスを対応付けて仮想パケット転送テーブルを形成します。VRF のインターフェイスは、イーサネット ポートなどの物理インターフェイス、または VLAN SVI などの論理インターフェイスにすることができますが、レイヤ 3 インターフェイスは、一度に複数の VRF に属することはできません。

（注） VRF-Lite インターフェイスは、レイヤ 3 インターフェイスである必要があります。

VRF-Lite には次のデバイスが含まれます。

- CE デバイスにおいて、カスタマーは、1 つまたは複数の プロバイダー エッジ（PE）ルータへのデータ リンクを介してサービス プロバイダー ネットワークにアクセスできます。CE デバイスは、サイトのローカル ルートをプロバイダー エッジ ルータにアドバタイズし、そこからリモート VPN ルートを学習します。Cisco Catalyst スイッチは、CE にすることができます。
- プロバイダー エッジ（PE）ルータは、スタティック ルーティングまたはルーティング プロトコル（BGP、RIPv1、RIPv2 など）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 プロバイダー 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。

- プロバイダー ルータ（またはコア ルータ）とは、サービス プロバイダー ネットワーク内にあり、CE デバイスに接続していないすべてのルータです。

VRF-Lite を使用すると、複数のお客様が 1 つの CE を共有できます。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。VRF-Lite により、CE デバイスは、個別の VRF テーブルを保持し、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まで拡張することができます。

次の図に、各 Cisco Catalyst スイッチが複数の仮想 CE として機能する設定を示します。VRF-Lite はレイヤ 3 機能であるため、VRF の各インターフェイスはレイヤ 3 インターフェイスである必要があります。

VRF を設定するには、VRF テーブルを作成し、VRF に対応付けられたレイヤ 3 インターフェイスを指定します。

VRF-Lite の設定に関するガイドライン

IPv4 と IPv6

- VRF-Lite が設定されたスイッチは複数のカスタマーで共有され、すべてのカスタマーが独自のルーティング テーブルを持ちます。
- カスタマーは別々の VRF テーブルを使用するので、同じ IP アドレスを再利用できます。
- VRF-Lite では、複数のカスタマーが PE と CE の間で同一の物理リンクを共有できます。
- Cisco Catalyst スイッチでは、物理ポートか VLAN SVI、またはその両方の組み合わせを使用して、VRF を設定できます。アクセス ポートまたはトランク ポート経由で SVI を接続できます。
- 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。
- レイヤ 3 TCAM リソースは、すべての VRF 間で共有されます。各 VRF が十分な CAM 領域を持つようにするには、**maximum routes** コマンドを使用します。
- VRF を使用した Cisco Catalyst スイッチは、1 つのグローバル ネットワークと複数の VRF をサポートできます。サポートされるルートの総数は、TCAM のサイズに制限されます。

- 1 つの VRF を IPv4 と IPv6 の両方に設定できます。
- 着信パケットの宛先アドレスが VRF テーブルにない場合、そのパケットはドロップされます。また、VRF ルートに TCAM 領域が十分にない場合、その VRF のハードウェア切り替えは無効になり、対応するデータ パケットがソフトウェアに送信されて処理されます。

IPv4 固有

- Cisco Catalyst スイッチでは、PIM-SM プロトコル と PIM-SSM プロトコルがサポートされます。

IPv6 固有

- VRF 認識 OSPFv3、EIGRPv6、および IPv6 スタティックルーティングがサポートされます。
- VRF 認識 IPv6 ルート アプリケーションには、ping、telnet、ssh、tftp、ftp、およびトレースルートが含まれています（このリストには管理インターフェイスは含まれていません。これは、その下に IPv4 も IPv6 も設定できますが、別々に処理されます）。

VRF-Lite の設定方法

ここでは、VRF-Lite の設定について説明します。

IPv4 用の VRF-Lite の設定

ここでは、IPv4 用の VRF-Lite の設定について説明します。

VRF 認識サービスの設定

IP サービスは、グローバルなインターフェイス上と、グローバルなルーティング インスタンス内で設定できます。IP サービスは複数のルーティング インスタンス上で稼働するように拡張されます。これが、VRF 認識です。システム内の任意の設定済み VRF であればいずれも、VRF 認識サービス用に指定できます。

VRF 認識サービスは、プラットフォームから独立したモジュールに実装されています。VRF は、Cisco IOS 内の複数のルーティング インスタンスを提供します。各プラットフォームには、サポートする VRF 数に関して独自の制限があります。

VRF 認識サービスには、次の特性があります。

- ユーザは、ユーザ指定の 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。ユーザは、特定の VRF の ARP エントリを表示できます。

ARP のユーザ インターフェイスの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**show ip arp vrf** *vrf-name* 例： `Device# show ip arp vrf vrf-name`	指定された VRF で、ARP テーブル（スタティック エントリおよびダイナミック エントリ）を表示します。
ステップ 2	**arp vrf** *vrf-name ip-address mac-address ARPA* 例： `Device(config)# arp vrf vrf-name ip-address mac-address ARPA`	指定された VRF でスタティック ARP エントリを作成します。

TACACS+ サーバ用の Per-VRF の設定

TACACS+ サーバ機能の per-VRF は TACACS+ サーバの per- 仮想単位ルート転送（per-VRF）の認証、認可、アカウンティング（AAA）を設定することができます。

VRF ルーティング テーブル（ステップ 3 および 4 で示すように）を作成し、インターフェイスを設定する（ステップ 6、7、および 8）ことができます。TACACS+ サーバの per-VRF 単位の実際の設定は、ステップ 10～13 で行われます。

始める前に

TACACS+ サーバの per-VRF を設定する前に、AAA およびサーバ グループを設定しておく必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**vrf definition** *vrf-name* 例： `Device(config)# vrf definition vrf-name`	VRF テーブルを設定し、VRF コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 4	**rd** *route-distinguisher* 例： `Device(config-vrf)# rd route-distinguisher`	VRF インスタンスに対するルーティングおよびフォワーディングテーブルを作成します。
ステップ 5	**exit** 例： `Device(config-vrf)# exit`	VRF コンフィギュレーション モードを終了します。
ステップ 6	**interface** *interface-name* 例： `Device(config)# interface interface-name`	インターフェイスを設定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 7	**vrf forwarding** *vrf-name* 例： `Device(config-if)# vrf forwarding vrf-name`	インターフェイスに VRF を設定します。
ステップ 8	**ip address** *ip-address mask* [**secondary**] 例： `Device(config-if)# ip address ip-address mask [secondary]`	インターフェイスに対するプライマリ IP アドレスまたはセカンダリ IP アドレスを設定します。
ステップ 9	**exit** 例： `Device(config-vrf)# exit`	インターフェイス コンフィギュレーション モードを終了します。
ステップ 10	**aaa group server tacacs+** *group-name* 例： `Device(config)# aaa group server tacacs+ tacacs1`	異なる TACACS+ サーバ ホストを別々のリストと方式にグループ化し、server-group コンフィギュレーション モードを開始します。
ステップ 11	**server-private** {*ip-address* | *name*} [**nat**] [**single-connection**] [**port** *port-number*] [**timeout** *seconds*] [**key** [**0** | **7**] *string*] 例： `Device(config-sg-tacacs+)# server-private 10.1.1.1 port 19 key cisco`	グループ サーバに対するプライベート TACACS+ サーバの IP アドレスを設定します。
ステップ 12	**vrf forwarding** *vrf-name* 例： `Device(config-sg-tacacs+)# vrf forwarding vrf-name`	AAA TACACS+ サーバ グループの VRF リファレンスを設定します。

	コマンドまたはアクション	目的
ステップ 13	**ip tacacs source-interface** *subinterface-name* 例： `Device(config-sg-tacacs+)# ip tacacs source-interface subinterface-name`	すべての発信 TACACS+ パケットに対して、指定されたインターフェイスの IP アドレスを使用します。
ステップ 14	**exit** 例： `Device(config-sg-tacacs)# exit`	server-group コンフィギュレーション モードを終了します。

例

次の例で、per-VRF TACACS+ の設定に必要なすべての手順をリストします。

```
Device> enable
Device# configure terminal
Device(config)# vrf definition cisco
Device(config-vrf)# rd 100:1
Device(config-vrf)# exit
Device(config)# interface Loopback0
Device(config-if)# vrf forwarding cisco
Device(config-if)# ip address 10.0.0.2 255.0.0.0
Device(config-if)# exit
Device(config-sg-tacacs+)# vrf forwarding cisco
Device(config-sg-tacacs+)# ip tacacs source-interface Loopback0
Device(config-sg-tacacs)# exit
```

マルチキャスト VRF の設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**ip routing** 例： `Device(config)# ip routing`	IP ルーティングをイネーブルにします。
ステップ 3	**vrf definition** *vrf-name* 例： `Device(config)# vrf definition vrf-name`	VRF テーブルを設定し、VRF コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 4	**ip multicast-routing vrf** *vrf-name* 例： `Device(config-vrf)# ip multicast-routing vrf vrf-name`	（任意）VRF テーブルでグローバル マルチキャスト ルーティングをイネーブルにします。
ステップ 5	**rd** *route-distinguisher* 例： `Device(config-vrf)# rd route-distinguisher`	ルート識別子を指定して VRF テーブルを作成します。自律システム（AS）番号および任意の数（xxx:y）または IP アドレスおよび任意の数（A.B.C.D:y）のどちらかを入力します。
ステップ 6	**route-target** {**export** | **import** | **both**} *route-target-ext-community* 例： `Device(config-vrf)# route-target {export | import | both} route-target-ext-community`	指定された VRF のインポート、エクスポート、またはインポートおよびエクスポート ルート ターゲット コミュニティのリストを作成します。AS システム番号と任意の番号（xxx:y）または IP アドレスと任意の番号（A.B.C.D:y）を入力します。 ルート ターゲット ext コミュニティ値は、ステップ 4 で入力した route-distinguisher 値と同じです。
ステップ 7	**import map** ルート マップ 例： `Device(config-vrf)# import map route-map`	（任意）VRF にルート マップを対応付けます。
ステップ 8	**interface** *interface-id* 例： `Device(config)# interface interface-id`	インターフェイス コンフィギュレーション モードを開始して、VRF に対応付けるレイヤ 3 インターフェイスを指定します。有効なインターフェイスは、ルーテッド ポートまたは SVI です。
ステップ 9	**vrf forwarding** *vrf-name* 例： `Device(config-if)# vrf forwarding vrf-name`	VRF をレイヤ 3 インターフェイスに対応付けます。
ステップ 10	**ip address** *ip-address***mask** 例： `Device(config-if)# ip address ip-address mask`	レイヤ 3 インターフェイスの IP アドレスを設定します。

	コマンドまたはアクション	目的
ステップ 11	**ip pim sparse-mode** 例： `Device(config-if)# ip pim sparse-mode`	VRF に関連付けられているレイヤ 3 インターフェイス上で、PIM をイネーブルにします。
ステップ 12	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。
ステップ 13	**show vrf definition** [**brief** | **detail** | **interfaces**] [*vrf-name*] 例： `Device# show vrf definition brief`	設定を確認します。設定した VRF に関する情報を表示します。
ステップ 14	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーションファイルに設定を保存します。

例

次に、VRF テーブル内にマルチキャストを設定する例を示します。

```
Device(config)# ip routing
Device(config)# vrf definition multiVrfA
Device(config-vrf)# ip multicast-routing vrf multiVrfA
Device(config-vrf)# interface GigabitEthernet3/1/0
Device(config-if)# vrf forwarding multiVrfA
Device(config-if)# ip address 172.21.200.203 255.255.255.0
Device(config-if)# ip pim sparse-mode
```

IPv4 VRF の設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**ip routing** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	**vrf definition** *vrf-name* 例： `Device(config)# vrf definition vrf-name`	VRF 名を指定し、VRF コンフィギュレーション モードを開始します。
ステップ 4	**rd** *route-distinguisher* 例： `Device(config-vrf)# rd route-distinguisher`	ルート識別子を指定して VRF テーブルを作成します。自律システム番号と任意の数値（xxx:y）、または IP アドレスと任意の数値（A.B.C.D:y）のいずれかを入力します。
ステップ 5	**route-target** {**export** | **import** | **both**} *route-target-ext-community* 例： `Device(config-vrf)# route-target {export | import | both} route-target-ext-community`	指定された VRF のインポート、エクスポート、またはインポートおよびエクスポート ルート ターゲット コミュニティのリストを作成します。AS システム番号と任意の番号（xxx:y）または IP アドレスと任意の番号（A.B.C.D:y）を入力します。
ステップ 6	**import map** ルート マップ 例： `Device(config-vrf)# import map route-map`	（任意）VRF にルートマップを対応付けます。
ステップ 7	**interface** *interface-id* 例： `Device(config-vrf)# interface interface-id`	インターフェイス コンフィギュレーション モードを開始して、VRF に対応付けるレイヤ 3 インターフェイスを指定します。インターフェイスにはルーテッド ポートまたは SVI を設定できます。
ステップ 8	**vrf forwarding** *vrf-name* 例： `Device(config-if)# vrf forwarding vrf-name`	VRF をレイヤ 3 インターフェイスに対応付けます。
ステップ 9	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。
ステップ 10	**show vrf definition** [**brief** | **detail** | **interfaces**] [*vrf-name*] 例： `Device# show vfr definition [brief | detail | interfaces] [vrf-name]`	設定を確認します。設定した VRF に関する情報を表示します。

