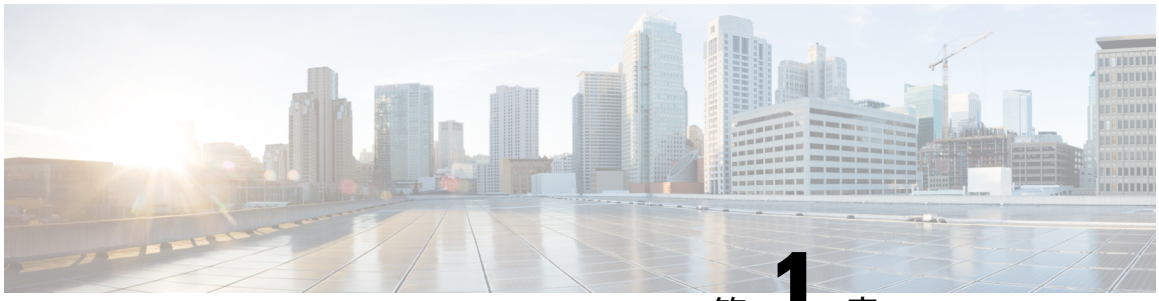
Cisco Catalyst IE3x00 高耐久性、IE3400 Heavy Duty、ESS3300 シリーズスイッチ IP ルーティング コンフィギュレーション ガイド

初版：2021 年 8 月 19 日

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

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



第 1 章

双方向フォワーディング検出の設定

双方向フォワーディング検出

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

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

双方向フォワーディング検出の前提条件

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

双方向フォワーディング検出の制約事項

- 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。

- BFD HA はサポートされていません。
- BFD エコーセッションスケール：100ms 間隔の最大 28 BFD セッション、デバイスごとに許可されるエコーモード BFD セッション。
- IOS XE 17.5.1では、BFD の OSPF および OSPFv3 サポートのみがサポートされています。
- ポートタイプ別のサポートされる時間間隔：

ポート タイプ	最小間隔
ルーテッドポート	100 ミリ秒
SVI	100 ミリ秒
L3 ポート	250ms

双方向フォワーディング検出について

BFD の動作

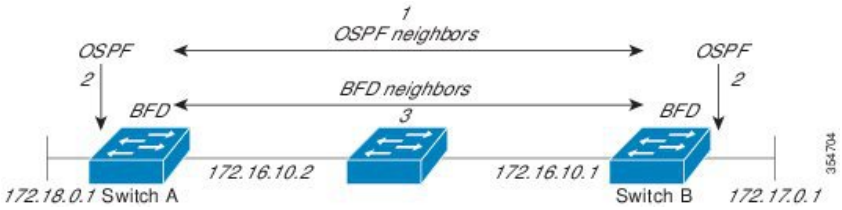
BFD は、2 つの隣接デバイス間の転送パスで、オーバーヘッドの少ない短期間の障害検出方法を提供します。これらのデバイスには、インターフェイス、データリンク、および転送プレーンが含まれます。

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

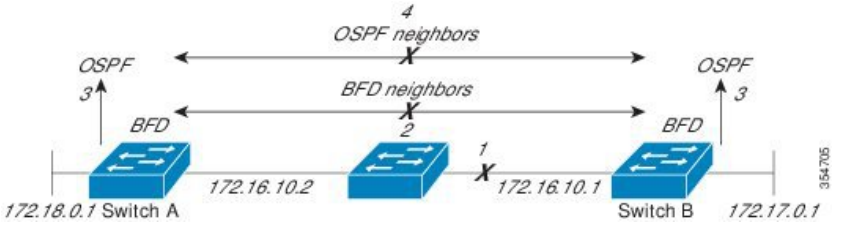
ネイバー関係

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

が開始されます（2）。OSPF ネイバー ルータでの BFD ネイバー セッションが確立されます（3）。



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



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

次のとき、OSPF では、BFD を使用して登録が行われます。

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

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

BFD の障害検出

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

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

EIGRP を実行している場合、1 つの BFD セッションだけが確立されます。BFD は両方のルーティングプロトコルとセッション情報を共有します。

BFD バージョンの相互運用性

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

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

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

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。

双方向フォワーディング検出の設定方法

インターフェイスでの BFD セッション パラメータの設定

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

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	次のいずれかの手順を実行します。 • **ip address** *ipv4-address mask* • **ipv6 address** *ipv6-address*/*mask*	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

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

OSPF に対する BFD サポートの設定

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

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

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

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

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

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

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

始める前に

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router ospf** *process-id* 例： `Device(config)#router ospf 4`	OSPF プロセスを指定し、ルータ コンフィギュレーション モードを開始します。

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

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

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

手順の概要

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

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *type number* 例： `Device(config)#interface fastethernet 6/0`	インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**ip ospf bfd** [**disable**] 例： `Device(config-if)#ip ospf bfd`	OSPF ルーティング プロセスに関連付けられた 1 つ以上のインターフェイスに対して、インターフェイスごとに BFD を有効または無効にします。 （注） ルータ コンフィギュレーション モードで **bfd all-interfaces** コマンドを使用して OSPF が関連付けられたすべてのインターフェイスで BFD を有効にした場合にのみ、**disable** キーワードを使用します。
ステップ 5	**end** 例： `Device(config-if)#end`	インターフェイス コンフィギュレーション モードを終了して、デバイスが特権 EXEC モードに戻ります。
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 サポートが有効になっているかどうかを検証するために使用できる情報を表示します。

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：	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

	`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 5	**bfd interval** *milliseconds* **mix_rx** *milliseconds* **multiplier** *interval-multiplier* 例： `Device(config-if)#bfd interval 500 min_rx 500 multiplier 5`	インターフェイスで BFD を有効にします。 bfd interval 設定は、それを設定したサブインターフェイスが削除されたときに削除されます。 bfd interval 設定は次のような場合には削除されません。 • IPv4 アドレスがインターフェイスから削除された場合 • IPv6 アドレスがインターフェイスから削除された場合 • IPv6 がインターフェイスから無効にされた場合 • 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 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 モードが有効になっていますが、方向ごとに個別に実行できるように、無効にすることもできます。

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

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

前提条件

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

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

非対称性のない BFD エコー モードの無効化

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

各 BFD デバイスに対してこの手順を繰り返します。

手順

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

BFD テンプレートの作成と設定

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

（注） 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。

手順の概要

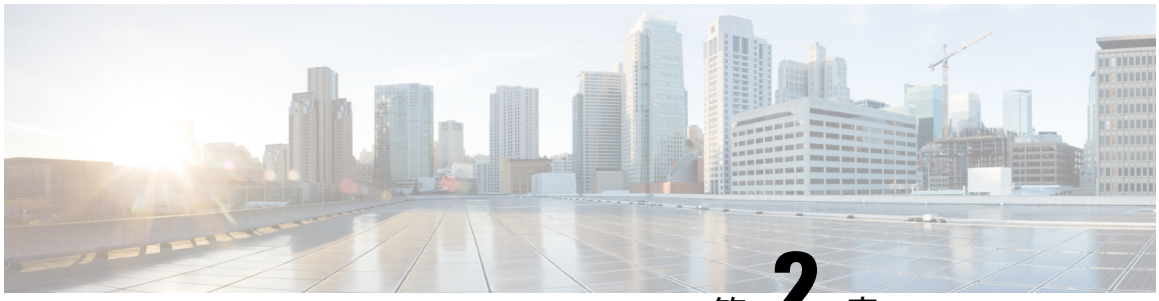
1. **enable**
2. **show bfd neighbors** [**details**]
3. **debug bfd** [**packet** | **event**]

手順の詳細

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

双方向フォワーディング検出の設定例

ここでは、双方向フォワーディング検出の設定例を示します。



第 2 章

IPv4 ポリシーベースルーティングの設定

ポリシーベース ルーティングの概要

（注） この製品のマニュアルセットは、偏向のない言語を使用するように配慮されています。このドキュメントセットでの偏向のない言語とは、年齢、障害、性別、人種的アイデンティティ、民族的アイデンティティ、性的指向、社会経済的地位、およびインターセクショナリティに基づく差別を意味しない言語として定義されています。製品ソフトウェアのユーザインターフェイスにハードコードされている言語、RFPのドキュメントに基づいて使用されている言語、または参照されているサードパーティ製品で使用されている言語によりドキュメントに例外が存在する場合があります。

PBR（ポリシーベースルーティング）は、設定されたポリシーに基づいてルーティングを決定するために使用される技術です。

ルータまたはスイッチがパケットを受信すると、転送の判断はパケットの宛先 IP アドレスに基づいて行われます。宛先 IP アドレスは、ルーティングテーブルのエントリの検索に使用されます。ただし、場合によっては、宛先 IP アドレスではなく送信元 IP アドレスなど、他の基準に基づいてパケットを転送する必要があります。そうすることで、宛先が同じ場合でも、異なる送信元から送信されるパケットを別のネットワークにルーティングできます。これは、複数のプライベートネットワークを相互接続する場合に役立ちます。

PBR が有効な場合は、アクセス コントロール リスト（ACL）を使用してトラフィックを分類し、各トラフィックがそれぞれ異なるパスを経由するようにします。PBR は着信パケットに適用されます。PBR が有効なインターフェイスで受信されたすべてのパケットは、ルート マップを通過します。ルートマップで定義された基準に基づいて、パケットは適切なネクストホップに転送（ルーティング）されます。

- 許可とマークされているルート マップ文は次のように処理されます。

- match コマンドは複数の 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 ip address acl1 acl2

  match ip address acl3

  （注） IPv6 はサポートされていません。

  パケットは、acl1、acl2、または acl3 で許可されている場合に許可されます。

  - 下された判断が許可の場合は、set コマンドで指定されたアクションがパケットに適用されます。
  - 下された判断が拒否の場合は、PBR アクション（set コマンドで指定）は適用されません。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と ACL はサポートされません。

標準 IP ACL を使用すると、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。優先順位番号または名前を使用して IP 優先順位値を設定することもできます。

ポリシーベースルーティングに関する注意事項と制約事項

- PBR を使用するには、スイッチ上で Network Advantage ライセンスを有効にしておく必要があります。
- デフォルトでは、ポリシーベースルーティング（PBR）はスイッチ上で無効になっています。PBR は、ルートマップが設定され、インターフェイスに適用されると有効になります。

- 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。
- PBR には、**route-map deny** ステートメントはサポートされません。
- ACL と拒否 ACE との照合はサポートされていません。
- 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。
- 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。
- ポリシールートマップが ACL および QoS とともにインターフェイスに適用される場合、ACL と QoS が優先されます。
- ルールが同じインターフェイス上の PBR ルールと一致する場合、IP ソースガードの優先順位が高くなります。
- SVI インターフェイスでは、IP ソースガードと PBR ルールが要件に基づいてマージされます。
- マルチキャスト トラフィックには、ポリシーによるルーティングが行われません。PBR が適用されるのはユニキャスト トラフィックだけです。
- ルーテッド ポートまたは SVI 上で、PBR を有効にできます。
- スイッチには最大 64 個の IP ポリシー ルート マップを定義できます。
- スイッチには、PBR 用として最大 64 個のアクセス コントロール エントリ（ACE）を定義できます。
- 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。エントリの最大数は 256 です。
- 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。
- 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。

- 単一のシーケンスルートマップでは、一度に 1 つの set 句のみがサポートされます。複数のシーケンスを持つルートマップでは、同じタイプの set 句のみが許可されます。たとえば、**set ip next-hop** が最初のシーケンスで使用される場合、2 番目のシーケンスにも同じ set 句 **set ip next-hop** が必要になります。
- 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。

次の表に、スイッチの ACL [Match Field] オプションに対する PBR のサポートを示します。

表 *1 : PBR* でサポートされる *ACL [Match Field]* オプション

Match Field	**サポート対象（Y/N）**
Source IP address	Y
Destination IP address	Y
Next Header（ICMP、IGMP など）	N
TCP/UDP Port	N
Type of Service（TOS）	N
Fragmentation Bit	N

次の表に、スイッチでサポートされる PBR 機能を示します。

表 *2 : PBR* 機能のサポート

機能	**サポート/スケール**
入力トラフィックの PBR	Y
出力トラフィックの PBR	N
物理インターフェイス（L2 ポート）の PBR	N
物理インターフェイス（ルーテッドポート）の PBR	Y
SVI インターフェイスの PBR	Y
ポートチャネル（L2）の PBR	N
ポートチャネル（L3）の PBR	N
VRF を使用した PBR	N

機能	サポート/スケール
IPv4 ACL の照合	Y （注） PBR でサポートされる ACL [Match Field] オプションについては、上記の表を参照してください。
拡張/標準 IPv4 ACL の照合	Y
パケット長に基づいた照合	N
拒否 ACE との照合	N
フラグメントビットを設定するアクション	N
優先順位を設定するアクション	N
ネクストホップを設定するアクション	Y
再帰的ネクストホップアクション	N
インターフェイスを設定するアクション	N
デフォルトインターフェイスを設定するアクション	N
IP 優先順位を設定するアクション	Y
IP VRF を設定するアクション	Y
IP デフォルトネクストホップの設定	N
IP デフォルト VRF の設定	N
マルチキャストトラフィックの PBR	N
IPv6 トラフィックの PBR	N
ルートマップ拒否	N
サポートされるルートマップの最大数	64
サポートされる ACL ポリシーの最大数	64
ローカル PBR	Y

PBR の設定方法

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。

始める前に

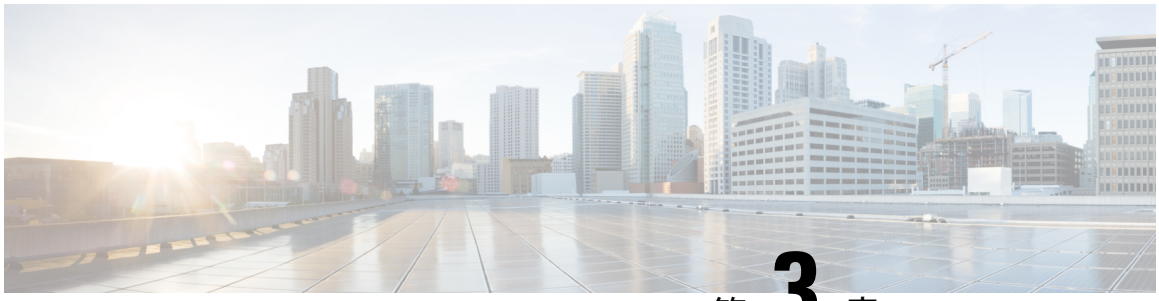
ポリシーベースルーティングに関する注意事項と制約事項（16ページ）を参照してください。