	コマンドまたはアクション	目的
ステップ 11	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーションファイルに設定を保存します。 VRF とそのすべてのインターフェイスを削除するには、**no vrf definition***vrf-name* グローバル コンフィギュレーション コマンドを使用します。VRF からインターフェイスを削除するには、**no vrf forwarding** インターフェイス コンフィギュレーション コマンドを使用します。

IPv6 用の VRF-Lite の設定

ここでは、IPv6 用の VRF-Lite の設定について説明します。

VRF 認識サービスの設定

IPv6 サービスは、グローバルなインターフェイス上と、グローバルなルーティング インスタンス内で設定できます。IPv6 サービスは複数のルーティング インスタンス上で稼働するように拡張されます。これが、VRF 認識です。システム内の任意の設定済み VRF であればいずれも、VRF 認識サービス用に指定できます。

VRF 認識サービスは、プラットフォームから独立したモジュールに実装されています。VRF は、Cisco IOS 内の複数のルーティング インスタンスを提供します。各プラットフォームには、サポートする VRF 数に関して独自の制限があります。

VRF 認識サービスには、次の特性があります。

- ユーザは、ユーザ指定の VRF 内のホストに ping を実行できます。
- ネイバー探索エントリは、個別の VRF で学習されます。ユーザは、特定の VRF のネイバー探索（ND）エントリを表示できます。

次のサービスは VRF 認識です。

- Ping
- ユニキャスト RPF（uRPF）
- traceroute
- FTP および TFTP
- [Telnet および SSH（Telnet and SSH）]
- NTP

PING のユーザ インターフェイスの設定

VRF 認識 ping を設定するには、次の作業を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**ping vrf** *vrf-name* **ipv6-host** 例： `Device# ping vrf vrf-name ipv6-host`	指定された VRF で、IPv6 ホストまたはアドレスに対して ping を実行します。

uRPF のユーザ インターフェイスの設定

VRF に割り当てられているインターフェイス上で、uRPF を設定できます。送信元の検索が VRF テーブルで実行されます。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *interface-id* 例： `Device(config)# interface interface-id`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 3	**no switchport** 例： `Device(config-if)# no switchport`	レイヤ 2 コンフィギュレーション モードからインターフェイスを削除します（物理インターフェイスの場合）。
ステップ 4	**vrf forwarding** *vrf-name* 例： `Device(config-if)# vrf forwarding vrf-name`	インターフェイス上で VRF を設定します。
ステップ 5	**ipv6 address** *ip-address***subnet-mask** 例： `Device(config-if)# ip address ip-address mask`	インターフェイスの IPv6 アドレスを入力します。
ステップ 6	**ipv6 verify unicast source reachable-via rx allow-default** 例：	インターフェイス上で uRPF 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)# ipv6 verify unicast source reachable-via rx allow-default`	
ステップ 7	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。

Traceroute のユーザ インターフェイスの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**traceroute vrf** *vrf-name ipv6address* 例： `Device# traceroute vrf vrf-name ipv6address`	宛先アドレスを取得する VPN VRF の名前を指定します。

Telnet および SSH のユーザ インターフェイスの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**telnet** *ipv6-address*/ **vrf** *vrf-name* 例： `Device# telnet ipv6-address/vrf vrf-name`	指定された VRF で、IPv6 ホストまたはアドレスに Telnet 経由で接続します。
ステップ 2	**ssh -l** *username* **-vrf** *vrf-name* **ipv6-host** 例： `Device# ssh -l username -vrf vrf-name ipv6-host`	指定された VRF で、IPv6 ホストまたはアドレスに SSH 経由で接続します。

NTP のユーザ インターフェイスの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 2	**ntp server vrf** *vrf-name* **ipv6-host** 例： `Device(config)# ntp server vrf vrf-name ipv6-host`	指定された VRF で NTP サーバを設定します。
ステップ 3	**ntp peer vrf** *vrf-name* **ipv6-host** 例： `Device(config)# ntp peer vrf vrf-name ipv6-host`	指定された VRF で NTP ピアを設定します。

IPv6 VRF の設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**vrf definition** *vrf-name* 例： `Device(config)# vrf definition vrf-name`	VRF 名を指定し、VRF コンフィギュレーション モードを開始します。
ステップ 3	**rd** *route-distinguisher* 例： `Device(config-vrf)# rd route-distinguisher`	（任意）ルート識別子を指定して VRF テーブルを作成します。自律システム番号および任意の数（xxx:y）、または IP アドレスおよび任意の数（A.B.C.D:y）のいずれかを入力します。
ステップ 4	**address-family** *ipv4* | *ipv6* 例： `Device(config-vrf)# address-family ipv4 | ipv6`	（任意）デフォルトは IPv4 です。IPv6 の必須設定。
ステップ 5	**route-target** {**export** | **import** | **both**} *route-target-ext-community* 例： `Device(config-vrf)# route-target {export | import | both} route-target-ext-community`	指定された VRF のインポート、エクスポート、またはインポートおよびエクスポート ルート ターゲット コミュニティのリストを作成します。AS システム番号と任意の番号（xxx:y）または IP アドレスと任意の番号（A.B.C.D:y）を入力します。

	コマンドまたはアクション	目的
		（注） このコマンドは、BGPが動作している場合にのみ有効です。
ステップ 6	**exit-address-family** 例： `Device(config-vrf)# exit-address-family`	VRF アドレス ファミリ コンフィギュレーションモードを終了し、VRF コンフィギュレーション モードに戻ります。
ステップ 7	**vrf definition** *vrf-name* 例： `Device(config)# vrf definition vrf-name`	VRF コンフィギュレーションモードを開始します。
ステップ 8	**ipv6 multicast multitopology** 例： `Device(config-vrf-af)# ipv6 multicast multitopology`	マルチキャスト固有の RPF トポロジを有効にします。
ステップ 9	**address-family ipv6 multicast** 例： `Device(config-vrf)# address-family ipv6 multicast`	マルチキャスト IPv6 アドレス ファミリを入力します。
ステップ 10	**end** 例： `Device(config-vrf-af)# end`	特権 EXEC モードに戻ります。

例

次に、VRF を設定する例を示します。

```
Device(config)#  vrf definition red
Device(config-vrf)# rd 100:1
Device(config-vrf)# address family ipv6
Device(config-vrf-af)# route-target both 200:1
Device(config-vrf)# exit-address-family
Device(config-vrf)# vrf definition red
Device(config-vrf)# ipv6 multicast multitopology
Device(config-vrf)# address-family ipv6 multicast
Device(config-vrf-af)# end
```

定義済み VRF へのインターフェイスの関連付け

手順

	コマンドまたはアクション	目的
ステップ 1	**interface** *interface-id* 例： `Device(config-vrf)# interface interface-id`	インターフェイスコンフィギュレーション モードを開始して、VRF に対応付けるレイヤ3インターフェイスを指定します。インターフェイスにはルーテッド ポートまたは SVI を設定できます。
ステップ 2	**no switchport** 例： `Device(config-if)# no switchport`	コンフィギュレーション モードからインターフェイスを削除します（物理インターフェイスの場合）。
ステップ 3	**vrf forwarding** *vrf-name* 例： `Device(config-if)# vrf forwarding vrf-name`	VRF をレイヤ 3 インターフェイスに対応付けます。
ステップ 4	**ipv6 enable** 例： `Device(config-if)# ipv6 enable`	インターフェイスで IPv6 をイネーブルにします。
ステップ 5	**ipv6 address** *ip-address subnet-mask* 例： `Device(config-if)# ipv6 address ip-address subnet-mask`	インターフェイスの IPv6 アドレスを入力します。
ステップ 6	**show ipv6 vrf** [**brief** | **detail** | **interfaces**] [*vrf-name*] 例： `Device# show ipv6 vrf [brief | detail | interfaces] [vrf-name]`	設定を確認します。設定した VRF に関する情報を表示します。
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

例

次に、インターフェイスを VRF に関連付ける例を示します。

```
Switch(config-vrf)# interface ethernet0/1
Switch(config-if)# vrf forwarding red
```

```
Switch(config-if)# ipv6 enable
Switch(config-if)# ipv6 address 5000::72B/64
```

ルーティング プロトコル経由での VRF へのルートの入力

ここでは、ルーティングプロトコル経由での VRF へのルートの入力について説明します。

VRF スタティック ルートの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**ipv6 route** [**vrf** *vrf-name*] *ipv6-prefix*/*prefix-length* {*ipv6-address* | **interface-type** *interface-number* [*ipv6-address*]} 例： `Device(config)# ipv6 route [vrf vrf-name] ipv6-prefix/prefix-length {ipv6-address | interface-type interface-number [ipv6-address]}`	VRF に固有のスタティック ルートを設定します。

例

```
Device(config)# ipv6 route vrf v6a 7000::/64 TenGigabitEthernet32 4000::2
```

OSPFv3 ルータ プロセスの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**router ospfv3** *process-id* 例： `Device(config)# router ospfv3 process-id`	IPv6 アドレス ファミリの OSPFv3 ルータ コンフィギュレーション モードを有効にします。

	コマンドまたはアクション	目的
ステップ 3	**area** *area-ID* [**default-cot** | **nssa** | **stub**] 例： `Device(config-router)# area area-ID [default-cot | nssa | stub]`	OSPFv3 エリアを設定します。
ステップ 4	**router-id** *router-id* 例： `Device(config-router)# router-id router-id`	固定ルータ ID を使用します。
ステップ 5	**address-family ipv6 unicast vrf** *vrf-name* 例： `Device(config-router)# address-family ipv6 unicast vrf vrf-name`	vrf vrf-name の OSPFv3 の IPv6 アドレス ファミリ コンフィギュレーション モードを開始します。
ステップ 6	**redistribute source-protocol** [*process-id*] **options** 例： `Device(config-router)# redistribute source-protocol [process-id] options`	あるルーティング ドメインから別のルーティング ドメインへ IPv6 ルートを再配布します。
ステップ 7	**end** 例： `Device(config-router)# end`	特権 EXEC モードに戻ります。

例

次に、OSPFv3 ルータ プロセスを設定する例を示します。

```
Device(config-router)# router ospfv3 1
Device(config-router)# router-id 1.1.1.1
Device(config-router)# address-family ipv6 unicast
Device(config-router-af)# exit-address-family
```

インターフェイス上での OSPFv3 のイネーブル化

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 2	**interface** *type-number* 例： `Device(config-vrf)# interface type-number`	インターフェイスのタイプと番号を指定し、スイッチをインターフェイス コンフィギュレーション モードにします。
ステップ 3	**ospfv3** *process-id* **area** *area-id* **ipv6** [**instance** *instance-id*] 例： `Device(config-if)# ospfv3 process-id area area-ID ipv6 [instance instance-id]`	IPv6 AF を設定したインターフェイスで OSPFv3 を有効にします。
ステップ 4	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。

例

次に、インターフェイス上で OSPFv3 を有効にする例を示します。

```
Device(config)# interface GigabitEthernet2/1
Device(config-if)# no switchport
Device(config-if)# ipv6 address 4000::2/64
Device(config-if)# ipv6 enable
Device(config-if)# ipv6 ospf 1 area 0
Device(config-if)# end
```

EIGRPv6 ルーティング プロセスの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**router eigrp** *virtual-instance-name* 例： `Device(config)# router eigrp virtual-instance-name`	EIGRP ルーティング プロセスを設定し、ルータ コンフィギュレーション モードを開始します。
ステップ 3	**address-family ipv6 vrf** *vrf-name* **autonomous-system** *autonomous-system-number* 例：	EIGRP IPv6 VRF-Lite を有効にし、アドレス ファミリ コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
	`Device(config-router)# address-family ipv6 vrf vrf-name autonomous-system autonomous-system-number`	
ステップ 4	**topology** {**base** | **topology-name tid number** 例： `Device(config-router-af)# topology {base | topology-name tid number`	指定されたトポロジ インスタンスで IP トラフィックをルーティングするよう EIGRP プロセスを設定し、アドレス ファミリ トポロジ コンフィギュレーション モードを開始します。
ステップ 5	**exit-aftopology** 例： `Device(config-router-af-topology)# exit-aftopology`	アドレス ファミリ トポロジ コンフィギュレーション モードを終了します。
ステップ 6	**eigrp router-id** *ip-address* 例： `Device(config-router)# eigrp router-id ip-address`	固定ルータ ID の使用を有効にします。
ステップ 7	end 例： `Device(config-router)# end`	ルータ コンフィギュレーション モードを終了します。

例

次に、EIGRP ルーティング プロセスを設定する例を示します。

```
Device(config)# router eigrp test
Device(config-router)# address-family ipv6 unicast vrf b1 autonomous-system 10
Device(config-router-af)# topology base
Device(config-router-af-topology)# exit-af-topology
Device(config-router)# eigrp router-id 2.3.4.5
Device(config-router)# exit-address-family
```

VRF-Lite に関する追加情報

ここでは、VRF-Lite に関する追加情報を提供します。

IPv4 と IPv6 間での VPN の共存

IPv4 を設定するための「以前の」CLI と、IPv6 用の「新しい」CLI 間には下位互換性があります。つまり、設定に両方の CLI を含めることができます。IPv4 CLI は、同じインターフェイス

上で、VRF 内で定義されている IP アドレスとともにグローバル ルーティング テーブルで定義されている IPv6 アドレスも備える機能を保持しています。

次に例を示します。

```
vrf definition red
 rd 100:1
 address family ipv6
 route-target both 200:1
 exit-address-family
!
vrf definition blue
 rd 200:1
 route-target both 200:1
!
interface Ethernet0/0
 vrf forwarding red
 ip address 50.1.1.2 255.255.255.0
 ipv6 address 4000::72B/64
!
interface Ethernet0/1
 vrf forwarding blue
 ip address 60.1.1.2 255.255.255.0
 ipv6 address 5000::72B/64
```

この例では、Ethernet0/0 用に定義されたすべてのアドレス（v4 と v6）が VRF red を参照します。Ethernet0/1 については、IP アドレスは VRF blue を参照しますが、ipv6 アドレスはグローバル IPv6 アドレス ルーティング テーブルを参照します。

VRF-Lite 設定の確認

ここでは、VRF-Lite 設定を確認する手順について説明します。

IPv4 VRF-Lite ステータスの表示

VRF-Liteの設定およびステータスに関する情報を表示するには、次の作業のいずれかを行います。

コマンド	目的
Device# **show ip protocols vrf** *vrf-name*	VRF に対応付けられたルーティング プロトコル情報を表示します。
Device# **show ip route vrf** *vrf-name* [**connected**] [*protocol* [*as-number*]] [**list**] [**mobile**] [**odr**] [**profile**] [**static**] [**summary**][**supernets-only**]	VRF に対応付けられた IP ルーティング テーブル情報を表示します。
Device# **show vrf definition** [**brief** | **detail** | **interfaces**] [*vrf-name*]	定義された VRF インスタンスに関する情報を表示します。

コマンド	目的
Device# **bidir vrf** *instance-name a.b.c.d* | **active** | **bidriectional**| **count** | **interface** | **proxy** | **pruned** | **sparse** | **ssm** | **static** | **summary**	定義された VRF インスタンスに関する情報を表示します。

次に、VRF インスタンス内のマルチキャスト ルート テーブル情報を表示する例を示します。

```
Switch# show ip mroute 226.0.0.2
IP Multicast Routing Table
Flags: S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
       N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
       Q - Received BGP S-A Route, q - Sent BGP S-A Route,
       V - RD & Vector, v - Vector, p - PIM Joins on route,
       x - VxLAN group, c - PFP-SA cache created entry
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 226.0.0.2), 00:01:17/stopped, RP 1.11.1.1, flags: SJCF
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Vlan100, Forward/Sparse, 00:01:17/00:02:36

(5.0.0.11, 226.0.0.2), 00:01:17/00:01:42, flags: FT
  Incoming interface: Vlan5, RPF nbr 0.0.0.0
  Outgoing interface list:
    Vlan100, Forward/Sparse, 00:01:17/00:02:36
```

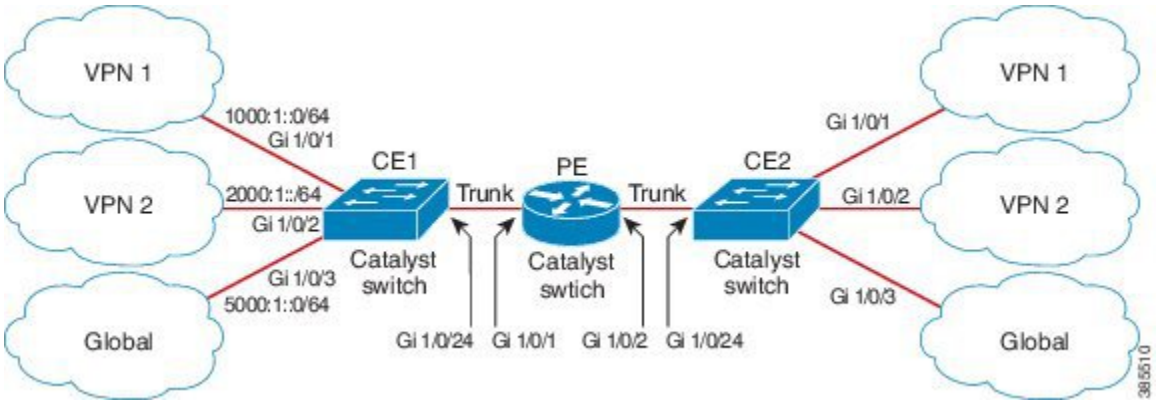
VRF-Lite の設定例

ここでは、VRF-Lite の設定例を示します。

IPv6 VRF-Lite の設定例

次に、CE-PE ルーティングに OSPFv3 を使用するトポロジを示します。

図 *6 : VRF-Lite* の設定例



CE1 スイッチの設定

```
ipv6 unicast-routing
vrf definition v1
 rd 100:1
 !
address-family ipv6
 exit-address-family
!

vrf definition v2
 rd 200:1
 !
address-family ipv6
 exit-address-family
!

interface Vlan100
 vrf forwarding v1
 ipv6 address 1000:1::1/64
 ospfv3 100 ipv6 area 0
!

interface Vlan200
 vrf forwarding v2
 ipv6 address 2000:1::1/64
 ospfv3 200 ipv6 area 0
!

interface GigabitEthernet 1/0/1
switchport access vlan 100
end

interface GigabitEthernet 1/0/2
switchport access vlan 200
end

interface GigabitEthernet 1/0/24
 switchport trunk encapsulation dot1q

switchport mode trunk
end

router ospfv3 100
 router-id 10.10.10.10
```

```
 !
 address-family ipv6 unicast vrf v1
  redistribute connected
  area 0 normal
 exit-address-family
!

router ospfv3 200
 router-id 20.20.20.20
 !
 address-family ipv6 unicast vrf v2
  redistribute connected
  area 0 normal
 exit-address-family
!
```

PE スイッチの設定

```
ipv6 unicast-routing

vrf definition v1
 rd 100:1
 !
address-family ipv6
 exit-address-family
!

vrf definition v2
 rd 200:1
 !
address-family ipv6
 exit-address-family
!
interface Vlan600
 vrf forwarding v1
 no ipv6 address
 ipv6 address 1000:1::2/64
 ospfv3 100 ipv6 area 0
!

interface Vlan700
 vrf forwarding v2
 no ipv6 address
 ipv6 address 2000:1::2/64
 ospfv3 200 ipv6 area 0
!

interface Vlan800
 vrf forwarding v1
 ipv6 address 3000:1::7/64
 ospfv3 100 ipv6 area 0
!
interface Vlan900
 vrf forwarding v2
 ipv6 address 4000:1::7/64
 ospfv3 200 ipv6 area 0
!

interface GigabitEthernet 1/0/1
 switchport trunk encapsulation dot1q
 switchport mode trunk
 exit

interface GigabitEthernet 1/0/2
```

```
 switchport trunk encapsulation dot1q

switchport mode trunk
 exit

router ospfv3 100
 router-id 30.30.30.30
 !
 address-family ipv6 unicast vrf v1
  redistribute connected
  area 0 normal
 exit-address-family
 !
address-family ipv6 unicast vrf v2
  redistribute connected
  area 0 normal
 exit-address-family
 !
```

CE2 スイッチの設定

```
ipv6 unicast-routing

vrf definition v1
 rd 100:1
 !
address-family ipv6
 exit-address-family
!

vrf definition v2
 rd 200:1
 !
address-family ipv6
 exit-address-family
!

interface Vlan100
 vrf forwarding v1

ipv6 address 1000:1::3/64
 ospfv3 100 ipv6 area 0
!

interface Vlan200
 vrf forwarding v2
 ipv6 address 2000:1::3/64
 ospfv3 200 ipv6 area 0
!

interface GigabitEthernet 1/0/1
switchport access vlan 100
end

interface GigabitEthernet 1/0/2
switchport access vlan 200
end

interface GigabitEthernet 1/0/24
 switchport trunk encapsulation dot1q
 switchport mode trunk
 end

router ospfv3 100
```

```
 router-id 40.40.40.40
 !
 address-family ipv6 unicast vrf v1
  redistribute connected
  area 0 normal
 exit-address-family
!

router ospfv3 200
 router-id 50.50.50.50
 !
 address-family ipv6 unicast vrf v2
  redistribute connected

area 0 normal
 exit-address-family
!
```

VRF-Lite に関するその他の参考資料

関連資料

関連項目	マニュアル タイトル
この章で使用するコマンドの完全な構文および使用方法の詳細。	の「IP マルチキャスト ルーティングのコマンド」の項を参照してください。*Command Reference (Catalyst 9200 Series Switches)*

標準および RFC

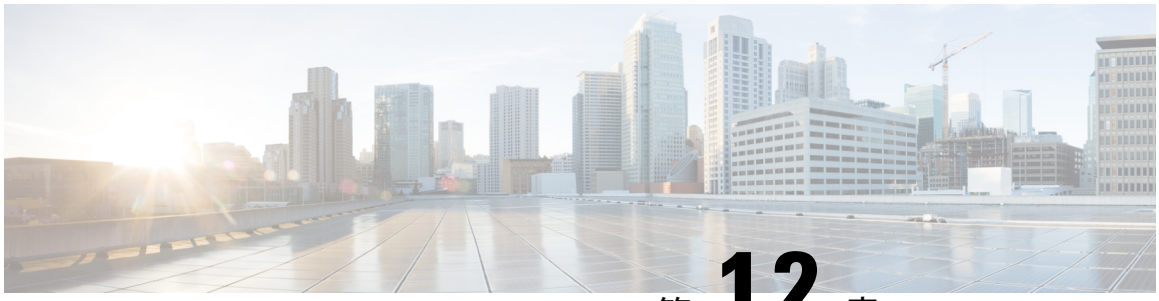
標準/RFC	タイトル
RFC 6763	『*DNS-Based Service Discovery*』
マルチキャスト DNS インターネット（ドラフト）	マルチキャスト

マルチキャスト VRF-Lite の機能履歴と情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

機能名	リリース	機能情報
VRF-Lite を使用した IPv6 マルチキャストのサポート	Cisco IOS XE Everest 16.6.1	IPv6 VRF-Lite によって、サービスプロバイダーは 1 つのインターフェイスを使用して、重複する IP アドレスを持つ複数の VPN をサポートできます。



第 12 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 プロバイダー ネットワークに接続され、サービス プロバイダーは、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 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 プロバイダーは、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 プロバイダーが複数の 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。

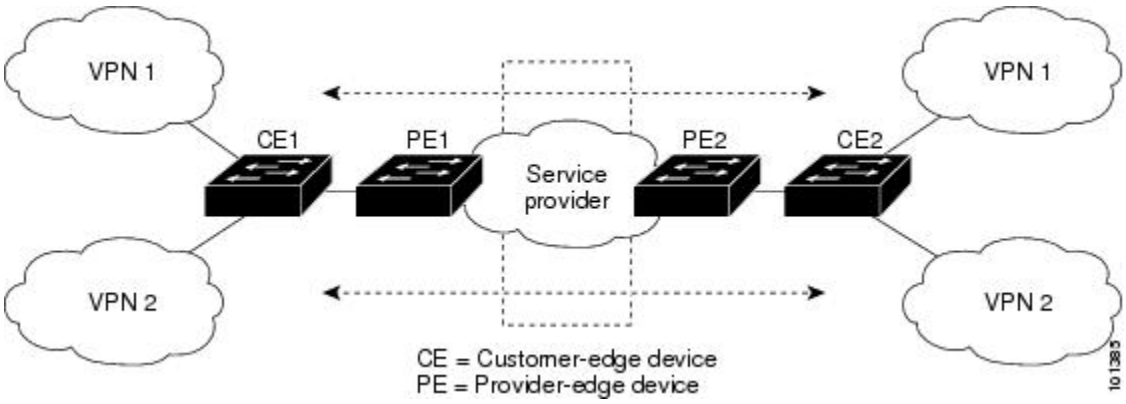
- お客様は、CE デバイスにより、1 つまたは複数のプロバイダー エッジ（PE）ルータへのデータ リンクを介してサービス プロバイダー ネットワークにアクセスできます。CE デバイスは、サイトのローカル ルートをルータにアドバタイズし、リモート VPN ルートをそこから学習します。スイッチを CE に設定することができます。
- CE デバイスに接続していないサービス プロバイダー ネットワークのルータは、プロバイダー ルータやコア ルータになります。