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	**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。

	コマンドまたはアクション	目的
ステップ 4	**match ip address** {*access-list-number* | *access-list-name*} [*access-list-number* |*...access-list-name*] 例： `Device(config-route-map)# match ip address 110 140`	1 つ以上の標準または拡張アクセス リストで許可されている送信元および宛先 IP アドレスを照合します。ACL は、複数の送信元および宛先 IP アドレスでも照合できます。 **match** コマンドを指定しない場合、ルートマップはすべてのパケットに適用されます。
ステップ 5	**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 6	**set ip vrf** *vrf-name* **next-hop** *ip-address* [*...ip-address*] 例： `Device(config-route-map)# set ip vrf myvrf next-hop 10.5.5.5`	VRF インターフェイスにポリシーベースルーティングを適用できます。
ステップ 7	**set ip precedence** [*number* | *name*] 例： `Device(config-route-map)# set ip precedence 5`	• 0：routine • 1：priority • 2：immediate • 3：flash • 4：flash-override • 5：critical • 6：internet • 7：network IP ヘッダーに優先順位を設定します。
ステップ 8	**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 9	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/1`	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 10	**ip policy route-map** *map-tag* 例： `Device(config-if)# ip policy route-map pbr-map`	レイヤ 3 インターフェイス上で PBR を有効にし、使用するルート マップを識別します。1 つのインターフェイスに設定できるルートマップは、1 つだけです。ただし、異なるシーケンス番号を持つ複数

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



第 3 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。

Note この製品のマニュアルセットは、偏向のない言語を使用するように配慮されています。このドキュメントセットでの偏向のない言語とは、年齢、障害、性別、人種的アイデンティティ、民族的アイデンティティ、性的指向、社会経済的地位、およびインターセクショナリティに基づく差別を意味しない言語として定義されています。製品ソフトウェアのユーザインターフェイスにハードコードされている言語、RFP のドキュメントに基づいて使用されている言語、または参照されているサードパーティ製品で使用されている言語によりドキュメントに例外が存在する場合があります。

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 アドレス

スイッチがサポートするのは、IPv6ユニキャストアドレスのみです。サイトローカルユニキャスト アドレスおよびマルチキャスト アドレスはサポートされません。

IPv6 の 128 ビット アドレスは、コロンで区切られた一連の 8 つの 16 進フィールド（n:n:n:n:n:n:n:n. の形式）で表されます。次に、IPv6 アドレスの例を示します。

2031:0000:130F:0000:0000:09C0:080F:130B

実装を容易にするために、各フィールドの先行ゼロは省略可能です。上記アドレスは、先行ゼロを省略した次のアドレスと同じです。

2031:0:130F:0:0:9C0:80F:130B

2 つのコロン（::）を使用して、ゼロが連続する 16 進フィールドを表すことができます。ただし、この短縮形を使用できるのは、各アドレス内で 1 回のみです。

2031:0:130F::09C0:080F:130B

IPv6 アドレス形式、アドレス タイプ、および IPv6 パケット ヘッダーの詳細については、Cisco.com で『*Cisco IOS IPv6 Configuration Library*』の http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipv6_basic/configuration/xe-3e/ip6b-xe-3e-book.html を参照してください。

- IPv6 アドレス形式
- IPv6 アドレス タイプ：マルチキャスト
- Ipv6 アドレス 出力表示
- 簡易 IPv6 パケット ヘッダー

サポート対象の IPv6 ユニキャスト ルーティング機能

ここでは、スイッチでサポートされている IPv6 プロトコル機能について説明します。

128 ビット幅のユニキャスト アドレス

スイッチは集約可能なグローバル ユニキャスト アドレスおよびリンク ローカル ユニキャスト アドレスをサポートします。サイト ローカル ユニキャスト アドレスはサポートされていません。

- 集約可能なグローバル ユニキャスト アドレスは、集約可能グローバル ユニキャスト プレフィックスの付いた IPv6 アドレスです。このアドレス構造を使用すると、ルーティング プレフィックスを厳格に集約することができ、グローバル ルーティング テーブル内のルー

ティング テーブル エントリ数が制限されます。これらのアドレスは、組織を経由して最終的にインターネット サービス プロバイダーに至る集約リンク上で使用されます。

これらのアドレスはグローバル ルーティング プレフィックス、サブネット ID、およびインターフェイス ID によって定義されます。現在のグローバル ユニキャスト アドレス割り当てには、バイナリ値 001（2000::/3）で開始するアドレス範囲が使用されます。プレフィックスが 2000::/3（001）～ E000::/3（111）のアドレスには、Extended Unique Identifier（EUI）64 フォーマットの 64 ビット インターフェイス ID を設定する必要があります。

- リンク ローカル ユニキャスト アドレスをすべてのインターフェイスに自動的に設定するには、修飾 EUI フォーマット内で、リンク ローカル プレフィックス FE80::/10（1111 1110 10）およびインターフェイスID を使用します。ネイバー探索プロトコル（NDP）およびステートレス自動設定プロセスでは、リンクローカルアドレスが使用されます。ローカルリンク上のノードは、リンクローカルアドレスを使用します。通信する場合に、グローバルに一意なアドレスは不要です。IPv6 ルータは、リンクローカルの送信元または宛先アドレスを持つパケットをその他のリンクに転送しません。

詳細については、Cisco.com で『*Cisco IOS IPv6 Configuration Library* 』の「Implementing IPv6 Addressing and Basic Connectivity」の章にある IPv6 ユニキャスト アドレスに関する項を参照してください。

IPv6 の DNS

IPv6 は、ドメイン ネーム システム（DNS）のレコード タイプを、DNS 名前/アドレスおよびアドレス/名前の検索プロセスでサポートします。DNS AAAA リソース レコード タイプは IPv6 アドレスをサポートし、IPv4 の A アドレス レコードと同等です。スイッチは IPv4 および IPv6 の DNS 解決をサポートします。

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できる可能性の高いルータとして、常に同じルータを選択するか、またはルータ リストを循環して選択できます。DRP を使用することにより、両方ともが到達可能または到達できる可能性の高い2台のルータの一方を他方に対して優先させるよう IPv6 ホストを設定することができます。

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

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

IPv6 のステートレス自動設定および重複アドレス検出

スイッチではステートレス自動設定が使用されているため、ホストやモバイル IP アドレスの管理のような、リンク、サブネット、およびサイトアドレス指定の変更を管理することができます。ホストは独自のリンクローカルアドレスを自動的に設定します。起動元ノードはルータに送信請求を送信して、インターフェイス設定をアドバタイズするようルータに要求します。

自動設定および重複アドレス検出の詳細については、Cisco.com で『*Cisco IOS IPv6 Configuration Library*』の「Implementing IPv6 Addressing and Basic Connectivity」の章を参照してください。

IPv6 アプリケーション

スイッチは、次のアプリケーションについて IPv6 をサポートします。

- ping、Traceroute、Telnet、および Trivial File Transfer Protocol（TFTP）
- IPv6 トランスポートによるセキュア シェル（SSH）
- IPv6 トランスポートによる HTTP サーバ アクセス

- IPv4 トランスポートによる AAAA の DNS レゾルバ
- IPv6 アドレスの Cisco Discovery Protocol（CDP）サポート

これらのアプリケーションの管理に関する詳細については、Cisco.com の『*Cisco IOS IPv6 Configuration Library*』を参照してください。

DHCP for IPv6 アドレスの割り当て

DHCPv6 を使用すると、DHCP サーバは IPv6 ネットワーク アドレスなどの設定パラメータを IPv6 クライアントに渡すことができます。このアドレス割り当て機能により、ホストが接続するネットワークに基づいて、適切なプレフィックス内での重複しないアドレス割り当てが管理されます。アドレスは、1 つまたは複数のプレフィックス プールから割り当てることができます。デフォルトのドメインおよび DNS ネーム サーバ アドレスなど、その他のオプションは、クライアントに戻すことができます。アドレス プールは、特定のインターフェイス、複数のインターフェイス上で使用する場合に割り当てられます。または、サーバが自動的に適切なプールを検出できます。

DHCP for IPv6 の設定については、「*DHCP for IPv6* アドレス割り当ての設定」のセクションを参照してください。

DHCPv6 クライアント、サーバ、またはリレーエージェント機能の設定の詳細については、Cisco.com で『*Cisco IOS IPv6 Configuration Library*』を参照してください。

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 のポリシーベース ルーティング

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

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

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

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

PBR を使用すると、次の処理を実行できます。

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

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

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

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

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。

『OSPF for IPv6』

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

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

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

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。

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。

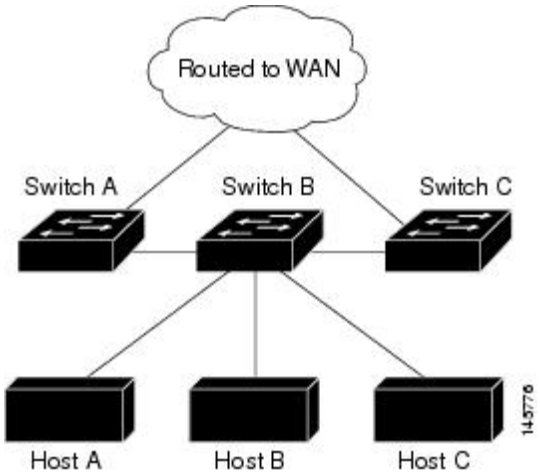
EIGRPv6 スタブ ルーティングを使用するネットワークでは、ユーザに対する IPv6 トラフィックの唯一の許容ルートは、EIGRPv6 スタブ ルーティングを設定しているスイッチ経由のみです。スイッチは、ユーザインターフェイスとして設定されているインターフェイスまたは他のデバイスに接続されているインターフェイスにルーテッド トラフィックを送信します。

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、ディストリビューションルータおよびリモートルータを設定する必要があります。指定したルートだけがスイッチから伝播されます。スイッチは、サマリー、接続ルート、およびルーティング アップデートに対するすべてのクエリーに応答します。

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

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

図 *1: 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。

SNMP and Syslog Over IPv6

IPv4 と IPv6 の両方をサポートするには、IPv6 のネットワーク管理で IPv4 および IPv6 のトランスポートが必要になります。Syslog over IPv6 は、このトランスポートのアドレス データ タイプをサポートします。

Simple Network Management Protocol (SNMP) と syslog over IPv6 は、次の機能を提供します。

- IPv4 と IPv6 両方のサポート
- SNMP に対する IPv6 トランスポート、および SNMP 変更による IPv6 ホストのトラップのサポート
- IPv6 アドレス指定をサポートするための SNMP および syslog に関連する MIB
- IPv6 ホストをトラップ レシーバとして設定

Over IPv6 をサポートするため、SNMP は既存の IP トランスポート マッピングを変更して、IPv4 と IPv6 を同時にサポートします。次の SNMP 動作は、IPv6 トランスポート管理をサポートします。

- デフォルト設定のユーザ データグラム プロトコル（UDP）SNMP ソケットを開く
- *SR_IPV6_TRANSPORT* と呼ばれる新しいトランスポート メカニズムを提供
- IPv6 トランスポートによる SNMP 通知の送信
- IPv6 トランスポートの SNMP 名のアクセス リストのサポート
- IPv6 トランスポートを使用した SNMP プロキシ転送のサポート
- SNMP マネージャ機能と IPv6 トランスポートの連動確認

設定手順を含む、SNMP over IPv6 については、Cisco.com で『*Cisco IOS IPv6 Configuration Library*』の「Managing Cisco IOS Applications over IPv6」の章を参照してください。

設定手順を含む、syslog over IPv6 については、Cisco.com で『*Cisco IOS IPv6 Configuration Library*』の「Implementing IPv6 Addressing and Basic Connectivity」の章を参照してください。

HTTP(S) Over IPv6

HTTP クライアントは要求を IPv4 HTTP サーバと IPv6 HTTP サーバの両方に送信し、これらのサーバは IPv4 HTTP クライアントと IPv6 HTTP クライアントの両方からの要求に応答します。IPv6 アドレスを含む URL は、16 ビット値をコロンで区切った 16 進数で指定する必要があります。

受信ソケット コールは、IPv4 アドレス ファミリまたは IPv6 アドレス ファミリを選択します。受信ソケットは、IPv4 ソケットまたは IPv6 ソケットのいずれかです。リスニング ソケットは、接続を示す IPvv4 と IPv6 の両方の信号を待ち受け続けます。IPv6 リスニング ソケットは、IPv6 ワイルドカード アドレスにバインドされています。

基本 TCP/IP スタックは、デュアル スタック環境をサポートします。HTTP には、TCP/IP スタック、およびネットワーク層相互作用を処理するためのソケットが必要です。

HTTP 接続を確立するには、基本ネットワーク接続（**ping**）がクライアントとサーバホストとの間に存在する必要があります。

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

サポートされていない IPv6 ユニキャスト ルーティング機能

スイッチは、次の IPv6 機能をサポートしません。

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

IPv6 機能の制限

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* ルーティングの有効化」を参照してください。

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

IPv6 アクティブスイッチおよびメンバーの機能は次のとおりです。

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

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

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

IPv6 のデフォルト設定

Table 3: IPv6 のデフォルト設定

機能	デフォルト設定
IPv6 ルーティング	すべてのインターフェイスでグローバルに無効 **Note** IP ルーティングはデフォルトで有効になっていますが、IPv6 ユニキャストルーティングはユーザが有効にする必要があります。
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） **Note** IPv6 ルーティングを有効にすると、IPv6 用 Cisco Express Forwarding および IPv6 用 distributed Cisco Express Forwarding（dCEF; 分散型シスコ エクスプレス フォワーディング）は自動的に有効になります。
IPv6 アドレス	未設定

IPv6 ユニキャストルーティングの設定方法

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

IPv6 アドレッシングの設定と IPv6 ルーティングの有効化

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

スイッチ上の IPv6 を設定する前に、次の注意事項に従ってください。

- スイッチでは、この章で説明されたすべての機能がサポートされるわけではありません。「サポートされていない IPv6 ユニキャスト ルーティング機能」を参照してください。

- **ipv6 address** インターフェイス コンフィギュレーション コマンドでは、16 ビット値を使用したコロン区切りの 16 進形式で指定したアドレスで *ipv6-address* 変数および *ipv6-prefix* 変数を入力する必要があります。*prefix-length* 変数（スラッシュ（/）で始まる）は、プレフィックス（アドレスのネットワーク部分）を構成するアドレスの上位連続ビット数を示す 10 進値です。

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

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

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

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

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

Procedure

	Command or Action	Purpose
ステップ 1	**enable** **Example:** `> 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** **Example:** `# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	Command or Action	Purpose
ステップ 3	**end** **Example:** `(config)# end`	特権 EXEC モードに戻ります。
ステップ 4	**reload** **Example:** `# 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 5	**configure terminal** **Example:** `# 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 6	**interface** *interface-id* **Example:** `(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ3インターフェイスを指定します。インターフェイスは物理インターフェイス、スイッチ仮想インターフェイス（SVI）、またはレイヤ 3 EtherChannel に設定できます。
ステップ 7	**no switchport** **Example:** `(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 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]*
	Example:
	`(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`
		• IPv6 アドレスの下位 64 ビットの拡張固有識別子（EUI）を使用して、グローバル IPv6 アドレスを指定します。ネットワーク プレフィックスだけを指定します。最終の 64 ビットは、スイッチの MAC アドレスから自動的に計算されます。これにより、インターフェイス上で IPv6 処理が有効になります。
		• インターフェイスの IPv6 アドレスを手動で設定します。
		• インターフェイスで IPv6 が有効な場合に自動設定されるリンクローカルアドレスでなく、インターフェイス上の特定のリンクローカルなアドレスを使用するように指定します。このコマンドにより、インターフェイス上で IPv6 処理が有効になります。

	Command or Action	Purpose
	(config-if)# **ipv6 enable**	• インターフェイスに IPv6 リンクローカルアドレスを自動設定し、インターフェイスでの IPv6 処理を有効にします。リンクローカルアドレスを使用できるのは、同じリンク上のノードと通信する場合だけです。
ステップ 9	**exit** **Example:** (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 10	**ipv6 unicast-routing** **Example:** (config)# **ipv6 unicast-routing**	IPv6 ユニキャスト データ パケットの転送を有効にします。
ステップ 11	**end** **Example:** (config)# **end**	特権 EXEC モードに戻ります。
ステップ 12	**show ipv6 interface** *interface-id* **Example:** # **show ipv6 interface gigabitethernet 1/0/1**	入力を確認します。
ステップ 13	**copy running-config startup-config** **Example:** # **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 と IPv6 の両方をサポートし、IPv6 ルーティングが有効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、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。

手順の概要

1. **enable**
2. **configure terminal**
3. **ip routing**
4. **ipv6 unicast-routing**
5. **interface** *interface-id*
6. **no switchport**
7. **ip address** *ip-address mask* [**secondary**]
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**
9. **end**
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*
11. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `> 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	**ip routing** 例： `(config)# ip routing`	スイッチ上でルーティングを有効にします。
ステップ 4	**ipv6 unicast-routing** 例： `(config)# ipv6 unicast-routing`	スイッチ上で IPv6 データ パケットの転送を有効にします。

	コマンドまたはアクション	目的
ステップ 5	**interface** *interface-id* 例： `(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： `(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： `(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**	• グローバル IPv6 アドレスを指定します。ネットワーク プレフィックスだけを指定します。最終の 64 ビットは、スイッチの MAC アドレスから自動的に計算されます。 • インターフェイスで IPv6 が有効な場合に自動設定されるリンクローカルアドレスでなく、インターフェイス上のリンクローカルアドレスを使用するように指定します。 • インターフェイスに IPv6 リンクローカルアドレスを自動設定し、インターフェイスでの IPv6 処理を有効にします。リンクローカルアドレスを使用できるのは、同じリンク上のノードと通信する場合だけです。 （注） インターフェイスから手動で設定したすべての IPv6 アドレスを削除するには、**no ipv6 address** インターフェイス コンフィギュレーション コマンドを引数なしで使用します。
ステップ 9	**end** 例： `(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*	入力を確認します。

	コマンドまたはアクション	目的
	• **show ip interface** *interface-id* • **show ipv6 interface** *interface-id*	
ステップ 11	**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。

デフォルト ルータ プリファレンス（DRP）の設定

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

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

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

インターフェイス上のルータの DRP を設定するには、次の手順を実行します。

Procedure

	Command or Action	Purpose
ステップ 1	**enable** **Example:** `> 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** **Example:** `# configure terminal`	グローバル コ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* **Example:** `(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**} **Example:**	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。

	Command or Action	Purpose
	(config-if)# **ipv6 nd router-preference medium**	
ステップ 5	**end** **Example:** (config)# **end**	特権 EXEC モードに戻ります。
ステップ 6	**show ipv6 interface** **Example:** # **show ipv6 interface**	設定を確認します。
ステップ 7	**copy running-config startup-config** **Example:** # **copy running-config startup-config**	（任意）コンフィギュレーション ファイルに設定を保存します。

IPv6 ICMP レート制限の設定

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

ICMP のレート制限パラメータを変更するには、次の手順を実行します。

Procedure

	Command or Action	Purpose
ステップ 1	**enable** **Example:** > **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** **Example:** # **configure terminal**	グローバル コ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*] **Example:**	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。

	Command or Action	Purpose
	`(config)# ipv6 icmp error-interval 50 20`	• *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** **Example:** `(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show ipv6 interface** [*interface-id*] **Example:** `# show ipv6 interface gigabitethernet0/1`	入力を確認します。
ステップ 6	**copy running-config startup-config** **Example:** `# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

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

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

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

Before you begin

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有効にし、**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 パケットの転送を有効にします。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 を有効にする必要があります。

Procedure

	Command or Action	Purpose
ステップ 1	**enable** **Example:** `> 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** **Example:** `# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ipv6 route** *ipv6-prefix/prefix length* {*ipv6-address* | *interface-id* [*ipv6-address*]} [*administrative distance*] **Example:** `(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

	Command or Action	Purpose
		10 進値です。10 進数値の前にスラッシュ記号が必要です。 • *ipv6-address*：指定したネットワークに到達するために使用可能なネクスト ホップの IPv6 アドレス。ネクスト ホップの IPv6 アドレスを直接接続する必要はありません。再帰処理が実行されて、直接接続されたネクスト ホップの IPv6 アドレスが検出されます。このアドレスはRFC 2373 に記載された形式（16 ビット値を使用したコロン区切りの 16 進表記で指定）で設定する必要があります。 • *interface-id*：Point-To-Point（ポイントツーポイント）インターフェイスおよびブロードキャスト インターフェイスからのダイレクト スタティック ルートを指定します。ポイントツーポイントインターフェイスの場合、ネクストホップの IPv6 アドレスを指定する必要はありません。ブロードキャストインターフェイスの場合は、常にネクストホップのIPv6アドレスを指定するか、または指定したプレフィックスをリンクに割り当てて、リンクローカルアドレスをネクストホップとして指定する必要があります。パケットの送信先となるネクストホップのIPv6アドレスを指定することもできます。 **Note** リンクローカルアドレスをネクストホップとして使用する場合は、*interface-id* を指定する必要があります（リンクローカルのネクストホップを隣接ルータに設定する必要もあります）。 • *administrative distance*：（任意）アドミニストレーティブ ディスタンス。指定できる範囲は 1 ～ 254 です。デフォルト値は 1 で、この場合、接続されたルートを除くその他のどのルート タイプよりも、スタティック ルートが優先します。フローティング スタティック ルートを設定する場合は、ダイナミック ルーティング プロトコルよりも大きなアドミニストレーティブ ディスタンスを使用します。

	Command or Action	Purpose
ステップ 4	**end** **Example:** `(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*]
	Example:
	`# show ipv6 static 2001:0DB8::/32 interface gigabitethernet2/0/1`
	または
	`# show ipv6 route static`
		IPv6 ルーティング テーブルの内容を表示して、設定を確認します。
		• **interface** *interface-id*：（任意）出力インターフェイスとして指定されたインターフェイスを含むスタティック ルートのみを表示します。
		• **recursive**：（任意）再帰スタティック ルートのみを表示します。**recursive** キーワードは **interface** キーワードと相互に排他的です。ただし、コマンド構文に IPv6 プレフィックスが指定されているかどうかに関係なく、使用できます。
		• **detail**：（任意）次に示す追加情報を表示します。
		• 有効な再帰ルートの場合、出力パス セットおよび最大分解深度
		• 無効なルートの場合、ルートが無効な理由

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

インターフェイスでの IPv6 PBR の有効化

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

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

いてルートを選択できるようにします。ルータは、ACL に基づいてパケットを分類し、ルーティング テーブルを選択し、宛先アドレスを検索し、パケットをルーティングします。

PBR for IPv6 を有効にするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] 例： `Device(config)# route-map rip-to-ospf permit`	ルーティング プロトコル間でルートを再配布する条件を定義するか、ポリシー ルーティングを有効にしてルート マップ コンフィギュレーション モードを開始します。

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

ステップ 5	次のいずれかを実行します。
	• **set ipv6 next-hop** *global-ipv6-address* [*global-ipv6-address...*]
	• **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。

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

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。

Before you begin

IPv6 RIP を実行するようにスイッチを設定する前に、**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有効にし、**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 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。

Procedure

	Command or Action	Purpose
ステップ 1	**enable** **Example:** `> 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** **Example:** `# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	Command or Action	Purpose
ステップ 3	**ipv6 router rip** *name* **Example:** `(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* **Example:** `(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** **Example:** `(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* **Example:** `(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** **Example:** `(config-if)# ipv6 rip cisco enable`	指定された IPv6 RIP ルーティング プロセスをインターフェイス上で有効にします。
ステップ 8	**ipv6 rip** *name* **default-information** {**only** | **originate**} **Example:** `(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。 **Note** 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

	Command or Action	Purpose
		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** **Example:** `(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**
	Example:
	`# show ipv6 rip cisco interface gigabitethernet 2/0/1`
	または
	`# 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** **Example:** `# 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。

Before you begin

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

次の注意事項に従ってください。

- IPv6 コマンドのデフォルト設定を変更する場合は注意してください。デフォルト設定を変更すると、IPv6 ネットワークの OSPF に悪影響が及ぶことがあります。
- インターフェイスで IPv6 OSPF を有効にする前に、**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有効にし、**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 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。

Procedure

	Command or Action	Purpose
ステップ 1	**enable** **Example:** `> 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** **Example:** `# configure terminal`	グローバル コ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* **Example:** `(config)# ipv6 router ospf 21`	プロセスに対して OSPF ルータ コンフィギュレーション モードを有効にします。プロセス ID は、IPv6 OSPF ルーティング プロセスを有効にする場合に管理上割り当てられる番号です。この ID はローカルに割り当てられ、1 ～ 65535 の正の整数を指定できます。
ステップ 4	**area** *area-id* **range** {*ipv6-prefix/prefix length*} [**advertise** | **not-advertise**] [**cost** *cost*] **Example:** `(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。

	Command or Action	Purpose
		• **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* **Example:** `(config)# maximum paths 16`	（任意）IPv6 OSPF がルーティング テーブルに入力する必要がある、同じ宛先への等コスト ルートの最大数を定義します。指定できる範囲は 1 ～ 32 で、デフォルトは 16 です。
ステップ 6	**exit** **Example:** `(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* **Example:** `(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	**router** *router-id* **Example:** `(config)# router ospfv3 1`	インターフェイス コンフィギュレーション モードに入り、設定するルータを指定します。
ステップ 9	**ipv6 ospf** *process-id* **area** *area-id* [**instance** *instance-id*] **Example:** `(config-if)# ipv6 ospf 21 area .3`	インターフェイスで IPv6 の OSPF を有効にします。 • **instance** *instance-id*：（任意）インスタンス ID
ステップ 10	**end** **Example:** `(config)# end`	特権 EXEC モードに戻ります。
ステップ 11	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*] **Example:**	• 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。

	Command or Action	Purpose

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

ステップ 12	**copy running-config startup-config** **Example:** `# 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。

IPv6 ユニキャスト リバース パス転送の設定

ユニキャスト リバース パス転送（ユニキャスト RPF）機能は、検証できない送信元 IP アドレスの IP パケットを廃棄することで、間違ったまたは偽造（スプーフィングされた）送信元 IP アドレスがネットワークに流れて発生する問題を軽減するのに役立ちます。たとえば、Smurf や Tribal Flood Network（TFN）など、多くの一般的なタイプの DoS 攻撃は、偽造された、または次々に変わる送信元 IP アドレスを使用して、攻撃を突き止めたりフィルタすることを攻撃者が阻止できるようにします。パブリック アクセスを提供するインターネット サービス プロバイダー（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。

DHCP for IPv6 アドレス割り当ての設定

この項では、DHCPv6 のアドレス割り当てについてだけ説明します。DHCPv6 クライアント、サーバ、またはリレー エージェント機能の設定の詳細については、Cisco.com で『*Cisco IOS IPv6 Configuration Library*』の「Implementing DHCP for IPv6」の章を参照してください。

DHCPv6 アドレス割り当てのデフォルト設定

デフォルトで、DHCPv6 機能はスイッチに設定されています。

DHCPv6 アドレス割り当ての設定時の注意事項

DHCPv6 アドレス割り当てを設定する場合は、次の注意事項に従ってください。

- 以下の手順では、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。
  - DHCPv6 IPv6 ルーティングは、レイヤ 3 インターフェイス上で有効である必要があります。
  - SVI：**interface vlan** *vlan_id* コマンドを使用して作成された VLAN 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** コマンドを使用して作成されたポートチャネル論理インターフェイス。
- スイッチは、DHCPv6 クライアント、サーバ、またはリレー エージェントとして動作できます。DHCPv6 クライアント、サーバ、およびリレー機能は、インターフェイスで相互に排他的です。
- DHCPv6 クライアント、サーバ、またはリレーエージェントは、アクティブスイッチ上でのみ稼働します。アクティブスイッチが再度選択された場合、新しいアクティブスイッチでは DHCPv6 設定が維持されます。ただし、DHCP サーバ データベース リース情報のローカルの RAM コピーは、維持されません。