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。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 転送：VPN サービス プロバイダー 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。

- ユーザは、ユーザ指定の 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 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。
- Cisco Catalyst 9200 シリーズ スイッチの各モデルでサポートされる VRF の数は次のとおりです。

スイッチ モデル	サポートされる VRF の数
C9200L-24T-4G	1
C9200L-24P-4G	
C9200L-48T-4G	
C9200L-48P-4G	
C9200L-24T-4X	
C9200L-24P-4X	
C9200L-48T-4X	
C9200L-48P-4X	
C9200-24T	4
C9200-24P	
C9200-48T	
C9200-48P	

スイッチ モデル	サポートされる VRF の数
C9200-24PB	32
C9200-48PB	

- Multi-VRF CE は、パケットのスイッチング レートに影響しません。
- VPN マルチキャストはサポートされません。
- プライベート VLAN で VRF をイネーブルにできます（逆も同様です）。
- インターフェイスでポリシーベース ルーティング（PBR）がイネーブルになっている場合は、VRF をイネーブルにできません（逆も同様です）。
- インターフェイスで Web Cache Communication Protocol（WCCP）がイネーブルになっている場合は、VRF をイネーブルにできません（逆も同様です）。

VRF の設定

次の操作を行ってください。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip routing** 例： `Device(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	**ip vrf** *vrf-name* 例： `Device(config)#ip vrf vpn1`	VRF 名を指定し、VRF コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 5	**rd** *route-distinguisher* 例： `Device(config-vrf)#rd 100:2`	ルート識別子を指定してVRFテーブルを作成します。AS 番号と任意の番号（xxx:y）またはIP アドレスと任意の番号（A.B.C.D:y）を入力します。
ステップ 6	**route-target** {**export** | **import** | **both**} *route-target-ext-community* 例： `Device(config-vrf)#route-target both 100:2`	指定されたVRFのインポート、エクスポート、またはインポートおよびエクスポート ルート ターゲット コミュニティのリストを作成します。ASシステム番号と任意の番号（xxx:y）またはIPアドレスと任意の番号（A.B.C.D:y）を入力します。*route-target-ext-community* は、ステップ 4 で入力した *route-distinguisher* と同一にする必要があります。
ステップ 7	**import map** *route-map* 例： `Device(config-vrf)#import map importmap1`	（任意）VRFにルートマップを対応付けます。
ステップ 8	**interface** *interface-id* 例： `Device(config-vrf)#interface gigabitethernet 1/0/1`	VRF に関連付けるレイヤ 3 インターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。インターフェイスにはルーテッドポートまたはSVIを設定できます。
ステップ 9	**ip vrf forwarding** *vrf-name* 例： `Device(config-if)#ip vrf forwarding vpn1`	VRF をレイヤ 3 インターフェイスに対応付けます。 （注） **ip vrf forwarding**が管理インターフェイスで有効になっている場合、アクセスポイントは加入しません。
ステップ 10	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 11	**show ip vrf** [**brief** | **detail** | **interfaces**] [*vrf-name*] 例：	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。

	コマンドまたはアクション	目的
	`Device#show ip vrf interfaces vpn1`	
ステップ 12	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーションファイルに設定を保存します。

Multi-VRF CE の設定方法

マルチキャスト VRF の設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip routing** 例： `Device(config)#ip routing`	IPルーティング モードをイネーブルにします
ステップ 4	**ip vrf** *vrf-name* 例： `Device(config)#ip vrf vpn1`	VRF 名を指定し、VRF コンフィギュレーション モードを開始します。
ステップ 5	**rd** *route-distinguisher* 例： `Device(config-vrf)#rd 100:2`	ルート識別子を指定してVRFテーブルを作成します。AS 番号と任意の番号（xxx:y）または IP アドレスと任意の番号（A.B.C.D:y）を入力します。

	コマンドまたはアクション	目的
ステップ 6	**route-target** {**export** | **import** | **both**} *route-target-ext-community* 例： `Device(config-vrf)#route-target import 100:2`	指定された VRF のインポート、エクスポート、またはインポートおよびエクスポート ルート ターゲット コミュニティのリストを作成します。AS システム番号と任意の番号（xxx:y）または IP アドレスと任意の番号（A.B.C.D:y）を入力します。*route-target-ext-community* は、ステップ 4 で入力した *route-distinguisher* と同一にする必要があります。
ステップ 7	**import map** *route-map* 例： `Device(config-vrf)#import map importmap1`	（任意）VRF にルート マップを対応付けます。
ステップ 8	**ip multicast-routing vrf** *vrf-name* **distributed** 例： `Device(config-vrf)#ip multicast-routing vrf vpn1 distributed`	（任意）VRF テーブルでグローバル マルチキャスト ルーティングをイネーブルにします。
ステップ 9	**interface** *interface-id* 例： `Device(config-vrf)#interface gigabitethernet 1/0/2`	VRF に関連付けるレイヤ 3 インターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。インターフェイスはルーテッド ポートまたは SVI に設定できます。
ステップ 10	**ip vrf forwarding** *vrf-name* 例： `Device(config-if)#ip vrf forwarding vpn1`	VRF をレイヤ 3 インターフェイスに対応付けます。
ステップ 11	**ip address** *ip-address* **mask** 例： `Device(config-if)#ip address 10.1.5.1 255.255.255.0`	レイヤ 3 インターフェイスの IP アドレスを設定します。

	コマンドまたはアクション	目的
ステップ 12	**ip pim sparse-dense mode** 例： `Device(config-if)#ip pim sparse-dense mode`	VRF に関連付けられているレイヤ 3 インターフェイス上で、PIM をイネーブルにします。
ステップ 13	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 14	**show ip vrf** [**brief** | **detail** | **interfaces**] [*vrf-name*] 例： `Device#show ip vrf detail vpn1`	設定を確認します。設定した VRF に関する情報を表示します。
ステップ 15	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

VPN ルーティング セッションの設定

VPN 内のルーティングは、サポートされている任意のルーティングプロトコル（RIP、OSPF、EIGRP、）、またはスタティックルーティングで設定できます。ここで説明する設定は OSPF のものですが、その他のプロトコルでも手順は同じです。

（注） VRF インスタンス内で EIGRP ルーティングプロセスが実行されるように設定するには、**autonomous-system** *autonomous-system-number* アドレス ファミリ コンフィギュレーション モード コマンドを入力して、自律システム番号を設定する必要があります。

手順

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

	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 ospf** *process-id* **vrf** *vrf-name* 例： `Device(config)#router ospf 1 vrf vpn1`	OSPF ルーティングをイネーブルにして VPN 転送テーブルを指定し、ルータ コンフィギュレーション モードを開始します。
ステップ 4	**log-adjacency-changes** 例： `Device(config-router)#log-adjacency-changes`	（任意）隣接ステートの変更を記録します。これは、デフォルトの状態です。
ステップ 5	**redistribute isis subnets** 例： `Device(config-router)#redistribute isis 10 subnets`	ISIS ネットワークから OSPF ネットワークに情報を再配布するようにスイッチを設定します。
ステップ 6	**network** *network-number* **area** *area-id* 例： `Device(config-router)#network 1 area 2`	OSPF が動作するネットワーク アドレスとマスク、およびそのネットワーク アドレスのエリア ID を定義します。
ステップ 7	**end** 例： `Device(config-router)#end`	特権 EXEC モードに戻ります。
ステップ 8	**show ip ospf** *process-id* 例： `Device#show ip ospf 1`	OSPF ネットワークの設定を確認します。
ステップ 9	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

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

コマンド	目的
show ip protocols vrf *vrf-name*	VRF に対応付けられたルーティング プロトコル情報を表示します。
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 ルーティング テーブル情報を表示します。
show ip vrf [**brief** | **detail** | **interfaces**] [*vrf-name*]	定義された VRF インスタンスに関する情報を表示します。

VRF 認識サービスの設定

次のサービスは、VRF 認識です。

- ARP
- ping
- 簡易ネットワーク管理プロトコル（SNMP）
- ユニキャスト RPF（uRPF）
- Syslog
- traceroute
- FTP および TFTP

ARP 用 VRF 認識サービスの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**show ip arp vrf** *vrf-name* 例： `Device#show ip arp vrf vpn1`	指定された VRF 内の ARP テーブルを表示します。

ping 用 VRF 認識サービスの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**ping vrf***vrf-name***ip-host** 例： `Device#ping vrf vpn1 ip-host`	指定された VRF 内の ARP テーブルを表示します。

SNMP 用 VRF 認識サービスの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**snmp-server trap authentication vrf** 例： `Device(config)#snmp-server trap authentication vrf`	VRF で、パケットに対して SNMP トラップをイネーブルにします。
ステップ 4	**snmp-server engineID remote** *host* **vrf** *vpn-instance engine-id string* 例： `Device(config)#snmp-server engineID remote 172.16.20.3 vrf vpn1 80000009030000B064EFE100`	スイッチ上で、リモート SNMP エンジンの名前を設定します。
ステップ 5	**snmp-server host** *host* **vrf** *vpn-instance* **traps** *community* 例：	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。

	コマンドまたはアクション	目的
	`Device(config)#snmp-server host 172.16.20.3 vrf vpn1 traps comaccess`	
ステップ 6	**snmp-server host** *host* **vrf** *vpn-instance* **informs** *community* 例： `Device(config)#snmp-server host 172.16.20.3 vrf vpn1 informs comaccess`	SNMP 通知動作の受信先を指定し、SNMP通知の送信に使用されるVRFテーブルを指定します。
ステップ 7	**snmp-server user** *user group* **remote** *host* **vrf** *vpn-instance security model* 例： `Device(config)#snmp-server user abcd remote 172.16.20.3 vrf vpn1 priv v2c 3des secure3des`	SNMP アクセス用に、VRF 上にあるリモート ホストの SNMP グループにユーザを追加します。
ステップ 8	**end** 例： `Device(config-if)#end`	特権 EXEC モードに戻ります。

NTP 用 VRF 認識サービスの設定

NTP 用の VRF 認識サービスの設定には、NTP サーバと、NTP サーバに接続された 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の間の接続を確認します。NTP サーバ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。 • プロンプトが表示されたらパスワードを入力します。

	コマンドまたはアクション	目的
	`Device>enable`	
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config)#interface gigabitethernet 1/0/1`	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の両方で同じである必要があります。

	コマンドまたはアクション	目的
ステップ 9	**ntp authenticate** 例： `Device(config)#ntp authenticate`	NTP 認証機能をイネーブルにします。NTP 認証はデフォルトでディセーブルになっています。
ステップ 10	**ntp trusted-key** *key-number* 例： `Device(config)#ntp trusted-key 1`	NTP クライアントで同期をとれるようにするために、NTP サーバによってその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 サーバを設定します。

NTP サーバでの NTP 用 VRF 認識サービスの設定

NTP サーバ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。

	コマンドまたはアクション	目的
		（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。
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 サーバと誤って同期する、ということが防止されます。
ステップ 6	**interface** *interface-id* 例： `Device(config)#interface gigabitethernet 1/0/3`	VRF に関連付けるレイヤ 3 インターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 7	**vrf forwarding** *vrf-name* 例： `Device(config-if)#vrf forwarding A`	VRF をレイヤ 3 インターフェイスに対応付けます。
ステップ 8	**ip address** *ip-address subnet-mask* 例： `Device(config-if)#ip address 1.1.1.2 255.255.255.0`	インターフェイスの IP アドレスを入力します。
ステップ 9	**exit** 例： `Device(config-if)exit`	インターフェイス コンフィギュレーション モードを終了します。

uRPF 用 VRF 認識サービスの設定

uRPF は、VRF に割り当てられたインターフェイス上で設定でき、送信元検索が VRF テーブルで実行されます。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface***interface-id* 例： `Device(config)#interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 4	**no switchport** 例： `Device(config-if)#no switchport`	レイヤ 2 コンフィギュレーション モードからインターフェイスを削除します（物理インターフェイスの場合）。
ステップ 5	**ip vrf forwarding** *vrf-name* 例： `Device(config-if)#ip vrf forwarding vpn2`	インターフェイス上で VRF を設定します。
ステップ 6	**ip address** *ip-address* 例： `Device(config-if)#ip address 10.1.5.1`	インターフェイスの IP アドレスを入力します。
ステップ 7	**ip verify unicast reverse-path** 例： `Device(config-if)#ip verify unicast reverse-path`	インターフェイス上で uRPF をイネーブルにします。
ステップ 8	**end** 例： `Device(config-if)#end`	特権 EXEC モードに戻ります。