DHCPv6 サーバ機能の有効化（CLI）

DHCPv6 プールの特性を変更するには、**no** 形式の DHCP プール コンフィギュレーション モードコマンドを使用します。インターフェイスに対して DHCPv6 サーバ機能を無効にするには、**no ipv6 dhcp server** 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。

インターフェイスで DHCPv6 サーバ機能を有効にするには、次の手順を実行します。

Procedure

	Command or Action	Purpose
ステップ 1	**enable** **Example:** `> 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** **Example:** `# configure terminal`	グローバル コ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 dhcp pool** *poolname* **Example:** `(config)# ipv6 dhcp pool 7`	DHCP プール コンフィギュレーション モードを開始して、IPv6 DHCP プールの名前を定義します。プール名は、記号文字列（Engineering など）または整数（0 など）です。
ステップ 4	**address prefix** *IPv6-prefix* {**lifetime**} {*t1 t1* | **infinite**} **Example:** `(config-dhcpv6)# address prefix 2001:1000::0/64 lifetime 3600`	（任意）アドレス割り当て用のアドレスプレフィックスを指定します。 このアドレスは、16 ビット値をコロンで区切った 16 進数で指定する必要があります。 **lifetime** *t1 t1*：IPv6 アドレス プレフィックスが有効な状態を維持するタイム インターバル（秒）を指定します。指定できる範囲は 5 ～ 4294967295 秒です。時間間隔なしの場合は、**infinite** を指定します。
ステップ 5	**link-address** *IPv6-prefix* **Example:** `(config-dhcpv6)# link-address 2001:1002::0/64`	（任意）link-address IPv6 プレフィックスを指定します。 着信インターフェイス上のアドレスまたはパケットのリンクアドレスが指定した IPv6 プレフィックスに一致する場合、サーバは設定情報プールを使用します。 このアドレスは、16 ビット値をコロンで区切った 16 進数で指定する必要があります。
ステップ 6	**vendor-specific** *vendor-id* **Example:** `(config-dhcpv6)# vendor-specific 9`	（任意）ベンダー固有のコンフィギュレーション モードを開始して、ベンダー固有の ID 番号を指定します。この番号は、ベンダーの IANA プライベート エンタープライズ番号です。指定できる範囲は 1 ～ 4294967295 です。

	Command or Action	Purpose
ステップ 7	**suboption** *number* { **address** *IPv6-address* | **ascii** *ASCII-string* | **hex** *hex-string*} **Example:** `(config-dhcpv6-vs)# suboption 1 address 1000:235D::`	（任意）ベンダー固有のサブオプション番号を入力します。指定できる範囲は 1 ～ 65535 です。IPv6 アドレス、ASCII テキスト、または 16 進文字列をサブオプション パラメータで定義されているように入力します。
ステップ 8	**exit** **Example:** `(config-dhcpv6-vs)# exit`	DHCP プール コンフィギュレーション モードに戻ります。
ステップ 9	**exit** **Example:** `(config-dhcpv6)# 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 10	**interface** *interface-id* **Example:** `(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 11	**ipv6 dhcp server** [*poolname* | **automatic**] [**rapid-commit**] [**preference** *value*] [**allow-hint**] **Example:** `(config-if)# ipv6 dhcp server automatic`	インターフェイスに対して DHCPv6 サーバ機能を有効にします。 • *poolname*：（任意）IPv6 DHCP プールのユーザ定義の名前。プール名は、記号文字列（Engineering など）または整数（0 など）です。 • **automatic**：（任意）サーバが、クライアントにアドレスを割り当てるときに使用するプールを自動的に決定できるようにします。 • **rapid-commit**：（任意）2 つのメッセージを交換する方式を許可します。 • **preference** 値：（任意）サーバによって送信されるアドバタイズメント メッセージ内のプリファレンス オプションで指定するプリファレンス値を設定します。範囲は 0 ～ 255 です。デフォルトのプリファレンス値は 0 です。 • **allow-hint**：（任意）サーバが SOLICIT メッセージに含まれるクライアントの提案を考慮す

	Command or Action	Purpose
		るかどうかを指定します。デフォルトでは、サーバはクライアントのヒントを無視します。
ステップ 12	**end** **Example:** `(config)# end`	特権 EXEC モードに戻ります。

ステップ 13	次のいずれかを実行します。
	• **show ipv6 dhcp pool**
	• **show ipv6 dhcp interface**
	Example:
	`# show ipv6 dhcp pool`
	または
	`# show ipv6 dhcp interface`
		• DHCPv6 プール設定を確認します。
		• DHCPv6 サーバ機能がインターフェイス上で有効であることを確認します。

ステップ 14	**copy running-config startup-config** **Example:** `# 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。

DHCPv6 クライアント機能の有効化

インターフェイスで DHCPv6 クライアントを有効にするには、次の手順を実行します。

Procedure

	Command or Action	Purpose
ステップ 1	**enable** **Example:** `> 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** **Example:** `# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	Command or Action	Purpose
ステップ 3	**interface** *interface-id* **Example:** `(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	**ipv6 address dhcp** [**rapid-commit**] **Example:** `(config-if)# ipv6 address dhcp rapid-commit`	インターフェイスで DHCPv6 サーバから IPv6 アドレスを取得できるようにします。 **rapid-commit**：（任意）アドレス割り当てに 2 つのメッセージを交換する方式を許可します。
ステップ 5	**ipv6 dhcp client request** [**vendor-specific**] **Example:** `(config-if)# ipv6 dhcp client request vendor-specific`	（任意）インターフェイスでベンダー固有のオプションを要求できるようにします。
ステップ 6	**end** **Example:** `(config)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show ipv6 dhcp interface** **Example:** `# show ipv6 dhcp interface`	DHCPv6 クライアントがインターフェイスで有効になっていることを確認します。

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。

Table 4: 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。

コマンド	目的
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

IPv6 アドレッシングの設定と IPv6 ルーティングの有効化：例

次に、IPv6 プレフィックス 2001:0DB8:c18:1::/64 に基づく、リンクローカルアドレスおよびグローバルアドレスを使用して、IPv6 を有効にする例を示します。EUI-64 インターフェイス ID が、両方のアドレスの下位 64 ビットで使用されます。**show ipv6 interface** EXEC コマンドの出力は、インターフェイスのリンクローカル プレフィックス FE80::/64 にインターフェイス ID（20B:46FF:FE2F:D940）を付加する方法を示すために追加されています。

```
(config)# ipv6 unicast-routing
(config)# interface gigabitethernet0/11

(config-if)# ipv6 address 2001:0DB8:c18:1::/64 eui 64
(config-if)# end
# show ipv6 interface gigabitethernet0/11
GigabitEthernet0/11 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::20B:46FF:FE2F:D940
  Global unicast address(es):
  2001:0DB8:c18:1:20B:46FF:FE2F:D940, subnet is 2001:0DB8:c18:1::/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
Hosts use stateless autoconfig for addresses.
```

デフォルト ルータ プリファレンスの設定：例

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。

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

IPv4 および IPv6 プロトコル スタックの設定：例

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

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

DHCPv6 サーバ機能の有効化：例

次の例では、*engineering* という IPv6 アドレス プレフィックスを持つプールを設定する方法を示します。

```
# configure terminal
(config)# ipv6 dhcp pool engineering
(config-dhcpv6)#address prefix 2001:1000::0/64
(config-dhcpv6)# end
```

次に、3 リンクアドレスおよび IPv6 アドレス プレフィックスを持つ *testgroup* と呼ばれるプールを設定する例を示します。

```
# configure terminal
(config)# ipv6 dhcp pool testgroup
(config-dhcpv6)# link-address 2001:1001::0/64
(config-dhcpv6)# link-address 2001:1002::0/64
(config-dhcpv6)# link-address 2001:2000::0/48
(config-dhcpv6)# address prefix 2001:1003::0/64
(config-dhcpv6)# end
```

次の例では、*350* というベンダー固有オプションを持つプールを設定する方法を示します。

```
# configure terminal
(config)# ipv6 dhcp pool 350
(config-dhcpv6)# address prefix 2001:1005::0/48
(config-dhcpv6)# vendor-specific 9
(config-dhcpv6-vs)# suboption 1 address 1000:235D::1
(config-dhcpv6-vs)# suboption 2 ascii "IP-Phone"
(config-dhcpv6-vs)# end
```

DHCPv6 クライアント機能の有効化：例

次に、IPv6 アドレスを取得して、rapid-commit オプションを有効にする例を示します。

```
(config)# interface gigabitethernet2/0/1
(config-if)# ipv6 address dhcp rapid-commit
```

IPv6 ICMP レート制限の設定：例

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。

```
(config)#ipv6 icmp error-interval 50 20
```

IPv6 のスタティック ルーティングの設定：例

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。

```
(config)# ipv6 route 2001:0DB8::/32 gigabitethernet 1/0/1 130
```

例：インターフェイスでの PBR の有効化

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

```
ipv6 access-list match-dest-1
  permit ipv6 any 2001:DB8:2001:1760::/32
route-map pbr-dest-1 permit 10
  match ipv6 address match-dest-1
  set interface GigabitEthernet 0/0/0
interface GigabitEthernet0/0/1
  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。

```
ipv6 access-list src-90
  permit ipv6 host 2001:DB8:2003::90 2001:DB8:2001:1000::/64
route-map pbr-src-90 permit 10
  match ipv6 address src-90
  set ipv6 next-hop 2001:DB8:2003:1::95
ipv6 local policy route-map pbr-src-90
```

IPv6 の RIP の設定：例

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

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

IPv6 の表示：例

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

```
# 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

関連資料

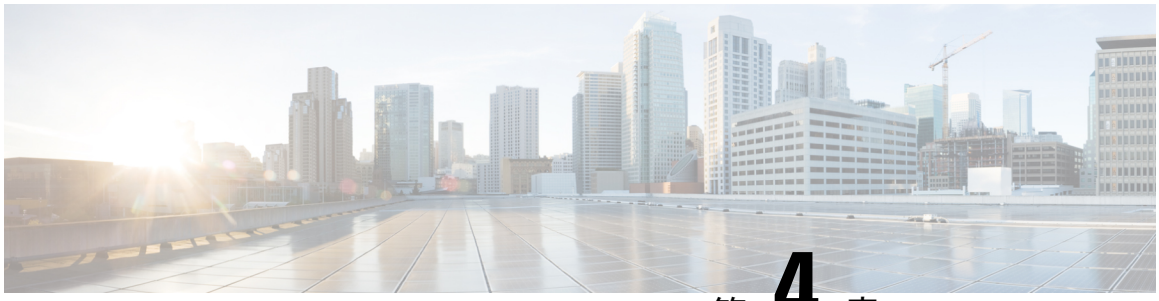
関連項目	マニュアル タイトル
Cisco IOS コマンド	『Cisco IOS Master Commands List, All Releases』

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

MIB

MIB	MIB のリンク
本リリースでサポートするすべての MIB	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs



第 4 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。

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。

（注） この製品のマニュアルセットは、偏向のない言語を使用するように配慮されています。このドキュメントセットでの偏向のない言語とは、年齢、障害、性別、人種的アイデンティティ、民族的アイデンティティ、性的指向、社会経済的地位、およびインターセクショナリティに基づく差別を意味しない言語として定義されています。製品ソフトウェアのユーザインターフェイスにハードコードされている言語、RFPのドキュメントに基づいて使用されている言語、または参照されているサードパーティ製品で使用されている言語によりドキュメントに例外が存在する場合があります。

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 *5: RIP* のデフォルト設定

機能	デフォルト設定
自動サマリー	有効
デフォルト情報送信元	無効
デフォルト メトリック	自動メトリック変換（組み込み）

機能	デフォルト設定
IP RIP 認証キーチェーン	認証なし 認証モード：クリア テキスト
IP RIP の起動	無効
IP スプリット ホライズン	メディアにより異なる
Neighbor	未定義
ネットワーク	指定なし
オフセット リスト	無効
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	有効
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 ルーティングを有効にします。他のパラメータを設定することもできます。スイッチでは、ネットワーク番号を設定するまでRIP コンフィギュレーション コマンドは無視されます。

手順

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

	コマンドまたはアクション	目的
	`Device# configure terminal`	
ステップ 3	**ip routing** 例： `Device(config)# ip routing`	IP ルーティングを有効にします。（IP ルーティングが無効になっている場合だけ、必須です）。
ステップ 4	**router rip** 例： `Device(config)# router rip`	RIP ルーティング プロセスを有効にし、ルータ コンフィギュレーション モードを開始します。
ステップ 5	**network** *network number* 例： `Device(config-router)# network 12.0.0.0`	ネットワークを RIP ルーティング プロセスと関連付けます。複数の **network** コマンドを指定できます。RIP ルーティング アップデートの送受信は、これらのネットワークのインターフェイスを経由する場合だけ可能です。 （注） RIP コマンドを有効にするには、ネットワーク番号を設定する必要があります。
ステップ 6	**neighbor** *ip-address* 例： `Device(config-router)# neighbor 10.2.5.1`	（任意）ルーティング情報を交換する隣接ルータを定義します。このステップを使用すると、RIP（通常はブロードキャストプロトコル）からのルーティング アップデートが非ブロードキャスト ネットワークに到達するようになります。
ステップ 7	**offset-list** [*access-list number* | *name*] {**in** | **out**} *offset* [*type number*] 例： `Device(config-router)# offset-list 103 in 10`	（任意）オフセットリストをルーティング メトリックに適用し、RIP によって取得したルートへの着信および発信メトリックを増加します。アクセス リストまたはインターフェイスを使用し、オフセット リストを制限できます。
ステップ 8	**timers basic** *update invalid holddown flush* 例： `Device(config-router)# timers basic 45 360 400 300`	（任意）ルーティング プロトコル タイマーを調整します。すべてのタイマーの有効範囲は 0 ～ 4294967295 秒です。 • *update*：ルーティング アップデートの送信間隔。デフォルトは 30 秒です。 • *invalid*：ルートが無効と宣言されるまでの時間。デフォルト値は 180 秒です。 • *holddown*：ルートがルーティング テーブルから削除されるまでの時間。デフォルト値は 180 秒です。