VRF 認識 RADIUS の設定

VRF 認識 RADIUS を設定するには、まず RADIUS サーバ上で 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* サーバグループ コンフィギュレーション コマンドと **ip radius source-interface** グローバル コンフィギュレーション コマンドがサポートされます。

syslog 用 VRF 認識サービスの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**logging on** 例： `Device(config)#logging on`	ストレージ ルータ イベント メッセージのロギングを、イネーブルまたは一時的にディセーブルにします。
ステップ 4	**logging host** *ip-address* **vrf** *vrf-name* 例： `Device(config)#logging host 10.10.1.0 vrf vpn1`	ロギング メッセージが送信される Syslog サーバのホスト アドレスを指定します。
ステップ 5	**logging buffered** *logging buffered size* **debugging** 例： `Device(config)#logging buffered critical 6000 debugging`	メッセージを内部バッファにロギングします。
ステップ 6	**logging trap debugging** 例： `Device(config)#logging trap debugging`	Syslog サーバに送信されるロギング メッセージを制限します。

	コマンドまたはアクション	目的
ステップ 7	**logging facility** *facility* 例： `Device(config)#logging facility user`	ロギング ファシリティにシステム ロギング メッセージを送信します。
ステップ 8	**end** 例： `Device(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： `Device(config)#traceroute vrf vpn2 10.10.1.1`	宛先アドレスを取得する VPN VRF の名前を指定します。

FTP および TFTP 用 VRF 認識サービスの設定

FTP および TFTP を VRF 認識とするには、いくつかの FTP/TFTP CLI を設定する必要があります。たとえば、インターフェイスに付加される VRF テーブルを使用する場合、E1/0 であれば、ip tftp source-interface E1/0 コマンドまたは ip ftp source-interface E1/0 コマンドを設定して、特定のルーティング テーブルを使用するように TFTP または FTP サーバに通知する必要があります。この例では、VRF テーブルが宛先 IP アドレスを検索するのに使用されます。これらの変更には下位互換性があり、既存の動作には影響を及ぼしません。つまり、VRF がそのインターフェイスに設定されていない場合でも、送信元インターフェイス CLI を使用して、特定のインターフェイスにパケットを送信できます。

手順

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

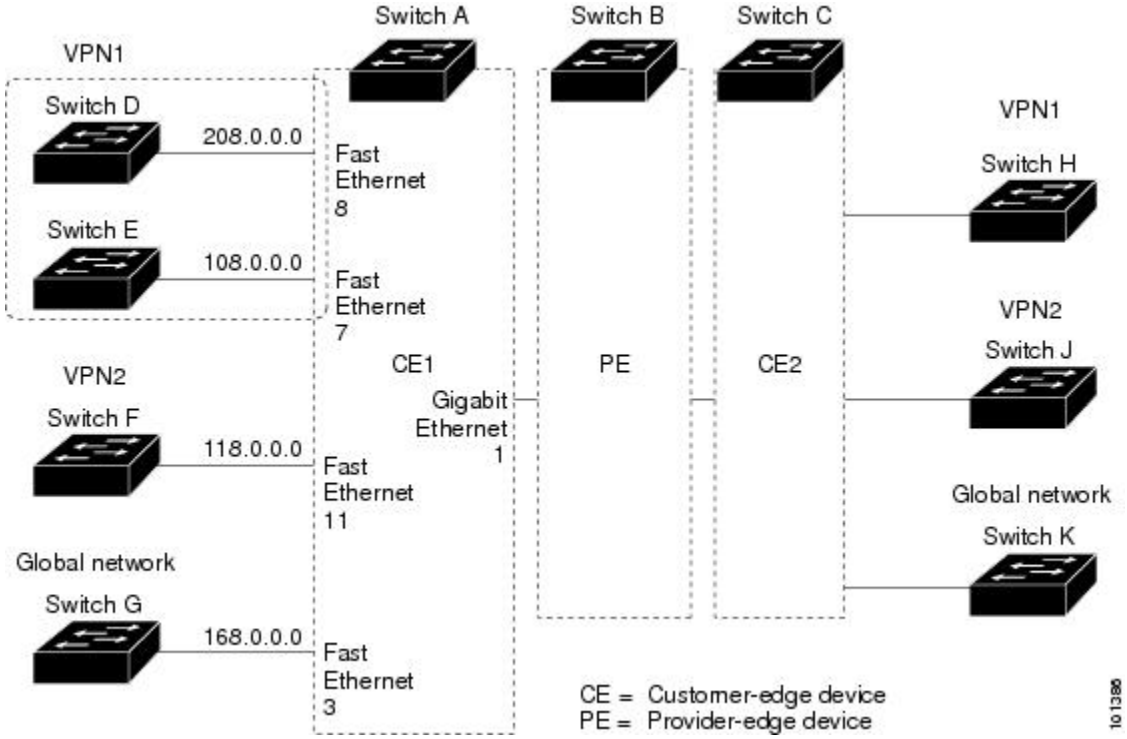
	コマンドまたはアクション	目的
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip ftp source-interface** *interface-type interface-number* 例： `Device(config)#ip ftp source-interface gigabitethernet 1/0/2`	FTP 接続の発信元 IP アドレスを指定します。
ステップ 4	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 5	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 6	**ip tftp source-interface** *interface-type interface-number* 例： `Device(config)#ip tftp source-interface gigabitethernet 1/0/2`	TFTP 接続用の送信元 IP アドレスを指定します。
ステップ 7	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。

Multi-VRF CE の設定例

Multi-VRF CE の設定例

VPN1、VPN2、およびグローバル ネットワークで使用されるプロトコルは OSPF です。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。

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

スイッチAのループバックおよび物理インターフェイスを設定します。ギガビットイーサネット ポート 1 は PE へのトランク接続です。ギガビット イーサネット ポート 8 と 11 は VPN に接続されます。

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

Device(config)#interface loopback2
Device(config-if)#ip vrf forwarding v12
Device(config-if)#ip address 8.8.2.8 255.255.255.0
Device(config-if)#exit
```

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

スイッチ A で使用する VLAN を設定します。VLAN 10 は、CE と PE 間の VRF 11 によって使用されます。VLAN 20 は、CE と PE 間の VRF 12 によって使用されます。VLAN 118 と 208 は、それぞれスイッチ F とスイッチ D を含む VPN に使用されます。

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

VPN1 と VPN2 で OSPF ルーティングを設定します。

```
Device(config)#router ospf 1 vrf vl1
Device(config-router)#redistribute isis subnets
Device(config-router)#network 208.0.0.0 0.0.0.255 area 0
Device(config-router)#exit
Device(config)#router ospf 2 vrf vl2
Device(config-router)#redistribute isis subnets
Device(config-router)#network 118.0.0.0 0.0.0.255 area 0
Device(config-router)#exit
```

スイッチ D は VPN 1 に属します。次のコマンドを使用して、スイッチ A への接続を設定します。

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

Device(config)#router ospf 101
Device(config-router)#network 208.0.0.0 0.0.0.255 area 0
Device(config-router)#end
```

スイッチ F は VPN 2 に属します。次のコマンドを使用して、スイッチ A への接続を設定します。

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

Device(config)#interface vlan118
Device(config-if)#ip address 118.0.0.11 255.255.255.0
Device(config-if)#exit

Device(config)#router ospf 101
Device(config-router)#network 118.0.0.0 0.0.0.255 area 0
Device(config-router)#end
```

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

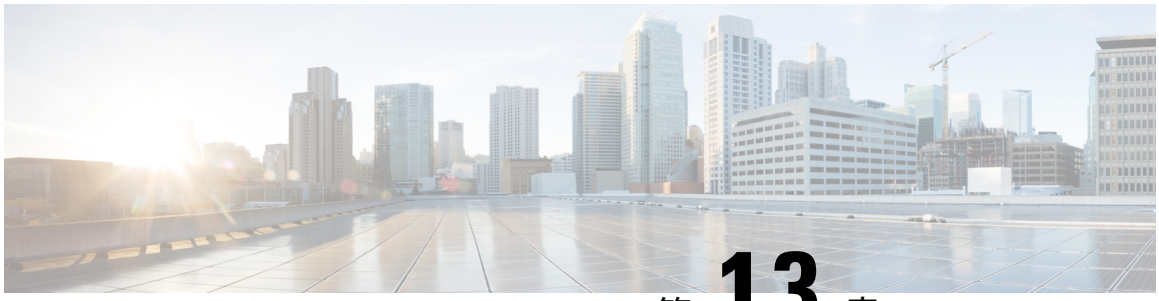
```
Device#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)#ip vrf v1
Device(config-vrf)#rd 100:1
Device(config-vrf)#route-target export 100:1
Device(config-vrf)#route-target import 100:1
Device(config-vrf)#exit