	コマンドまたはアクション	目的
		• *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）自動要約を無効にします。デフォルトでは、クラスフル ネットワーク境界を通過するときにサブプレフィックスがサマライズされます。サマライズを無効にし（RIP バージョン 2 だけ）、クラスフル ネットワーク境界にサブネットおよびホスト ルーティング情報をアドバタイズします。
ステップ 11	**output-delay** *delay* 例： `Device(config-router)# output-delay 8`	（任意）送信する RIP アップデートにパケット間遅延を追加します。デフォルトでは、複数のパケットからなる RIP アップデートのパケットに、パケット間遅延が追加されません。パケットを低速なデバイスに送信する場合は、8 ～ 50 ミリ秒のパケット間遅延を追加できます。
ステップ 12	**end** 例： `Device(config-router)# end`	特権 EXEC モードに戻ります。
ステップ 13	**show ip protocols** 例： `Device# show ip protocols`	入力を確認します。
ステップ 14	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

RIP 認証の設定

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

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するインターフェイスを指定します。
ステップ 4	**ip rip authentication key-chain** *name-of-chain* 例： `Device(config-if)# ip rip authentication key-chain trees`	RIP 認証を有効にします。
ステップ 5	**ip rip authentication mode** {**text** | **md5**} 例： `Device(config-if)# ip rip authentication mode md5`	プレーン テキスト認証（デフォルト）または MD5 ダイジェスト認証を使用するように、インターフェイスを設定します。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
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。

Before you begin

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。

Procedure

	Command or Action	Purpose
ステップ 1	**enable** **Example:** `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** **Example:** `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* **Example:** `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* **Example:** `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。

	Command or Action	Purpose
ステップ 5	**exit** **Example:** `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* **Example:** `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** **Example:** `Device(config-if)# ipv6 rip cisco enable`	指定された IPv6 RIP ルーティング プロセスをインターフェイス上で有効にします。
ステップ 8	**ipv6 rip** *name* **default-information** {**only** | **originate**} **Example:** `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。 **Note** 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** **Example:** `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** **Example:** `Device# show ipv6 rip cisco interface gigabitethernet 2/0/1`	• 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。

	Command or Action	Purpose
	または `Device# show ipv6 rip`	
ステップ 11	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

サマリー アドレスおよびスプリット ホライズンの設定

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

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

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

手順

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

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

スプリット ホライズンの設定

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

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

手順

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

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

例：IPv6 用の RIP の設定

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

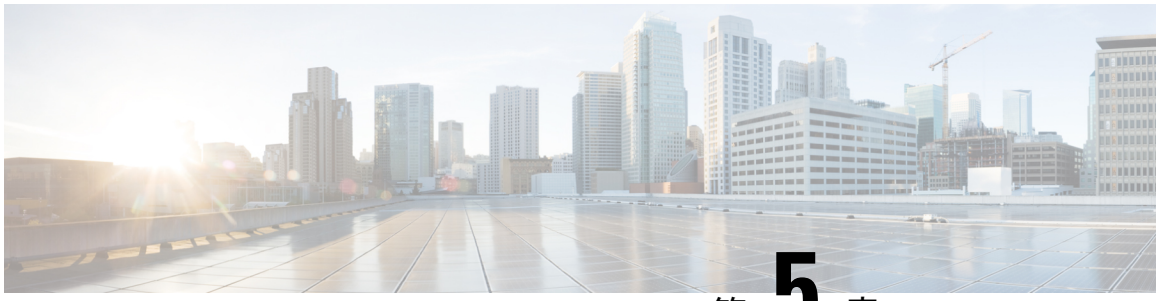
```
Device> enable
Devce# configure terminal
Device(config)# ipv6 router rip cisco
Device(config-router)# maximum-paths 8
Device(config)# exit
Device(config)# interface gigabitethernet2/0/11
Device(config-if)# ipv6 rip cisco enable
```

サマリーアドレスおよびスプリットホライズンの設定例

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

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

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



第 5 章

VRF-Lite の設定

VRF-Lite について

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

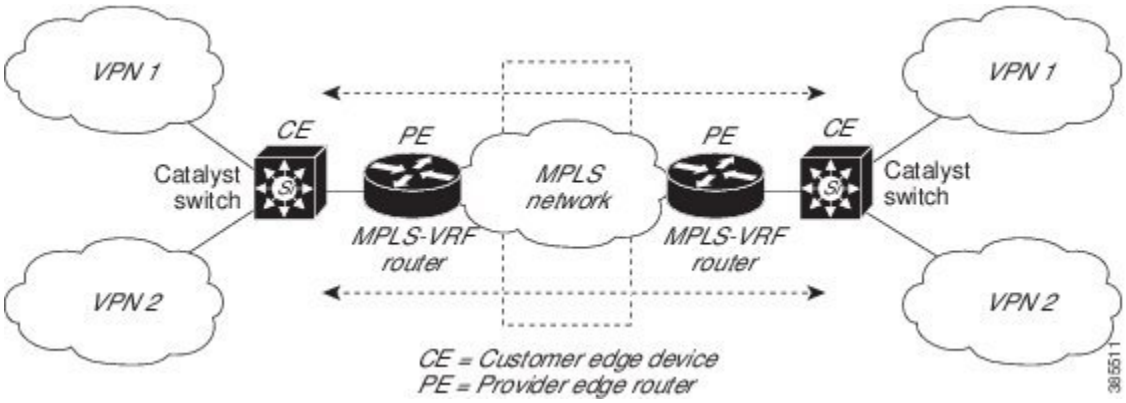
（注） VRF-Lite インターフェイスは、レイヤ 3 インターフェイスである必要があります。

VRF-Lite には次のデバイスが含まれます。

- カスタマーエッジ（CE）デバイスにおいて、カスタマーは、1 つまたは複数の プロバイダー エッジ（PE）ルータへのデータ リンクを介してサービス プロバイダー ネットワークにアクセスできます。CE デバイスは、サイトのローカル ルートをプロバイダー エッジ ルータにアドバタイズし、そこからリモート VPN ルートを学習します。Cisco Catalyst スイッチは、CE にすることができます。
- プロバイダー ルータ（またはコア ルータ）とは、サービス プロバイダー ネットワーク内にあり、CE デバイスに接続していないすべてのルータです。

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

図 *2:* 複数の仮想 *CE* として機能する *Cisco Catalyst* スイッチ



次の図に、VRF-Lite の CE 対応ネットワークでのパケット転送プロセスを示します。

- CE が VPN からパケットを受信すると、CE は入力インターフェイスに基づいたルーティング テーブルを検索します。ルートが見つかると、CE はパケットを PE に転送します。
- 入力 PE は、CE からパケットを受信すると、VRF 検索を実行します。ルートが見つかると、ルータは対応する MPLS ラベルをパケットに追加し、MPLS ネットワークに送信します。
- 出力 PE は、ネットワークからパケットを受信すると、ラベルを除去してそのラベルを使用し、正しい VPN ルーティング テーブルを識別します。次に、出力 PE が通常のルート検索を行います。ルートが見つかると、パケットを正しい隣接デバイスに転送します。
- CE が出力 PE からパケットを受信すると、CE は入力インターフェイスを使用して正しい VPN ルーティング テーブルを検索します。ルートが見つかると、CE はパケットを VPN 内に転送します。

VRF を設定するには、VRF テーブルを作成し、VRF に対応付けられたレイヤ 3 インターフェイスを指定します。次に、VPN および CE と PE 間でルーティング プロトコルを設定します。プロバイダーのバックボーンで VPN ルーティング情報を配信する場合は、BGP が優先ルーティング プロトコルです。VRF-Lite ネットワークには、次の 3 つの主要なコンポーネントがあります。

- VPN ルート ターゲット コミュニティ：VPN コミュニティの他のすべてのメンバをリストします。VPN コミュニティ メンバーごとに VPN ルート ターゲットを設定する必要があります。
- VPN コミュニティ PE ルータのマルチプロトコル BGP ピアリング： VPN コミュニティのすべてのメンバに VRF の到着可能性情報を伝播します。VPN コミュニティのすべての PE ルータで BGP ピアリングを設定する必要があります。

- VPN 転送：VPN サービスプロバイダー ネットワークのすべての VPN コミュニティ メンバ間のすべてのトラフィックを転送します。

（注） この製品のマニュアルセットは、偏向のない言語を使用するように配慮されています。このドキュメントセットでの偏向のない言語とは、年齢、障害、性別、人種的アイデンティティ、民族的アイデンティティ、性的指向、社会経済的地位、およびインターセクショナリティに基づく差別を意味しない言語として定義されています。製品ソフトウェアのユーザインターフェイスにハードコードされている言語、RFP のドキュメントに基づいて使用されている言語、または参照されているサードパーティ製品で使用されている言語によりドキュメントに例外が存在する場合があります。

VRF-Lite の設定に関するガイドライン

IPv4 と IPv6

- VRF-Lite が設定されたスイッチは複数のカスタマーで共有され、すべてのカスタマーが独自のルーティング テーブルを持ちます。
- カスタマーは別々の VRF テーブルを使用するので、同じ IP アドレスを再利用できます。別々の VPN では IP アドレスの重複が許可されます。
- VRF-Lite では、複数のカスタマーが PE と CE の間で同一の物理リンクを共有できます。複数の VLAN を持つトランク ポートでは、パケットがカスタマー間で分離されます。すべてのカスタマーが独自の VLAN を持ちます。
- PE ルータでは、VRF-Lite の使用と複数の CE の使用には違いがありません。#unique_103 では、複数の仮想レイヤ 3 インターフェイスが VRF-Lite デバイスに接続されています。
- Cisco Catalyst スイッチでは、物理ポートか VLAN SVI、またはその両方の組み合わせを使用して、VRF を設定できます。アクセス ポートまたはトランク ポート経由で SVI を接続できます。
- カスタマーは、別のカスタマーと重複しないかぎり、複数の VLAN を使用できます。カスタマーの VLAN は、スイッチに保存されている適切なルーティング テーブルの識別に使用される特定のルーティング テーブル ID にマッピングされます。
- レイヤ 3 TCAM リソースは、すべての VRF 間で共有されます。各 VRF が十分な CAM 領域を持つようにするには、**maximum routes** コマンドを使用します。
- VRF を使用した Cisco Catalyst スイッチは、1 つのグローバル ネットワークと複数の VRF をサポートできます。サポートされるルートの総数は、TCAM のサイズに制限されます。
- 1 つの VRF を IPv4 と IPv6 の両方に設定できます。

- 着信パケットの宛先アドレスが VRF テーブルにない場合、そのパケットはドロップされます。また、VRF ルートに TCAM 領域が十分にない場合、その VRF のハードウェアスイッチングは無効になり、対応するデータ パケットがソフトウェアに送信されて処理されます。

IPv4 固有

- CE と PE 間のほとんどのルーティング プロトコル（BGP、OSPF、EIGRP、RIP、およびスタティック ルーティング）を使用できます。ただし、次の理由から External BGP（EBGP）を使用することを推奨します。
  - BGP では、複数の CE とのやり取りに複数のアルゴリズムを必要としません。
  - BGP は、さまざまな管理者によって稼働するシステム間でルーティング情報を渡すように設計されています。
  - BGP は、ルートの属性の CE への引き渡しを単純化します。
- マルチキャスト VRF-Lite はサポートされていません。
- **router ospf** の **capability vrf-lite** サブコマンドは、PE と CE 間のルーティング プロトコルとして OSPF が設定されている場合に使用する必要があります。

IPv6 固有

- VRF 認識 OSPFv3、BGPv6、EIGRPv6、および IPv6 スタティックルーティングがサポートされます。
- VRF 認識 IPv6 ルート アプリケーションには、ping、telnet、ssh、tftp、ftp、およびトレースルートが含まれています（このリストには Mgt 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

IPv4 用の VRF-Lite の設定

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： `Switch# 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： `Switch(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： `Switch> 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： `Switch# configure terminal`	グローバル コ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： `Switch(config)# ip vrf 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： `Switch (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： `Switch (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： `Switch (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： `Switch (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： `Switch (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： `Switch (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： `Switch (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： `Switch (config-sg-tacacs+)# server-private 10.1.1.1 port 19 key cisco`	グループ サーバに対するプライベート TACACS+ サーバの IP アドレスを設定します。

	コマンドまたはアクション	目的
ステップ 12	**vrf forwarding** *vrf-name* 例： `Switch (config-sg-tacacs+)# vrf forwarding vrf-name`	AAA TACACS+ サーバ グループの VRF リファレンスを設定します。
ステップ 13	**ip tacacs source-interface** *subinterface-name* 例： `Switch (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： `Switch (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。

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

VPN ルーティング セッションの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Switch# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**router ospf** *process-id* **vrf vrf-name** 例： `Switch(config)# router ospf process-id vrf vrf-name`	OSPF ルーティングを有効にし、VPN 転送テーブルを指定して、ルータ コンフィギュレーション モードを開始します。
ステップ 3	**capability vrf-lite**	

	コマンドまたはアクション	目的
	例： `Switch(config-router)# capability vrf-lite`	
ステップ 4	**log-adjacency-changes** 例： `Switch(config-router)# log-adjacency-changes`	（任意）隣接状態（デフォルト）の変更を記録します。
ステップ 5	**redistribute bgp** *autonomous-system-number subnets* 例： `Switch(config-router)# redistribute bgp autonomous-system-number subnets`	BGP ネットワークから OSPF ネットワークに情報を再配布するようにスイッチを設定します。
ステップ 6	**network** *network-number* **area** *area-id* 例： `Switch(config-router)# network network-number area area-id`	OSPF が動作するネットワーク アドレスとマスク、およびそのネットワーク アドレスのエリア ID を定義します。
ステップ 7	**end** 例： `Switch(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： `Switch# show ip ospf process-id`	OSPF ネットワークの設定を確認します。
ステップ 9	copy running-config startup-config 例： `Switch# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。 VPN 転送テーブルと OSPF ルーティング プロセスの関連付けを解除するには、**no router ospf** *process-id* **vrf** *vrf-name* グローバル コンフィギュレーション コマンドを使用します。

例

```
Switch(config)# vrf definition VRF-RED
Switch(config-vrf)# rd 1:1
Switch(config-vrf)# exit
Switch(config)# router eigrp virtual-name
Switch(config-router)# address-family ipv4 vrf VRF-RED autonomous-system 1
Switch(config-router-af)# network 10.0.0.0 0.0.0.255
Switch(config-router-af)# topology base
Switch(config-router-topology)# default-metric 10000 100 255 1 1500
Switch(config-router-topology)# exit-af-topology
Switch(config-router-af)# exit-address-family
```

BGP PE/CE ルーティング セッションの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Switch# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**router bgp** *autonomous-system-number* 例： `Switch(config)# router bgp autonomous-system-number`	その他の BGP ルータに渡された AS 番号で BGP ルーティング プロセスを設定し、ルータ コンフィギュレーション モードを開始します。
ステップ 3	**network** *network-number* **mask** *network-mask* 例： `Switch(config-router)# network network-number mask network-mask`	BGP を使用してアナウンスするネットワークおよびマスクを指定します。
ステップ 4	**redistribute ospf** *process-id match internal* 例： `Switch(config-router)# redistribute ospf process-id match internal`	OSPF 内部ルートを再配布するようにスイッチを設定します。
ステップ 5	**network** *network-number* **area** *area-id* 例： `Switch(config-router)# network network-number area area-id`	OSPF が動作するネットワーク アドレスとマスク、およびそのネットワーク アドレスのエリア ID を定義します。
ステップ 6	**address-family ipv4 vrf** *vrf-name* 例： `Switch(config-router-af)# address-family ipv4 vrf vrf-name`	PE から CE のルーティング セッションの BGP パラメータを定義し、VRF アドレス ファミリ モードを開始します。
ステップ 7	**neighbor** *address* **remote-as** *as-number* 例： `Switch(config-router-af)# neighbor address remote-as as-number`	PE と CE ルータの間の BGP セッションを定義します。
ステップ 8	**neighbor** *address* **activate** 例： `Switch(config-router-af)# neighbor address activate`	IPv4 アドレス ファミリのアドバタイズメントをアクティブ化します。
ステップ 9	**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
	`Switch(config-router-af)# end`	
ステップ 10	**show ip bgp** [**ipv4**] [**neighbors**] 例： `Switch# show ip bgp [ipv4] [neighbors]`	BGP 設定を確認します。 BGP ルーティングプロセスを削除するには、**no router bgp** *autonomous-system-number* グローバル コンフィギュレーション コマンドを使用します。ルーティング特性を削除するには、コマンドにキーワードを指定してこのコマンドを使用します。

IPv4 VRF の設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Switch# configure terminal`	グローバル 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： `Switch# configure terminal`	グローバル コ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： `Switch(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： `Switch(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： `Switch(config-vrf)# route-target {export | import | both} route-target-ext-community`	指定された VRF のインポート、エクスポート、またはインポートおよびエクスポート ルート ターゲット コミュニティのリストを作成します。AS システム番号と任意の番号（xxx:y）またはIP アドレスと任意の番号（A.B.C.D:y）を入力します。 （注） このコマンドは、BGP が動作している場合にのみ有効です。
ステップ 6	**import map** ルート マップ 例： `Switch(config-vrf)# import map route-map`	（任意）VRF にルート マップを対応付けます。

	コマンドまたはアクション	目的
ステップ 7	**interface** *interface-id* 例： `Switch(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： `Switch(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： `Switch(config-if)# end`	特権 EXEC モードに戻ります。
ステップ 10	**show ip vrf** [**brief** | **detail** | **interfaces**] [*vrf-name*] 例： `Switch# show ip 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 11	**copy running-config startup-config** 例： `Switch# 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。 （注） 次のコマンドの完全な構文と使用方法については、このリリースに対応するスイッチ コマンド リファレンスおよび『Cisco IOS Switching Services Command Reference』を参照してください。 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

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： `Switch# ping vrf vrf-name ipv6-host`	指定された VRF で、IPv6 ホストまたはアドレスに対して ping を実行します。

uRPF のユーザ インターフェイスの設定

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

手順の概要

1. **configure terminal**
2. **interface** *interface-id*
3. **no switchport**
4. **vrf forwarding** *vrf-name*
5. **ipv6 address** *ip-address***subnet-mask**
6. **ipv6 verify unicast source reachable-via rx allow-default**
7. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Switch# configure terminal`	グローバル 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： `Switch (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： `Switch (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： `Switch (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： `Switch (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： `Switch(config-if)# ipv6 verify unicast source reachable-via rx allow-default`	インターフェイス上で uRPF を有効にします。
ステップ 7	**end** 例： `Switch(config-if)# end`	特権 EXEC モードに戻ります。

Traceroute のユーザ インターフェイスの設定

手順の概要

1. **traceroute vrf** *vrf-name ipv6address*

手順の詳細

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

Telnet および SSH のユーザ インターフェイスの設定

手順の概要

1. **telnet** *ipv6-address*/ **vrf** *vrf-name*
2. **ssh -l** *username* **-vrf** *vrf-name* **ipv6-host**

手順の詳細

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

NTP のユーザ インターフェイスの設定

手順の概要

1. **configure terminal**
2. **ntp server vrf** *vrf-name* **ipv6-host**
3. **ntp peer vrf** *vrf-name* **ipv6-host**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**ntp server vrf** *vrf-name* **ipv6-host** 例： `(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： `(config)# ntp peer vrf vrf-name ipv6-host`	指定された VRF で NTP ピアを設定します。

IPv6 VRF の設定

手順の概要

1. **configure terminal**
2. **vrf definition** *vrf-name*
3. **rd** *route-distinguisher*
4. **address-family** *ipv4* | *ipv6*
5. **route-target** {**export** | **import** | **both**} *route-target-ext-community*
6. **exit-address-family**
7. **vrf definition** *vrf-name*
8. **ipv6 multicast multitopology**
9. **address-family ipv6 multicast**
10. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Switch# configure terminal`	グローバル 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： `Switch(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： `Switch(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： `Switch(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： `Switch(config-vrf)# route-target {export | import | both} route-target-ext-community`	指定された VRF のインポート、エクスポート、またはインポートおよびエクスポート ルート ターゲット コミュニティのリストを作成します。AS システム番号と任意の番号（xxx:y）または IP アドレスと任意の番号（A.B.C.D:y）を入力します。 （注） このコマンドは、BGP が動作している場合にのみ有効です。

	コマンドまたはアクション	目的
ステップ 6	**exit-address-family** 例： `Switch(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： `Switch(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： `Switch(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： `Switch(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： `Switch(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。

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

定義済み VRF へのインターフェイスの関連付け

手順の概要

1. **interface** *interface-id*
2. **no switchport**
3. **vrf forwarding** *vrf-name*
4. **ipv6 enable**
5. **ipv6 address** *ip-address subnet-mask*

6. **show ipv6 vrf** [**brief** | **detail** | **interfaces**] [*vrf-name*]
7. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**interface** *interface-id* 例： `Switch(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： `Switch(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： `Switch(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： `Switch(config-if)# ipv6 enable`	インターフェイスで IPv6 を有効にします。
ステップ 5	**ipv6 address** *ip-address subnet-mask* 例： `Switch(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： `Switch# 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： `Switch# 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

VRF スタティック ルートの設定

手順の概要

1. **configure terminal**
2. **ipv6 route** [**vrf** *vrf-name*] *ipv6-prefix*/*prefix-length* {*ipv6-address* | **interface-type** *interface-number* [*ipv6-address*]}

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Switch# configure terminal`	グローバル 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： `Switch(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

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

OSPFv3 ルータ プロセスの設定

手順の概要

1. **configure terminal**
2. **router ospfv3** *process-id*
3. **area** *area-ID* [**default-cot** | **nssa** | **stub**]
4. **router-id** *router-id*
5. **address-family ipv6 unicast vrf** *vrf-name*
6. **redistribute source-protocol** [*process-id*] **options**
7. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Switch# configure terminal`	グローバル 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： `Switch(config)# router ospfv3 process-id`	IPv6 アドレス ファミリの OSPFv3 ルータ コンフィギュレーション モードを有効にします。
ステップ 3	**area** *area-ID* [**default-cot** | **nssa** | **stub**] 例： `Switch(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： `Switch(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： `Switch(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： `Switch(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： `Switch(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。

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

インターフェイス上での OSPFv3 の有効化

手順の概要

1. **configure terminal**
2. **interface** *type-number*
3. **ospfv3** *process-id* **area** *area-id* **ipv6** [**instance** *instance-id*]
4. **end**

手順の詳細

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

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

EIGRPv6 ルーティング プロセスの設定

手順の概要

1. **configure terminal**
2. **router eigrp** *virtual-instance-name*
3. **address-family ipv6 vrf** *vrf-name* **autonomous-system** *autonomous-system-number*

4. **topology** {**base** | **topology-name tid number**
5. **exit-aftopology**
6. **eigrp router-id** *ip-address*
7. end

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Switch# configure terminal`	グローバル 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： `Switch(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： `Switch(config-router)# address-family ipv6 vrf vrf-name autonomous-system autonomous-system-number`	EIGRP IPv6 VRF-Lite を有効にし、アドレス ファミリ コンフィギュレーション モードを開始します。
ステップ 4	**topology** {**base**	**topology-name tid number** 例： `Switch(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： `Switch(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： `Switch(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： `Switch(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。

```
Switch(config)# router eigrp test
```

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

EBGPv6 ルーティング プロセスの設定

手順の概要

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

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Switch# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**router bgp** *as-number* 例： `Switch(config)# router bgp as-number`	指定したルーティング プロセスのルータ コンフィギュレーション モードを開始します。
ステップ 3	**neighbor** *peer-group-name peer-group* 例： `Switch(config-router)# neighbor peer-group-name peer-group`	マルチプロトコルBGP ピア グループを作成します。
ステップ 4	**neighbor** {*ip-address* | *ipv6-address*[*%*] | *peer-group-name*} **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number* ...] 例： `Switch(config-router)# neighbor {ip-address | ipv6-address[%] | peer-group-name}remote-as autonomous-system-number [alternate-as autonomous-system-number ...]`	指定した自律システム内のネイバーのIPv6アドレスを、ローカル ルータの IPv6 マルチプロトコル BGP ネイバーテーブルに追加します。

	コマンドまたはアクション	目的
ステップ 5	**address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast** | **vpnv6**] 例： `Switch(config-router)# address-family ipv6 [vrf vrf-name] [unicast | multicast | vpnv6]`	IPv6 アドレス ファミリを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。 • unicast キーワードは、IPv6 ユニキャスト アドレス ファミリを指定します。デフォルトでは、address-family ipv6 コマンドでユニキャスト キーワードが指定されていない場合、スイッチは IPv6 ユニキャスト アドレス ファミリのコンフィギュレーション モードになります。 • multicast キーワードは、IPv6 マルチキャスト アドレス プレフィックスを指定します。
ステップ 6	**neighbor** *ipv6-address* **peer-group** *peer-group-name* 例： `Switch(config-router-af)# neighbor ipv6-address peer-group peer-group-name`	BGP ネイバーの IPv6 アドレスをピア グループに割り当てます。
ステップ 7	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*[*%*]} **route-map** *map-name* {**in** | **out**} 例： `Switch(config-router-af)# neighbor {ip-address | peer-group-name | ipv6-address[%]}route-map map-name {in | out}`	着信ルートまたは発信ルートにルートマップを適用します。ルートマップへの変更は、ピアリングがリセットされるまで、またはソフト リセットが実行されるまで、現在のピアでは有効になりません。soft キーワードと in キーワードを指定して clear bgp ipv6 コマンドを使用すると、ソフト リセットが実行されます。
ステップ 8	exit 例： `Switch(config-router-af)# exit`	アドレス ファミリ コンフィギュレーション モードを終了し、ルータをルータ コンフィギュレーション モードに戻します。

例

次に、EBRPv6 を設定する例を示します。

```
Switch(config)# router bgp 2
Switch(config-router)# bgp router-id 2.2.2.2
Switch(config-router)# bgp log-neighbor-changes
Switch(config-router)# no bgp default ipv4-unicast
Switch(config-router)# neighbor 2500::1 remote-as 1
Switch(config-router)# neighbor 4000::2 remote-as 3
Switch(config-router)# address-family ipv6 vrf b1
Switch(config-router-af)# network 2500::/64
Switch(config-router-af)# network 4000::/64
Switch(config-router-af)# neighbor 2500::1 remote-as 1
Switch(config-router-af)# neighbor 2500::1 activate
Switch(config-router-af)# neighbor 4000::2 remote-as 3
Switch(config-router-af)# neighbor 4000::2 activate
Switch(config-router-af)# exit-address-family
```

VRF-Lite に関する追加情報

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
!
ip vrf 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

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
`Switch# show ip protocols vrf vrf-name`	VRF に対応付けられたルーティング プロトコル情報を表示します。

コマンド	目的
Switch# **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。
Switch# **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。
Switch# **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
```

IPv6 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
	VRF に対応付けられたルーティング プロトコル情報を表示します。

コマンド	目的
Switch# **show ipv6 mfib vrf** *instance-name a.b.c.d* | **active** | **all** | **count** | **linkscope**| **route** | **summary** | **update-sets** | **verbose**	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。

```
show ipv6 mroute vrf vrf1 FF05:ABCD:0:0:0:0:0:1
Multicast Routing Table
Flags: S - Sparse, B - Bidir Group, s - SSM Group,
C - Connected, L - Local, I - Received Source Specific Host Report,
P - Pruned, R - RP-bit set, F - Register flag, T - SPT-bit set,
J - Join SPT, Y - Joined MDT-data group,
y - Sending to MDT-data group

g - BGP signal originated, G - BGP Signal received,
N - BGP Shared-Tree Prune received, n - BGP C-Mroute suppressed,
q - BGP Src-Active originated, Q - BGP Src-Active received
E - Extranet
Timers: Uptime/Expires
Interface state: Interface, State

(*, FF05:ABCD::1), 00:06:22/never, RP 1010:ABCD::10, flags: SCJ
Incoming interface: Port-channel33
RPF nbr: FE80::2E31:24FF:FE06:134A
Immediate Outgoing interface list:
TenGigabitEthernet4/0/18, Forward, 00:06:22/never

(3232:ABCD::2, FF05:ABCD::1), 00:04:54/00:02:16, flags: SJT
Incoming interface: Port-channel33
RPF nbr: FE80::2E31:24FF:FE06:134A
Inherited Outgoing interface list:
TenGigabitEthernet4/0/18, Forward, 00:06:22/never
```

次に、**show ipv6 mfib** コマンドの出力例を示します。