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

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

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

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

Device(config)#router bgp 100
Device(config-router)#address-family ipv4 vrf v2
Device(config-router-af)#neighbor 83.0.0.8 remote-as 800
```

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

マルチ VRF CE の機能情報

表 *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 Fuji 16.9.2	この機能が導入されました
VRF のサポート	Cisco IOS XE Amsterdam 17.2.1	Cisco Catalyst 9200 シリーズ スイッチの C9200-24PB モデルと C9200-48PB モデルでは、32 の VRF がサポートされています。



第 13 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 プロバイダー（ISP）の場合、uRPF が IP ルーティング テーブルと整合性の取れた有効な送信元アドレスを持つパケットだけを転送することによって、そのような攻撃をそらします。この処理により、ISP のネットワーク、その顧客、および残りのインターネットが保護されます。

（注）

- ユニキャスト RPF は、 でサポートされています。

IP uRPF 設定の詳細については、『*Cisco IOS Security Configuration Guide*』の「Other Security Features」の章を参照してください。

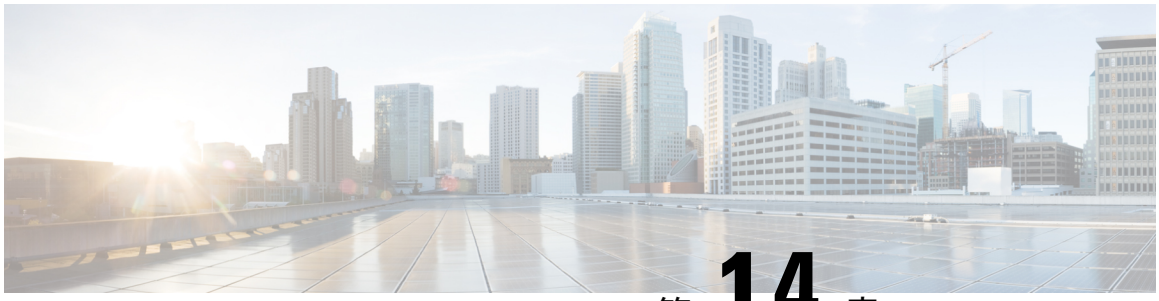
IPv6 ユニキャスト リバース パス転送の設定

ユニキャスト リバース パス転送（ユニキャスト RPF）機能は、検証可能な送信元 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 プロバイダー（ISP）の場合、uRPF が IP ルーティング テーブルと整合性の取れた有効な送信元アドレスを持つパケットだけを転送することによって、そのような攻撃をそらします。この処理により、ISP のネットワーク、その顧客、および残りのインターネットが保護されます。

（注）
- スイッチが複数のスイッチ タイプが混在する混合ハードウェア スタック内にある場合は、ユニキャスト RPF を設定しないでください。

IP ユニキャスト RPF 設定の詳細については、『*Cisco IOS Security Configuration Guide, Release 12.4*』の「*Other Security Features*」の章を参照してください。



第 14 章

プロトコル独立機能

プロトコル独立機能

分散型シスコ エクスプレス フォワーディング

シスコ エクスプレス フォワーディングに関する情報

シスコ エクスプレス フォワーディング（CEF）は、ネットワーク パフォーマンスを最適化するために使用されるレイヤ 3 IP スイッチング技術です。CEF には高度な IP 検索および転送アルゴリズムが実装されているため、レイヤ 3 スイッチングのパフォーマンスを最大化できます。高速スイッチング ルート キャッシュよりも CPU にかかる負担が少ないため、CEF はより多くの CPU 処理能力をパケット転送に割り当てることができます。スイッチ スタックでは、ハードウェアによって distributed CEF（dCEF）が使用されます。動的なネットワークでは、ルーティングの変更によって、高速スイッチング キャッシュ エントリが頻繁に無効になります。高速スイッチングキャッシュ エントリが無効になると、トラフィックがルートキャッシュによって高速スイッチングされずに、ルーティング テーブルによってプロセス スイッチングされることがあります。CEF および dCEF は転送情報ベース（FIB）検索テーブルを使用して、宛先ベースの IP パケット スイッチングを実行します。

CEF および dCEF での 2 つの主要なコンポーネントは、分散 FIB と分散隣接テーブルです。

- FIB はルーティング テーブルや情報ベースと同様、IP ルーティング テーブルに転送情報のミラーイメージが保持されます。ネットワーク内でルーティングまたはトポロジが変更されると、IP ルーティング テーブルがアップデートされ、これらの変更が FIB に反映されます。FIB には、IP ルーティング テーブル内の情報に基づいて、ネクストホップのアドレス情報が保持されます。FIB にはルーティング テーブル内の既知のルートがすべて格納されているため、CEF はルート キャッシュをメンテナンスする必要がなく、トラフィックのスイッチングがより効率化され、トラフィック パターンの影響も受けません。
- リンク層上でネットワーク内のノードが 1 ホップで相互に到達可能な場合、これらのノードは隣接関係にあると見なされます。CEF は隣接テーブルを使用し、レイヤ 2 アドレッシ

ング情報を付加します。隣接テーブルには、すべての FIB エントリに対する、レイヤ 2 のネクストホップのアドレスが保持されます。

スイッチまたはスイッチスタックは、ギガビット速度の回線レート IP トラフィックを達成するため特定用途向け集積回路（ASIC）を使用しているので、CEF または dCEF 転送はソフトウェア転送パス（CPU により転送されるトラフィック）にだけ適用されます。

シスコ エクスプレス フォワーディングの設定方法

デフォルトで、CEF または dCEF はグローバルにイネーブルに設定されています。何らかの理由でこれが無効になった場合は、**ip cef**または **ip cef distributed** グローバル コンフィギュレーション コマンドを使用し、再度有効に設定できます。

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： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**ip cef** 例： `Device(config)# ip cef`	非スタッキング スイッチで CEF の動作をイネーブルにします。 ステップ 4 に進みます。
ステップ 3	**ip cef distributed** 例： `Device(config)# ip cef distributed`	アクティブ スイッチで CEF の動作をイネーブルにします。

	コマンドまたはアクション	目的
ステップ 4	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーションモードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 5	**ip route-cache cef** 例： `Device(config-if)# ip route-cache cef`	ソフトウェア転送トラフィック用のインターフェイスでCEF をイネーブルにします。
ステップ 6	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show ip cef** 例： `Device# show ip cef`	すべてのインターフェイスの CEF ステータスを表示します。
ステップ 8	**show cef linecard** [**detail**] 例： `Device# show cef linecard detail`	（任意）非スタッキング スイッチの CEF 関連インターフェイス情報を表示します。
ステップ 9	**show cef linecard** [*slot-number*] [**detail**] 例： `Device# show cef linecard 5 detail`	（任意）スタック内のすべてのスイッチ、または指定されたスイッチに対して、スイッチのCEF 関連インターフェイス情報をスタック メンバ別に表示します。 （任意）*slot-number* には、スタック メンバーのスイッチ番号を入力します。
ステップ 10	**show cef interface** [*interface-id*] 例： `Device# show cef interface gigabitethernet 1/0/1`	すべてのインターフェイスまたは指定されたインターフェイスの詳細な CEF 情報を表示します。
ステップ 11	**show adjacency** 例： `Device# show adjacency`	CEF の隣接テーブル情報を表示します。

	コマンドまたはアクション	目的
ステップ 12	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーションファイルに設定を保存します。

CEF トラフィック用のロードバランシングスキーム

CEFトラフィック用のロードバランシングスキームの設定に関する制約事項

- デバイスまたはデバイススタックメンバのロードバランシングを同じように、グローバルに設定する必要があります。
- CEF トラフィックのパケットごとのロードバランシングはサポートされていません。

CEF ロード バランシングの概要

CEF のロードバランシングを行うと、トラフィックを複数のパスに分散することにより、リソースを最適化することができます。CEFのロードバランシングは、送信元と宛先のパケット情報の組み合わせに基づいて動作します。

ロードバランシングは宛先単位で設定できます。ロードバランシングの判断はアウトバウンドインターフェイス上で行われるため、ロード バランシングは、アウトバウンド インターフェイスで設定する必要があります。

CEF トラフィックに対する宛先別ロードバランシング

宛先単位のロードバランシングにより、デバイスは、複数のパスを使用して、複数の発信元と宛先ホストのペアにわたって負荷を共有することができます。指定された発信元と宛先ホストのペアは、複数のパスを使用可能な場合であっても、同じパスを使用することが保証されています。異なるペアを宛先とするトラフィック ストリームは、異なるパスを使用します。

CEF がイネーブルの場合、宛先別ロード バランシングはデフォルトでイネーブルです。CEF をイネーブルにした場合、宛先単位のロードバランシングを使用するための追加タスクはありません。多くの状況では、ロード バランシングの方法として宛先単位を使用します。

宛先単位のロードバランシングはトラフィックの統計的な分散に依存しているため、発信元と宛先ホストのペア数が増大すると、ロード シェアリングがさらに有効になります。

宛先単位のロード バランシングを使用することにより、個々のホスト ペアのパケットが順に到達することが保証されます。特定のホストペアに宛てられたすべてのパケットは、（複数の場合も）同じリンクを介して転送されます。

CEF トラフィックに対するロード バランシング アルゴリズム

CEF トラフィックで使用するために、次のロード バランシング アルゴリズムが用意されています。ロードバランシング アルゴリズムは、**ip cef load-sharing algorithm** コマンドで選択します。

- オリジナルアルゴリズム：オリジナルのロード バランシング アルゴリズムでは、すべてのデバイスで同じアルゴリズムが使用されるため、複数のデバイスにわたるロードシェアリングで歪みが発生します。ネットワーク環境に応じて、アルゴリズムを選択する必要があります。
- ユニバーサルアルゴリズム：ユニバーサル ロード バランシング アルゴリズムでは、ネットワーク上の各デバイスは、発信元と宛先の各アドレスペアに対して異なるロードシェアリングの判断を行うことができます。これにより、ロードシェアリングの不均衡が解決されます。デバイスは、デフォルトではユニバーサル ロード シェアリングを実行するよう設定されています。

CEF トラフィックに対するロードバランシングの設定方法

ここでは、CEF トラフィックに対するロードバランシングの設定について説明します。

CEF の宛先別ロードバランシングの有効化または無効化

CEF の宛先単位のロードバランシングを有効または無効にするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device# enable`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config-if)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 4	[**no**] **ip load-sharing per-destination** 例：	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。

	コマンドまたはアクション	目的
	`Device(config-if)# ip load-sharing per-destination`	**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

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device# enable`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip cef load-sharing algorithm** {**original** | **universal** [*id*] } 例： `Device(config)# ip cef load-sharing algorithm universal`	CEF のロードバランシング アルゴリズムを選択します。 • **original** キーワードは、送信元 IP と宛先 IP のハッシュに基づいて、ロード バランシング アルゴリズムとしてオリジナルアルゴリズムを設定します。 • **universal** キーワードは、送信元 IP、宛先 IP、レイヤ 3 プロトコル、レイヤ 4 送信元ポート、レイヤ 4 宛先ポート、および IPv6 トラフィック

	コマンドまたはアクション	目的
		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	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router** {**rip** | **ospf** | **eigrp**} 例： `Device(config)# router eigrp`	ルータ コンフィギュレーション モードを開始します。
ステップ 4	**maximum-paths** *maximum* 例： `Device(config-router)# maximum-paths 2`	プロトコル ルーティング テーブルのパラレル パスの最大数を設定します。指定できる範囲は 1 ～ 16 です。ほとんどの IP ルーティング プロトコルでデフォルトは 4 ですが、BGP の場合だけ 1 です。
ステップ 5	**end** 例： `Device(config-router)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show ip protocols** 例： `Device# show ip protocols`	*Maximum path* フィールドの設定を確認します。
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

スタティック ユニキャスト ルート

スタティック ユニキャスト ルートに関する情報

スタティック ユニキャスト ルートは、特定のパスを通過して送信元と宛先間でパケットを送受信するユーザ定義のルートです。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
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定義のルートです。ルータが特定の宛先へのルートを構築できない場合、スタティック ルートは重要で、到達不能なすべてのパケットが送信される最終ゲートウェイを指定する場合に有効です。

スタティック ルートを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip route prefix mask** {*address* | *interface*} [*distance*] 例： `Device(config)# ip route prefix mask gigabitethernet 1/0/4`	スタティック ルートを確立します。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show ip route** 例： `Device# show ip route`	設定を確認するため、ルーティング テーブルの現在の状態を表示します。
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

次のタスク

スタティックルートを削除するには、**no ip route** *prefix mask* {*address*| *interface*} グローバル コンフィギュレーション コマンドを使用します。ユーザによって削除されるまで、スタティック ルートはデバイスに保持されます。

デフォルトのルートおよびネットワーク

デフォルトのルートおよびネットワークに関する情報

ルータは、他のすべてのネットワークへのルートを学習できません。完全なルーティング機能を実現するには、一部のルータをスマートルータとして使用し、それ以外のルータのデフォルト ルートをスマート ルータ宛てに指定します（スマート ルータにはインターネットワーク全体のルーティング テーブルに関する情報が格納されます）。これらのデフォルト ルートは動的に学習できますが、ルータごとに設定することもできます。ほとんどのダイナミックな内部ルーティング プロトコルには、スマート ルータを使用してデフォルト情報を動的に生成し、他のルータに転送するメカニズムがあります。

指定されたデフォルト ネットワークに直接接続されたインターフェイスがルータに存在する場合は、そのデバイス上で動作するダイナミック ルーティング プロトコルによってデフォルト ルートが生成されます。RIP の場合は、疑似ネットワーク 0.0.0.0 がアドバタイズされます。

ネットワークのデフォルトを生成しているルータには、そのルータ自身のデフォルト ルートも指定する必要があります。ルータが自身のデフォルト ルートを生成する方法の 1 つは、適切なデバイスを経由してネットワーク 0.0.0.0 に至るスタティック ルートを指定することです。

ダイナミック ルーティング プロトコルによってデフォルト情報を送信するときは、特に設定する必要はありません。ルーティング テーブルは定期的にスキャンされ、デフォルト ルートとして最適なデフォルト ネットワークが選択されます。IGRP ネットワークでは、システムのデフォルト ネットワークの候補が複数存在する場合もあります。Cisco ルータでは、デフォルト ルートまたは最終ゲートウェイを設定するため、アドミニストレーティブ ディスタンスおよびメトリック情報を使用します。

ダイナミックなデフォルト情報がシステムに送信されない場合は、**ip default-network** グローバル コンフィギュレーション コマンドを使用し、デフォルトルートの候補を指定します。このネットワークが任意の送信元のルーティング テーブルに格納されている場合は、デフォルト ルートの候補としてフラグ付けされます。ルータにデフォルト ネットワークのインターフェイスが存在しなくても、そこへのパスが格納されている場合、そのネットワークは 1 つの候補と見なされ、最適なデフォルト パスへのゲートウェイが最終ゲートウェイになります。

デフォルトのルートおよびネットワークの設定方法