```
Switch# show ipv6 mfib vrf vrf1 FF05:ABCD:0:0:0:0:0:1
Entry Flags:    C - Directly Connected, S - Signal, IA - Inherit A flag,
                ET - Data Rate Exceeds Threshold, K - Keepalive
                DDE - Data Driven Event, HW - Hardware Installed
                ME - MoFRR ECMP entry, MNE - MoFRR Non-ECMP entry, MP - MFIB
                MoFRR Primary, RP - MRIB MoFRR Primary, P - MoFRR Primary
                MS  - MoFRR  Entry in Sync, MC - MoFRR entry in MoFRR Client.
I/O Item Flags: IC - Internal Copy, NP - Not platform switched,
                NS - Negate Signalling, SP - Signal Present,
                A - Accept, F - Forward, RA - MRIB Accept, RF - MRIB Forward,
                MA - MFIB Accept, A2 - Accept backup,
                RA2 - MRIB Accept backup, MA2 - MFIB Accept backup

Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kbits per second
Other counts:      Total/RPF failed/Other drops
I/O Item Counts:   FS Pkt Count/PS Pkt Count
VRF testvrf1
 (*,FF05:ABCD::1) Flags: C HW
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   HW Forwarding:   295/0/512/0, Other: 0/0/0
   Port-channel33 Flags: A NS
   TenGigabitEthernet4/0/18 Flags: F NS
     Pkts: 0/0
 (3232:ABCD::2,FF05:ABCD::1) Flags: HW
```

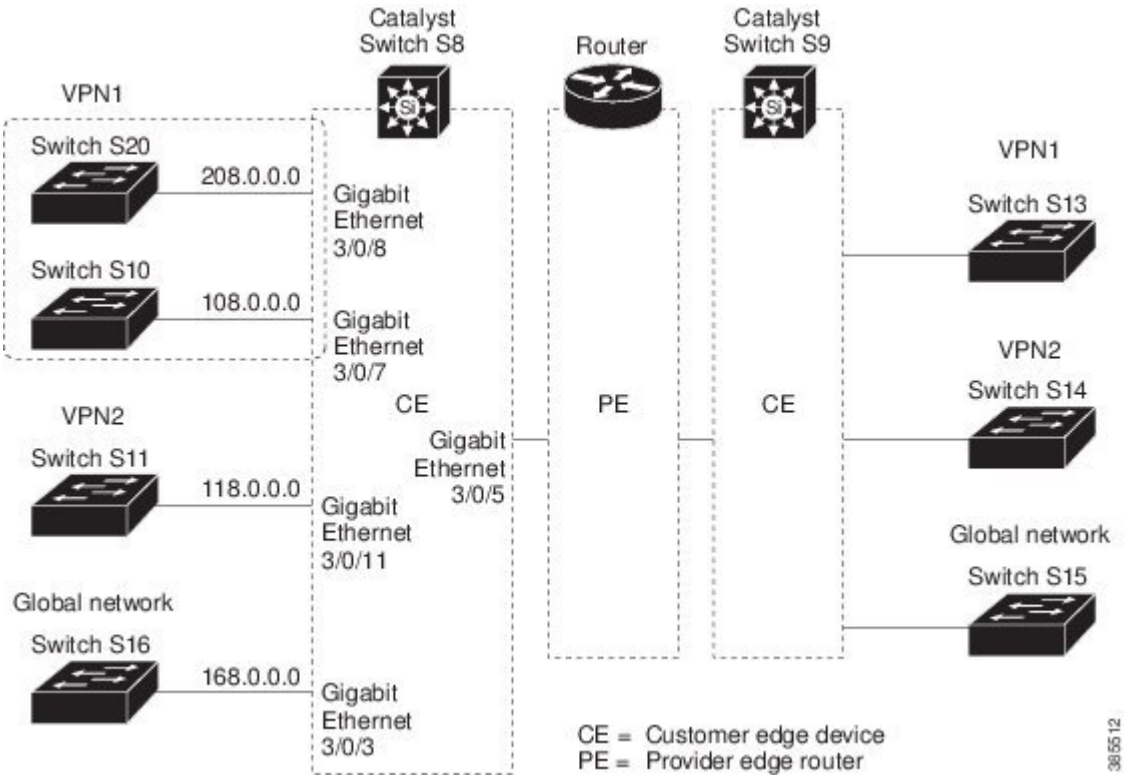
```
    SW Forwarding: 50/0/512/0, Other: 111/0/111
    HW Forwarding:   4387686/14849/512/59398, Other: 0/0/0
    Port-channel33 Flags: A
    TenGigabitEthernet4/0/18 Flags: F NS
      Pkts: 0/50

Switch#
```

VRF-Lite の設定例

IPv4 VRF-Lite の設定例

VPN1、VPN2、およびグローバル ネットワークで使用されるプロトコルは OSPF です。CE/PE 接続には BGP が使用されます。後続のコマンド例では、CE スイッチ S8 を設定する方法が示されており、スイッチ S20 と S11 の VRF 設定、およびスイッチ S8 のトラフィックに関連する PE ルータコマンドが含まれています。その他のスイッチの設定のコマンドは含まれていませんが、類似したものになります。

図 *3 : VRF-Lite* の設定例



スイッチ S8 の設定

スイッチ S8 上のルーティングを有効にし、VRF を設定します。

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

スイッチ S8 上で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。ファスト イーサネット インターフェイス 3/5 は、PE へのトランク接続です。インターフェイス 3/7 および 3/11 は、VPN に接続します。

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

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

Switch(config)# interface FastEthernet3/5
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# no ip address
Switch(config-if)# exit

Switch(config)# interface FastEthernet3/8
Switch(config-if)# switchport access vlan 208
Switch(config-if)# no ip address
Switch(config-if)# exit

Switch(config)# interface FastEthernet3/11
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# no ip address
Switch(config-if)# exit
```

スイッチ S8 上で使用される 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 および 208 は、それぞれスイッチ S11 およびスイッチ S20 を含む VPN の VRF に使用されます。

```
Switch(config)# interface Vlan10
Switch(config-if)# vrf forwarding v11
Switch(config-if)# ip address 38.0.0.8 255.255.255.0
Switch(config-if)# exit

Switch(config)# interface Vlan20
Switch(config-if)# vrf forwarding v12
Switch(config-if)# ip address 83.0.0.8 255.255.255.0
Switch(config-if)# exit

Switch(config)# interface Vlan118
Switch(config-if)# vrf forwarding v12
Switch(config-if)# ip address 118.0.0.8 255.255.255.0
Switch(config-if)# exit
```

```
Switch(config)# interface Vlan208
Switch(config-if)# vrf forwarding v11
Switch(config-if)# ip address 208.0.0.8 255.255.255.0
Switch(config-if)# exit
```

VPN1 および VPN2 に OSPF ルーティングを設定します。

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

CE から PE のルーティングに BGP を設定します。

```
Switch(config)# router bgp 800
Switch(config-router)# address-family ipv4 vrf vl2
Switch(config-router-af)# redistribute ospf 2 match internal
Switch(config-router-af)# neighbor 83.0.0.3 remote-as 100
Switch(config-router-af)# neighbor 83.0.0.3 activate
Switch(config-router-af)# network 8.8.2.0 mask 255.255.255.0
Switch(config-router-af)# exit

Switch(config-router)# address-family ipv4 vrf vl1
Switch(config-router-af)# redistribute ospf 1 match internal
Switch(config-router-af)# neighbor 38.0.0.3 remote-as 100
Switch(config-router-af)# neighbor 38.0.0.3 activate
Switch(config-router-af)# network 8.8.1.0 mask 255.255.255.0
Switch(config-router-af)# end
```

スイッチ S20 の設定

CE に接続するように S20 を設定します。

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# ip routing
Switch(config)# interface Fast Ethernet 0/7
Switch(config-if)# no switchport
Switch(config-if)# ip address 208.0.0.20 255.255.255.0
Switch(config-if)# exit

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

スイッチ S11 の設定

CE に接続するように S11 を設定します。

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# ip routing
Switch(config)# interface Gigabit Ethernet 0/3
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# no ip address
Switch(config-if)# exit

Switch(config)# interface Vlan118
```

```
Switch(config-if)# ip address 118.0.0.11 255.255.255.0
Switch(config-if)# exit

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

PE スイッチ S3 の設定

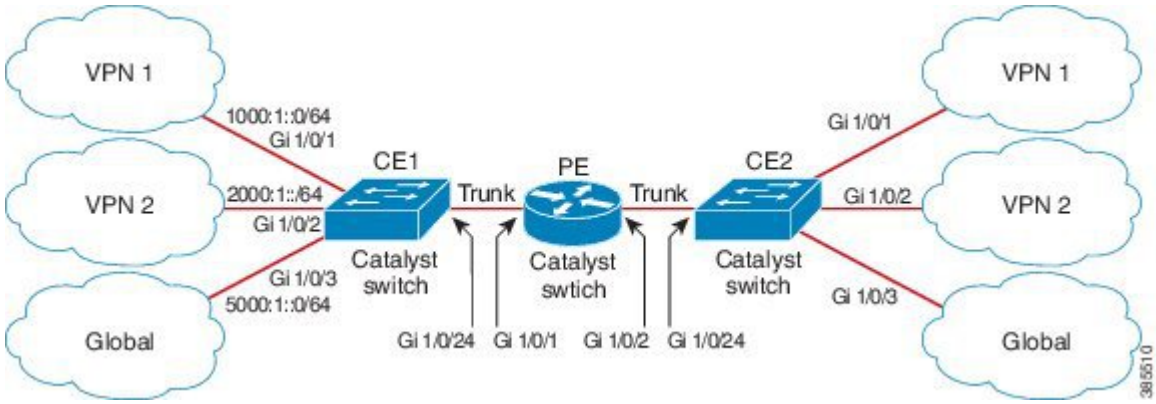
スイッチ S3（ルータ）上では、次のコマンドはスイッチ S8 への接続だけを設定します。

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

Router(config)# ip vrf v2
Router(config-vrf)# rd 100:2
Router(config-vrf)# route-target export 100:2
Router(config-vrf)# route-target import 100:2
Router(config-vrf)# exit

Router(config)# ip cef
Router(config)# interface Loopback1
Router(config-if)# vrf forwarding v1
Router(config-if)# ip address 3.3.1.3 255.255.255.0
Router(config-if)# exit

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

Router(config)# interface Fast Ethernet3/0.10
Router(config-if)# encapsulation dot1q 10
Router(config-if)# vrf forwarding v1
Router(config-if)# ip address 38.0.0.3 255.255.255.0
Router(config-if)# exit

Router(config)# interface Fast Ethernet3/0.20
Router(config-if)# encapsulation dot1q 20
Router(config-if)# vrf forwarding v2
Router(config-if)# ip address 83.0.0.3 255.255.255.0
Router(config-if)# exit

Router(config)# router bgp 100
Router(config-router)# address-family ipv4 vrf v2
Router(config-router-af)# neighbor 83.0.0.8 remote-as 800
Router(config-router-af)# neighbor 83.0.0.8 activate
Router(config-router-af)# network 3.3.2.0 mask 255.255.255.0
Router(config-router-af)# exit
Router(config-router)# address-family ipv4 vrf v1
Router(config-router-af)# neighbor 83.0.0.8 remote-as 800
Router(config-router-af)# neighbor 83.0.0.8 activate
Router(config-router-af)# network 3.3.1.0 mask 255.255.255.0
Router(config-router-af)# end
```

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 *4 : 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

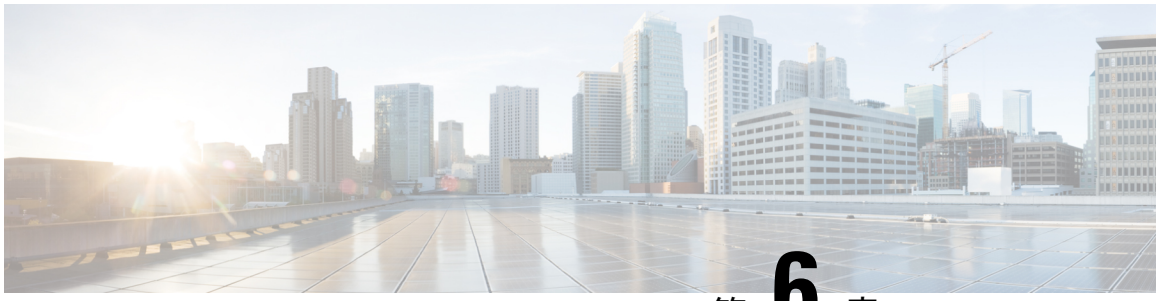
```
ipv6 unicast-routing