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例：	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
	`Device# configure terminal`	
ステップ 2	**ip default-network** *network number* 例： `Device(config)# ip default-network 1`	デフォルト ネットワークを指定します。
ステップ 3	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 4	**show ip route** 例： `Device# show ip route`	最終ゲートウェイで選択されたデフォルト ルートを表示します。
ステップ 5	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ルーティング情報を再配信するためのルート マップ

ルート マップの概要

スイッチでは複数のルーティング プロトコルを同時に実行し、ルーティング プロトコル間で情報を再配信できます。ルーティング プロトコル間での情報の再配信は、サポートされているすべての IP ベース ルーティング プロトコルに適用されます。

2 つのドメイン間で拡張パケット フィルタまたはルート マップを定義することにより、ルーティング ドメイン間でルートの再配信を条件付きで制御することもできます。**match** および **set** ルート マップ コンフィギュレーション コマンドは、ルートマップの条件部を定義します。**match** コマンドは、条件が一致する必要があることを指定しています。**set** コマンドは、ルーティングアップデートが match コマンドで定義した条件を満たす場合に行われる処理を指定します。再配布はプロトコルに依存しない機能ですが、**match** および **set** ルート マップ コンフィギュレーション コマンドの一部は特定のプロトコル固有のものです。

route-map コマンドのあとに、**match** コマンドおよび **set** コマンドをそれぞれ 1 つまたは複数指定します。**match** コマンドを指定しない場合は、すべて一致すると見なされます。**set** コマンドを指定しない場合、一致以外の処理はすべて実行されません。このため、少なくとも 1 つの **match** または **set** コマンドを指定する必要があります。

（注） **set** ルート マップ コンフィギュレーション コマンドを使用しないルートマップは、CPU に送信されるので、CPU の使用率が高くなります。

ルートマップステートメントは、**permit** または **deny** として識別することもできます。ステートメントが拒否としてマークされている場合、一致基準を満たすパケットは通常の転送チャネルを通じて送り返されます（宛先ベースルーティング）、ステートメントが許可としてマークされている場合は、一致基準を満たすパケットに set コマンドが適用されます。一致基準を満たさないパケットは、通常のルーティング チャネルを通じて転送されます。

ルート マップの設定方法

次に示すステップ 3 ～ 14 はそれぞれ任意ですが、少なくとも 1 つの **match** ルート マップ コンフィギュレーション コマンド、および 1 つの **set** ルート マップ コンフィギュレーション コマンドを入力する必要があります。

（注） キーワードは、ルート配信を制御する手順で定義されているものと同じです。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**route-map***map-tag* [**permit** | **deny**] [*sequence number*] 例： `Device(config)# route-map rip-to-ospf permit 4`	再配信を制御するために使用するルート マップを定義し、ルート マップ コンフィギュレーション モードを開始します。 *map-tag*：ルートマップ用のわかりやすい名前を指定します。**redistribute**ルータ コンフィギュレーション コマンドはこの名前を使用して、このルートマップを参照します。複数のルートマップで同じマップタグ名を共有できます。 （任意）**permit** が指定され、このルートマップの一致条件が満たされている場合は、setアクションの制御に従ってルートが再配信されます。**deny**が指定が指定されている場合、ルートは再配信されません。

	コマンドまたはアクション	目的
		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： `Device(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： `Device(config-route-map)# match community-list 150`	BGP コミュニティ リストのマッチングを行います。
ステップ 5	**match ip address** {*access-list-number* | *access-list-name*} [*...access-list-number* | *...access-list-name*] 例： `Device(config-route-map)# match ip address 5 80`	名前または番号を指定し、標準アクセス リストと照合します。1 ～ 199 の整数を指定できます。
ステップ 6	**match metric** metric-value 例： `Device(config-route-map)# match metric 2000`	指定されたルートメトリックと一致させます。*metric-value* には、0 ～ 4294967295 の値が指定された、EIGRP のメトリックを指定できます。
ステップ 7	**match ip next-hop** {*access-list-number* | *access-list-name*} [*...access-list-number* | *...access-list-name*] 例： `Device(config-route-map)# match ip next-hop 8 45`	指定されたアクセス リスト（番号 1 ～ 199）のいずれかで送信される、ネクストホップのルータアドレスと一致させます。
ステップ 8	**match tag** *tag value* [*...tag-value*] 例： `Device(config-route-map)# match tag 3500`	1 つまたは複数のルート タグ値からなるリスト内の指定されたタグ値と一致させます。0 ～ 4294967295 の整数を指定できます。

	コマンドまたはアクション	目的
ステップ 9	**match interface***type number* [*...type-number*] 例： `Device(config-route-map)# match interface gigabitethernet 1/0/1`	指定されたインターフェイスの 1 つから、指定されたネクスト ホップへのルートと一致させます。
ステップ 10	**match ip route-source** {*access-list-number* | *access-list-name*} [*...access-list-number* | *...access-list-name*] 例： `Device(config-route-map)# match ip route-source 10 30`	アドバタイズされた指定のアクセス リストによって指定したアドレスに一致します。
ステップ 11	**match route-type** {**local** | **internal** | **external** [**type-1** | **type-2**]} 例： `Device(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： `Device(config-route-map)# set dampening 30 1500 10000 120`	BGP ルート ダンプニング係数を設定します。
ステップ 13	**set local-preference** *value* 例： `Device(config-route-map)# set local-preference 100`	ローカル BGP パスに値を割り当てます。
ステップ 14	**set origin** {**igp** | **egp** *as* | **incomplete**} 例： `Device(config-route-map)#set origin igp`	BGP 送信元コードを設定します。

	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： `Device(config-route-map)# set as-path tag`	BGP の自律システム パスを変更します。
ステップ 16	**set level** {**level-1** | **level-2** | **level-1-2** | **stub-area** | **backbone**} 例： `Device(config-route-map)# set level level-1-2`	ルーティング ドメインの指定エリアにアドバタイズされるルートのレベルを設定します。**stub-area** および **backbone** は、OSPF NSSA およびバックボーンエリアです。
ステップ 17	**set metric** *metric value* 例： `Device(config-route-map)# set metric 100`	再配布されるルートを指定するためのメトリック値を設定します（EIGRP のみ）。*metric value* は -294967295 ～ 294967295 の整数です。
ステップ 18	**set metric***bandwidth delay reliability loading mtu* 例： `Device(config-route-map)# set metric 10000 10 255 1 1500`	再配布されるルートを指定するためのメトリック値を設定します（EIGRP のみ）。 • *bandwidth*：0 ～ 4294967295 の範囲のルートのメトリック値または IGRP 帯域幅（キロビット/秒単位）。 • *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。

	コマンドまたはアクション	目的
ステップ 19	**set metric-type** {**type-1** | **type-2**} 例： `Device(config-route-map)# set metric-type type-2`	再配信されるルートに OSPF 外部メトリック タイプを設定します。
ステップ 20	**set metric-type internal** 例： `Device(config-route-map)# set metric-type internal`	ネクストホップの IGP メトリックと一致するように、EBGP ネイバーにアドバタイズされるプレフィックスの Multi-Exit 識別子（MED）値を設定します。
ステップ 21	**set weight** *number* 例： `Device(config-route-map)# set weight 100`	ルーティング テーブルの BGP 重みを設定します。指定できる値は 1 ～ 65535 です。
ステップ 22	**end** 例： `Device(config-route-map)# end`	特権 EXEC モードに戻ります。
ステップ 23	**show route-map** 例： `Device# show route-map`	設定を確認するため、設定されたすべてのルートマップ、または指定されたルート マップだけを表示します。
ステップ 24	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ルート配信の制御方法

次に示すステップ 3 ～ 14 はそれぞれ任意ですが、少なくとも 1 つの **match** ルート マップ コンフィギュレーション コマンド、および 1 つの **set** ルート マップ コンフィギュレーション コマンドを入力する必要があります。

（注） キーワードは、再配信用にルート マップを設定する手順で定義されているものと同じです。

ルーティング プロトコルのメトリックを、必ずしも別のルーティング プロトコルのメトリックに変換する必要はありません。たとえば、RIP メトリックはホップ カウントで、IGRP メト

リックは5つの特性の組み合わせです。このような場合は、メトリックを独自に設定し、再配信されたルートに割り当てます。ルーティング情報を制御せずにさまざまなルーティングプロトコル間で交換するとルーティングループが発生し、ネットワーク動作が著しく低下することがあります。

メトリック変換の代わりに使用されるデフォルトの再配信メトリックが定義されていない場合は、ルーティング プロトコル間で自動的にメトリック変換が発生することがあります。

- RIPはスタティック ルートを自動的に再配信できます。スタティック ルートにはメトリック 1（直接接続）が割り当てられます。
- デフォルト モードになっている場合、どのプロトコルも他のルーティング プロトコルを再配信できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**router** {**rip** | **ospf** | **eigrp**} 例： `Device(config)# router eigrp 10`	ルータ コンフィギュレーション モードを開始します。
ステップ 3	**redistribute** *protocol* [*process-id*] {**level-1** | **level-1-2** | **level-2**} [**metric** *metric-value*] [**metric-type** *type-value*] [**match internal** | **external** *type-value*] [**tag** *tag-value*] [**route-map** *map-tag*] [**weight** *weight*] [**subnets**] 例： `Device(config-router)# redistribute eigrp 1`	ルーティング プロトコル間でルートを再配信します。route-map を指定しないと、すべてのルートが再配信されます。キーワード **route-map** に *map-tag* を指定しないと、ルートは配信されません。
ステップ 4	**default-metric** *number* 例： `Device(config-router)# default-metric 1024`	現在のルーティングプロトコルが、再配信されたすべてのルートに対して同じメトリック値を使用するように設定します（RIP と OSPF）。
ステップ 5	**default-metric** bandwidth delay reliability loading mtu 例：	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。

	コマンドまたはアクション	目的
	`Device(config-router)# default-metric 1000 100 250 100 1500`	
ステップ 6	**end** 例： `Device(config-router)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show route-map** 例： `Device# show route-map`	設定を確認するため、設定されたすべてのルートマップ、または指定されたルート マップだけを表示します。
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ポリシーベース ルーティング

PBR の設定に関する制約事項

- ポリシーベースルーティング（PBR）は、トラフィックの GRE トンネルへの転送ではサポートされません。これは、任意のインターフェイスに適用される PBR と、トラフィックの GRE トンネルへの転送（PBR ネクストホップもしくはデフォルトの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 がイネーブルの場合は、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 がイネーブルの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

- 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 をイネーブルにできます。
- 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（CPU）で生成されたパケットまたはローカルパケットは、通常どおりにポリシー ルーティングされません。スイッチ上でローカル PBR をグローバルに有効にすると、そのスイッチから送信されたすべてのユニキャストパケットがローカル PBR の影響を受けます。ロー

カル PBR に関してサポートされているプロトコルは、NTP、DNS、MSDP、SYSLOG、および TFTP です。ローカル PBR は、デフォルトで無効に設定されています。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**route-map** *map-tag* [**permit**] [*sequence number*] 例： `Device(config)# route-map pbr-map permit`	パケットの出力場所を制御するために使用するルートマップを定義し、ルート マップのコンフィギュレーション モードを開始します。 • *map-tag* － ：ルートマップ用のわかりやすい名前を指定します。 **ip policy route-map** インターフェイス コンフィギュレーション コマンドは、この名前を使用して、このルートマップを参照します。同じ map-tag がある複数の route-map 文は、1 つの route-map を定義します。 •（任意） **permit** － ： **permit** が指定され、このルートマップの一致条件が満たされている場合は、set アクションの制御に従ってルートがポリシールーティングされます。 •（任意） *sequence number* － ：シーケンス番号は、特定のルートマップ内の route-map ステートメントの位置を示します。
ステップ 4	**match ip address** {*access-list-number* | *access-list-name*} [*access-list-number* |*...access-list-name*] 例：	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、

	コマンドまたはアクション	目的
	`Device(config-route-map)# match ip address 110 140`	複数の送信元および宛先IPアドレスでも照合できます。 **match** コマンドを指定しない場合、ルートマップはすべてのパケットに適用されます。
ステップ 5	**match length min max** 例： `Device(config-route-map)# match length 64 1500`	パケット長と照合します。
ステップ 6	**set ip next-hop** *ip-address* [*...ip-address*] 例： `Device(config-route-map)# set ip next-hop 10.1.6.2`	基準と一致するパケットの動作を指定します。パケットのルーティング先となるネクストホップを設定します（ネクストホップは隣接している必要があります）。
ステップ 7	**exit** 例： `Device(config-route-map)# exit`	グローバル コンフィギュレーション モードに戻ります。
ステップ 8	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するインタフェースを指定します。
ステップ 9	**ip policy route-map** *map-tag* 例： `Device(config-if)# ip policy route-map pbr-map`	レイヤ 3 インターフェイス上で PBR をイネーブルにし、使用するルートマップを識別します。1 つのインターフェイスに設定できるルート マップは、1 つだけです。ただし、異なるシーケンス番号を持つ複数のルートマップエントリを設定できます。これらのエントリは、最初の一致が見つかるまで、シーケンス番号順に評価されます。一致が見つからない場合、パケットは通常どおりにルーティングされます。
ステップ 10	**ip route-cache policy** 例： `Device(config-if)# ip route-cache policy`	（任意）PBR の高速スイッチングを有効にします。PBR の高速スイッチングを有効にするには、PBR を有効にする必要があります。
ステップ 11	**exit** 例：	グローバル コンフィギュレーション モードに戻ります。

	コマンドまたはアクション	目的
	`Device(config-if)# exit`	
ステップ 12	**ip local policy route-map** *map-tag* 例： `Device(config)# ip local policy route-map local-pbr`	（任意）ローカルPBRを有効にして、スイッチから送信されるパケットにPBRを実行します。ローカルPBRは、スイッチによって生成されるパケットに適用されます。着信パケットには適用されません。
ステップ 13	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 14	**show route-map** [*map-name*] 例： `Device# show route-map`	（任意）設定を確認するため、設定されたすべてのルートマップ、または指定されたルートマップだけを表示します。
ステップ 15	**show ip policy** 例： `Device# show ip policy`	（任意）インターフェイスに付加されたポリシー ルート マップを表示します。
ステップ 16	**show ip local policy** 例： `Device# show ip local policy`	（任意）ローカルPBRが有効であるかどうか、および有効である場合は使用されているルート マップを表示します。

ルーティング情報のフィルタリング

ルーティング プロトコル情報をフィルタリングする場合は、以下の作業を実行します。

（注） OSPF プロセス間でルートが再配信される場合、OSPF メトリックは保持されません。

受動インターフェイスの設定

ローカルネットワーク上の他のルータが動的にルートを取得しないようにするには、**passive-interface** ルータ コンフィギュレーション コマンドを使用し、ルーティング アップデート メッセージがルータインターフェイスから送信されないようにします。OSPF プロトコルでこのコマンドを使用すると、パッシブに指定したインターフェイス アドレスがOSPF ドメインのスタブ ネットワークとして表示されます。OSPFルーティング情報は、指定されたルータインターフェイスから送受信されません。

多数のインターフェイスが存在するネットワークで、インターフェイスを手動でパッシブに設定する作業を回避するには、**passive-interface default** ルータ コンフィギュレーション コマンドを使用し、すべてのインターフェイスをデフォルトでパッシブになるように設定します。このあとで、隣接関係が必要なインターフェイスを手動で設定します。

パッシブとして有効にしたインターフェイスを確認するには、**show ip ospf interface** などのネットワークモニタリング用特権EXECコマンドを使用します。アクティブとして有効にしたインターフェイスを確認するには、**show ip interface** 特権 EXEC コマンドを使用します。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**router** {**rip** | **ospf** | **eigrp**} 例： `Device(config)# router ospf`	ルータ コンフィギュレーション モードを開始します。
ステップ 3	**passive-interface** *interface-id* 例： `Device(config-router)# passive-interface gigabitethernet 1/0/1`	指定されたレイヤ3インターフェイス経由のルーティング アップデートの送信を抑制します。
ステップ 4	**passive-interface default** 例： `Device(config-router)# passive-interface default`	（任意）すべてのインターフェイスを、デフォルトでパッシブとなるように設定します。
ステップ 5	**no passive-interface** *interface type* 例： `Device(config-router)# no passive-interface gigabitethernet1/0/3 gigabitethernet 1/0/5`	（任意）隣接関係を送信する必要があるインターフェイスだけをアクティブにします。
ステップ 6	**network** *network-address* 例： `Device(config-router)# network 10.1.1.1`	（任意）ルーティングプロセス用のネットワーク リストを指定します。*network-address* は IP アドレスです。

	コマンドまたはアクション	目的
ステップ 7	**end** 例： `Device(config-router)# end`	特権 EXEC モードに戻ります。
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ルーティング アップデートのアドバタイズおよび処理の制御

アクセス制御リストと **distribute-list** ルータ コンフィギュレーション コマンドを組み合わせて使用すると、ルーティングアップデート中にルートのアドバタイズを抑制し、他のルータが 1 つまたは複数のルートを取得しないようにできます。この機能を OSPF で使用した場合は外部ルートにだけ適用されるため、インターフェイス名を指定できません。

distribute-list ルータ コンフィギュレーション コマンドを使用し、着信したアップデートのリストのうち特定のルートを処理しないようにすることもできます。（OSPF にこの機能は適用されません）。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router** {**rip** | **eigrp**} 例： `Device(config)# router eigrp 10`	ルータ コンフィギュレーション モードを開始します。
ステップ 4	**distribute-list** {*access-list-number* | *access-list-name*} **out** [*interface-name* | *routing process* | *autonomous-system-number*] 例：	アクセス リスト内のアクションに応じて、ルーティングアップデート内のルートのアドバタイズを許可または拒否します。

	コマンドまたはアクション	目的
	`Device(config-router)# distribute 120 out gigabitethernet 1/0/7`	
ステップ 5	**distribute-list** {*access-list-number* | *access-list-name*} **in** [*type-number*] 例： `Device(config-router)# distribute-list 125 in`	アップデートにリストされたルートの処理を抑制します。
ステップ 6	**end** 例： `Device(config-router)# end`	特権 EXEC モードに戻ります。
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ルーティング情報の送信元のフィルタリング

一部のルーティング情報が他の情報よりも正確な場合があるため、フィルタリングを使用して、さまざまな送信元から送られる情報にプライオリティを設定できます。「アドミニストレーティブディスタンス」は、ルータやルータのグループなど、ルーティング情報の送信元の信頼性を示す数値です。大規模ネットワークでは、他のルーティングプロトコルよりも信頼できるルーティング プロトコルが存在する場合があります。アドミニストレーティブ ディスタンスの値を指定すると、ルータはルーティング情報の送信元をインテリジェントに区別できるようになります。常にルーティング プロトコルのアドミニストレーティブ ディスタンスが最短（値が最小）であるルートが選択されます。

各ネットワークには独自の要件があるため、アドミニストレーティブディスタンスを割り当てる一般的な注意事項はありません。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例：	グローバル 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** {**rip** | **ospf** | **eigrp**} 例： `Device(config)# router eigrp 10`	ルータ コンフィギュレーション モードを開始します。
ステップ 4	**distance** *weight* {*ip-address* {*ip-address mask*}} [*ip access list*] 例： `Device(config-router)# distance 50 10.1.5.1`	アドミニストレーティブ ディスタンスを定義します。 *weight*：アドミニストレーティブディスタンスは 10 ～ 255 の整数です。単独で使用した場合、*weight* はデフォルトのアドミニストレーティブ ディスタンスを指定します。ルーティング情報の送信元に他の指定がない場合に使用されます。アドミニストレーティブ ディスタンスが 255 のルートはルーティング テーブルに格納されません。 （任意）*ip access list*：着信ルーティング アップデートに適用される IP 標準または IP 拡張アクセス リストです。
ステップ 5	**end** 例： `Device(config-router)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show ip protocols** 例： `Device# show ip protocols`	指定されたルーティング プロセス用のデフォルトのアドミニストレーティブ ディスタンスを表示します。
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

認証キーの管理

キー管理を使用すると、ルーティングプロトコルで使用される認証キーを制御できます。一部のプロトコルでは、キー管理を使用できません。認証キーは EIGRP および RIP バージョン 2 で使用できます。

前提条件

認証キーを管理する前に、認証をイネーブルにする必要があります。プロトコルに対して認証をイネーブルにする方法については、該当するプロトコルについての説明を参照してください。認証キーを管理するには、キー チェーンを定義してそのキー チェーンに属するキーを識別し、各キーの有効期間を指定します。各キーは、独自のキー識別子（**key** *number* キーチェーン コンフィギュレーション コマンドで指定されたもの）を保持し、ローカルに格納されています。キー ID、およびメッセージに関連付けられたインターフェイスの組み合わせにより、使用中の認証アルゴリズムおよび Message Digest 5（MD5）認証キーが一意に識別されます。

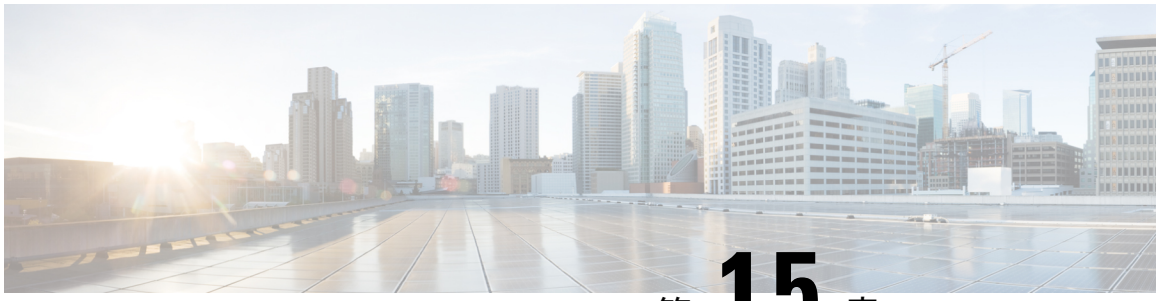
認証キーの設定方法

有効期間が指定された複数のキーを設定できます。存在する有効なキーの数にかかわらず、送信される認証パケットは1つだけです。最小の番号から順にキー番号が調べられ、最初に見つかった有効なキーが使用されます。キー変更中は、有効期間が重なっても問題ありません。これらの有効期間は、ルータに通知する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**key chain** *name-of-chain* 例： `Device(config)# key chain key10`	キー チェーンを識別し、キーチェーン コンフィギュレーション モードを開始します。
ステップ 3	**key** *number* 例： `Device(config-keychain)# key 2000`	キー番号を識別します。有効値は 0 ～ 2147483647 です。
ステップ 4	**key-string** *text* 例： `Device(config-keychain)# Room 20, 10th floor`	キー ストリングを確認します。ストリングには 1 ～ 80 文字の大文字および小文字の英数字を指定できますが、最初の文字に数字を指定できません。
ステップ 5	**accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} 例： `Device(config-keychain)# accept-lifetime 12:30:00 Jan 25 1009 infinite`	（任意）キーを受信できる期間を指定します。 *start-time* および *end-time* 構文には、*hh:mm:ss Month date year* または *hh:mm:ss date Month year* のいずれかを使用できま

	コマンドまたはアクション	目的
		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： `Device(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： `Device(config-keychain)# end`	特権 EXEC モードに戻ります。
ステップ 8	**show key chain** 例： `Device# show key chain`	認証キーの情報を表示します。
ステップ 9	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。



第 15 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（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

	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 tunnel***number* **例：** `Device(config)#interface tunnel 0`	指定したインターフェイスのインターフェイス コンフィギュレーション モードを開始します。 • *number* はトンネルインターフェイスに関連付けられた番号です。

	コマンドまたはアクション	目的
ステップ 4	**ip vrf forwarding***vrf-name* 例： `Device(config-if)#ip vrf forwarding green`	バーチャル プライベート ネットワーク（VPN）ルーティングおよび転送（VRF）インスタンスをインターフェイスまたはサブインターフェイスに関連付けます。 • *vrf-name* は VRF に割り当てられる名前です。
ステップ 5	**ip address***ip-address subnet-mask* 例： `Device(config-if)#ip address 10.7.7.7 255.255.255.255`	インターフェイス IP アドレスとサブネット マスクを指定します。 • *ip-address* でインターフェイスの IP アドレスを指定します。 • *subnet-mask* でインターフェイスのサブネットマスクを指定します。
ステップ 6	**tunnel source** {*ip-address* | *type number*} 例： `Device(config-if)#tunnel source loop 0`	トンネル インターフェイスの送信元を指定します。 • *ip-address* でトンネル内のパケットの送信元アドレスとして使用する IP アドレスを指定します。 • *type* でインターフェイスのタイプ（シリアルなど）を指定します。 • *number* でポート、コネクタ、またはインターフェイスカードの番号を指定します。この番号は、設置時、またはシステムへの追加時に、工場で割り当てられます。また、show interfaces コマンドを使用して表示できます。
ステップ 7	**tunnel destination** {*hostname* | *ip-address*} 例： `Device(config-if)#tunnel destination 10.5.5.5`	トンネルの宛先を指定します。 • *hostname* で宛先ホストの名前を指定します。 • *ip-address* で宛先ホストの IP アドレスを指定します。

	コマンドまたはアクション	目的
ステップ 8	**tunnel vrf***vrf-name* 例： `Device(config-if)#tunnel vrf finance1`	特定のトンネル宛先に VPN ルーティングおよび転送（VRF）インスタンスを関連付けます。 • *vrf-name* は VRF に割り当てられる名前です。

GRE トンネル IP 送信元および宛先 VRF メンバーシップの設定例

次に、VRF green を使用してインターフェイス e0 で受信されたパケットを、VRF blue を使用し、インターフェイス e1 を通じてトンネルから外部へ転送する例を示します。

```
ip vrf blue rd 1:1

ip vrf green rd 1:2

interface loop0
ip vrf forwarding blue
ip address 10.7.7.7 255.255.255.255

interface tunnel0
ip vrf forwarding green
ip address 10.3.3.3 255.255.255.0 tunnel source loop 0
tunnel destination 10.5.5.5 tunnel vrf blue

interface ethernet0
ip vrf forwarding green
ip address 10.1.1.1 255.255.255.0

interface ethernet1
ip vrf forwarding blue
ip address 10.2.2.2 255.255.255.0

ip route vrf blue 10.5.5.5 255.255.255.0 ethernet 1
```

その他の参考資料

表 *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』

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

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

表 *30 : 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。