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
!
```



第 6 章

BGP の設定

BGP に関する情報

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

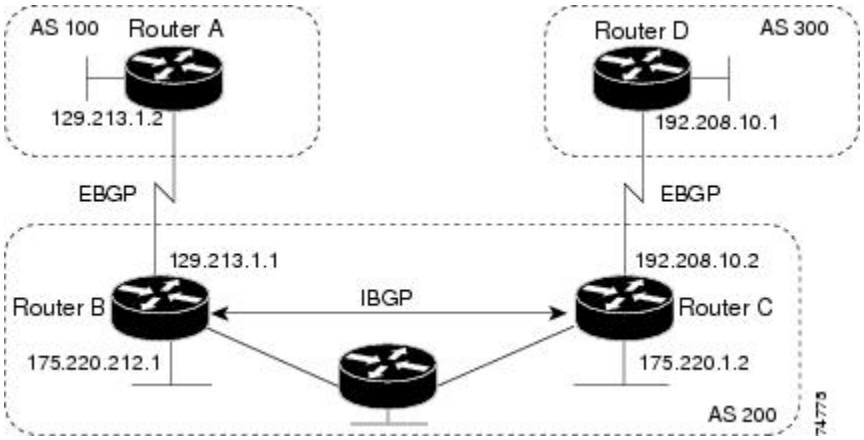
（注） この製品のマニュアルセットは、偏向のない言語を使用するように配慮されています。このドキュメントセットでの偏向のない言語とは、年齢、障害、性別、人種的アイデンティティ、民族的アイデンティティ、性的指向、社会経済的地位、およびインターセクショナリティに基づく差別を意味しない言語として定義されています。製品ソフトウェアのユーザインターフェイスにハードコードされている言語、RFP のドキュメントに基づいて使用されている言語、または参照されているサードパーティ製品で使用されている言語によりドキュメントに例外が存在する場合があります。

BGP ネットワーク トポロジ

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

されるか（IBGP）という点で異なります。下の図に、EBGP と IBGP の両方を実行しているネットワークを示します。

図 *5: EBGP*、*IBGP*、および複数の自律システム



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

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

このネットワークの特徴は次のとおりです。

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

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

BGP の場合、各ルートはネットワーク番号、情報が通過した自律システムのリスト（自律システム パス）、および他のパス属性リストで構成されます。BGP システムの主な機能は、AS パ

スのリストに関する情報など、ネットワークの到達可能性情報を他の BGP システムと交換することです。この情報は、AS が接続されているかどうかを判別したり、ルーティング ループをプルーニングしたり、AS レベル ポリシー判断を行うために使用できます。

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

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

NSF 認識

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

BGP ルーティングに関する情報

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

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

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

AS が別の AS からさらに別の AS にトラフィックを渡す場合は、アドバタイズ対象のルートに矛盾が存在しないことが重要です。BGP がルートをアドバタイズしてから、ネットワーク内のすべてのルータが IGP を通してルートを学習した場合、AS は一部のルータがルーティングできなかったトラフィックを受信することがあります。このような事態を避けるため、BGP は IGP が AS に情報を伝播し、BGP が IGP と同期化されるまで、待機する必要があります。同期化は、デフォルトで有効に設定されています。AS が特定の AS から別の AS にトラフィックを

渡さない場合、または自律システム内のすべてのルータで BGP が稼働している場合は、同期化を無効にし、IGP内で伝送されるルート数を少なくして、BGPがより短時間で収束するようにします。

ルーティング ポリシーの変更

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

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

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

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

下の表に、ハード リセットとソフト リセットの利点および欠点を示します。

表 *6:* ハード リセットとソフト リセットの利点および欠点

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

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

BGP 判断属性

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

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

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

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

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

ルート マップ

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

BGP フィルタリング

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

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

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

BGP フィルタリングのプレフィックス リスト

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

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

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

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

BGP コミュニティ フィルタリング

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

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

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

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

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

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

BGP ネイバーおよびピア グループ

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

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

集約ルート

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

ルーティング ドメイン コンフェデレーション

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

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

BGP ルート リフレクタ

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

ルートリフレクタを使用すると、学習されたルートをネイバーに渡す場合に他の方法が使用されるため、すべての IBGP スピーカーを完全メッシュ構造にする必要はありません。IBGP ピアをルート リフレクタに設定すると、その IBGP ピアは IBGP によって学習されたルートを一連の IBGP ネイバーに送信するようになります。ルート リフレクタの内部ピアには、クライアント ピアと非クライアント ピア（AS 内の他のすべてのルータ）の 2 つのグループがあります。ルート リフレクタは、これらの 2 つのグループ間でルートを反映させます。ルート リフレクタおよびクライアント ピアは、クラスタを形成します。非クライアント ピアは相互に完全メッシュ構造にする必要がありますが、クライアントピアはその必要はありません。クラスタ内のクライアントは、そのクラスタ外の IBGP スピーカーと通信しません。

アドバタイズされたルートを受信したルートリフレクタは、ネイバーに応じて、次のいずれかのアクションを実行します。

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

通常、クライアントのクラスタにはルート リフレクタが 1 つあり、クラスタはルート リフレクタのルータ ID で識別されます。冗長性を高めて、シングル ポイントでの障害を回避するには、クラスタに複数のルートリフレクタを設定する必要があります。このように設定した場合は、ルート リフレクタが同じクラスタ内のルート リフレクタからのアップデートを認識できるように、クラスタ内のすべてのルート リフレクタに同じクラスタ ID（4 バイト）を設定する必要があります。クラスタを処理するすべてのルートリフレクタは完全メッシュ構造にし、一連の同一なクライアント ピアおよび非クライアント ピアを設定する必要があります。

ルート ダンプニング

ルート フラップ ダンプニングは、インターネットワーク内でフラッピング ルートの伝播を最小化するための BGP 機能です。ルートの状態が使用可能、使用不可能、使用可能、使用不可能という具合に、繰り返し変化する場合、ルートはフラッピングと見なされます。ルートダンプニングが有効の場合は、フラッピングしているルートにペナルティ値が割り当てられます。

ルートの累積ペナルティが、設定された制限値に到達すると、ルートが稼働している場合であっても、BGPはルートのアドバタイズメントを抑制します。再使用限度は、ペナルティと比較される設定可能な値です。ペナルティが再使用限度より小さくなると、起動中の抑制されたルートのアドバタイズメントが再開されます。

IBGPによって取得されたルートには、ダンプニングが適用されません。このポリシーにより、IBGP ピアのペナルティが AS 外部のルートよりも大きくなることはありません。

条件付き BGP ルートの注入

BGP を通じてアドバタイズされるルートは、通常、使用されるルートの数が最小化され、グローバル ルーティング テーブルのサイズが小さくなるように集約されます。しかし、共通のルート集約では、より具体的なルーティング情報（より正確であるが、パケットを宛先に転送するために必要なわけではない）がわかりにくくなってしまいます。ルーティングの精度は、共通のルート集約により低下します。これは、トポロジ的に大きな領域に広がる複数のアドレスやホストを表すプレフィックスを 1 つのルートに正確に反映させることはできないからです。シスコ ソフトウェアには、プレフィックスを BGP 由来とする方法がいくつか用意されています。BGP 条件付きルート注入機能の導入以前は、既存の方法として、再配布や**network** または **aggregate-address** コマンドが使用されていました。ただし、これらの方法は、より具体的なルーティング情報（開始されるルートと一致するもの）がルーティング テーブルまたは BGP テーブルのいずれかに存在することを前提にしています。

BGP の条件付きルートの注入により、一致するものがなくても、プレフィックスを BGP ルーティング テーブルにすることができます。この機能を使って、管理ポリシーやトラフィック エンジニアリング情報に基づいて、より具体的なルートを生成することができます。これにより、設定された条件が満たされた場合にだけ BGP ルーティング テーブルに注入される、より具体的なルートへのパケットの転送をさらに厳密に制御できるようになります。この機能を有効にすると、条件に応じて、あまり具体的ではないプレフィックスにより具体的なプレフィックスを注入または置き換えることにより、共通のルート集約の精度を高めることができるようになります。元のプレフィックスと同じ、またはより具体的なプレフィックスだけが注入されます。BGP 条件付きルート注入を有効にするには、**bgp inject-map exist-map** コマンドを使用します。また、BGP 条件付きルート注入では、2 つのルート マップ（注入マップと存在マップ）を使用して、1つ（または複数）のより具体的なプレフィックスがBGPルーティングテーブルに注入されます。存在マップは、BGP スピーカーが追跡するプレフィックスを指定します。注入マップは、ローカル BGP テーブルで作成され、このテーブルにインストールされるプレフィックスを定義します。

（注） 注入マップおよび存在マップで一致となるプレフィックスはルートマップ句ごとに1つだけです。さらにプレフィックスを注入するには、ルート マップ句を追加で設定する必要があります。複数のプレフィックスが使用されている場合は、一致する最初のプレフィックスが使用されます。

BGP Peer テンプレート

構成管理など、ピア グループの制約の一部に対応するため、BGP アップデート グループ コンフィギュレーションをサポートする BGP ピア テンプレートが導入されました。

ピア テンプレートは、ポリシーを共有するネイバーに適用可能なコンフィギュレーション パターンです。ピア テンプレートは再利用が可能で、継承がサポートされているため、ネットワーク オペレータはピア テンプレートを使用して、ポリシーを共有している BGP ネイバーに対して異なるネイバー コンフィギュレーションをグループ化し適用できます。また、ネットワーク オペレータは、別のピア テンプレートからコンフィギュレーションを継承できるというピア テンプレートの機能を使用して、非常に複雑なコンフィギュレーション パターンを定義できるようになります。

ピア テンプレートには 2 種類あります。

- ピア セッション テンプレート。アドレス ファミリ モードおよび NLRI コンフィギュレーション モードすべてに共通する一般的なセッション コマンドのコンフィギュレーションをグループ化し、適用するために使用されます。
- ピア ポリシー テンプレート。特定のアドレス ファミリおよびNLRI コンフィギュレーション モードで適用されるコマンドのコンフィギュレーションをグループ化し、適用するために使用されます。

ピア テンプレートにより、柔軟性が高まり、ネイバー コンフィギュレーションの機能が強化されます。また、ピア テンプレートはピア グループ コンフィギュレーションに代わるものを提供し、ピア グループの制約の一部を解決します。ピアテンプレートを使用した BGP ピアデバイスも、自動アップデート グループ コンフィギュレーションの恩恵を受けています。BGP ピア テンプレートが設定され、BGP ダイナミック アップデート ピア グループがサポートされたことにより、ネットワーク オペレータは BGP でピア グループを設定する必要がなくなります。また、ネットワークはコンフィギュレーションの柔軟性が高まり、コンバージェンスが高速化されたことによる恩恵を受けます。

（注） BGP ネイバーを、ピア グループとピア テンプレートの両方と連動するようには設定できません。BGP ネイバーは、1 つのピア グループだけに属するように設定するか、またはピア テンプレートからポリシーを継承するように設定します。

ピア ポリシー テンプレートには、次の制約事項が適用されます。

- ピア ポリシー テンプレートは、直接的、または間接的に、最高 8 個のピア ポリシー テンプレートを継承できます。
- BGP ネイバーを、ピア グループとピア テンプレートの両方と連動するようには設定できません。BGP ネイバーは、1 つのピア グループだけに属するように設定するか、またはピア テンプレートだけからポリシーを継承するように設定できます。

ピア テンプレートでの継承

継承機能は、ピア テンプレート操作の重要なコンポーネントです。ピア テンプレートでの継承は、たとえば、ファイルとディレクトリ ツリーなど、一般的なコンピューティングで見られるノードとツリーの構造に似ています。ピア テンプレートは、別のピア テンプレートから直接、または間接的にコンフィギュレーションを継承することができます。直接継承されたピアテンプレートは、構造体のツリーを表します。間接継承されたピア テンプレートはツリーのノードを表します。個々のノードもまた継承をサポートしているため、ブランチを作成して、そこから直接継承されたピアテンプレートすなわちツリーの起点へ連なる全ての間接継承されたピアテンプレートの設定を適用することができます。

この構造により、ネイバーのグループに通常、再適用されるコンフィギュレーション文を繰り返す必要がなくなります。これは、共通のコンフィギュレーション文を一度適用しておくと、その後は共通のコンフィギュレーションを持つネイバー グループに適用されるピア グループにより間接継承されるからです。ノードとツリー内部の別々の箇所で重複するコンフィギュレーション文は、ツリーの起点で直接継承したテンプレートによりフィルタ処理されます。直接継承されたテンプレートは、重複する間接継承された文を直接継承された文で上書きします。

継承によりネイバー コンフィギュレーションのスケーラビリティと柔軟性がさらに広がり、複数のピアテンプレート コンフィギュレーションを連ねることで、共通のコンフィギュレーション文を継承する単純なコンフィギュレーションを作成したり、共通に継承されるコンフィギュレーションとともに非常に限定的なコンフィギュレーション文を適用する複雑なコンフィギュレーションを作成したりできるようになります。ピア セッション テンプレートおよびピア ポリシーテンプレートでの継承の設定についての詳細は、これ以降のセクションで説明します。

BGP ネイバーが継承したピア テンプレートを使用する場合、特定のテンプレートに関連付けられているポリシーを判断するのが難しいことがあります。**show ip bgptemplate peer-policy** コマンドに、特定のテンプレートに関連付けられているローカルポリシーおよび継承されたポリシーの詳しいコンフィギュレーションを表示するためのキーワード **detail** が追加されました。

ピア セッション テンプレート

ピア セッション テンプレートは、一般的なセッション コマンドのコンフィギュレーションをグループ化し、セッション コンフィギュレーション要素を共有するネイバーのグループに適用するために使用されます。異なるアドレス ファミリで設定されているネイバーに共通する一般的なセッション コマンドは、同じピア セッション テンプレートに設定できます。ピア セッション テンプレートの作成と設定は、ピア セッション コンフィギュレーション モードで行います。ピア セッション テンプレートで設定できるのは、一般的なセッション コマンドだけです。次の一般的なセッション コマンドは、ピア セッション テンプレートでサポートされています。

- **description**
- **disable-connected-check**
- **ebgp-multihop**
- **exit peer-session**

- **inherit peer-session**
- **local-as**
- **password**
- **remote-as**
- **shutdown**
- **timers**
- **translate-update**
- **update-source**
- **version**

一般的なセッション コマンドをピア セッションで一度設定しておくと、ピア セッション テンプレートの直接適用、またはピア セッション テンプレートの間接継承によって、多数のネイバーに適用できます。ピア セッション テンプレートのコンフィギュレーションにより、自律システム内のすべてのネイバーに共通に適用される一般的なセッション コマンドのコンフィギュレーションが簡素化されます。

ピア セッション テンプレートは、直接継承と間接継承をサポートします。一度にピアの設定に使用できるピア セッション テンプレートは 1 つだけです。また、このピア セッション テンプレートは、間接継承されたピア セッション テンプレートを 1 つだけ含むことができます。

（注） 1 つのピア セッション テンプレートを使って、複数の継承文を設定しようとすると、エラーメッセージが表示されます。

この動作により、BGP ネイバーは 1 つのセッション テンプレートだけを直接継承し、最高 7 個のピア セッション テンプレートを間接継承できます。したがって、1 つのネイバーに最高 8 個のピア セッション コンフィギュレーション（直接継承されたピア セッション テンプレートのコンフィギュレーションと最高 7 個の間接継承されたピア セッション テンプレートのコンフィギュレーション）を適用できます。継承されたピア セッション コンフィギュレーションは、ブランチの最後のノードが最初に評価されて適用され、ツリーの起点で直接適用されたピア セッション テンプレートが最後に適用されます。直接適用されたピア セッション テンプレートは、継承されたピア セッション テンプレート コンフィギュレーションよりも優先されます。継承されたピア セッション テンプレートで重複するコンフィギュレーション文はすべて、直接適用されたピア セッション テンプレートにより上書きされます。したがって、基本セッション コマンドが異なる値で再び適用される場合は、後の値が優先され、間接継承されたテンプレートに設定されていた前の値は上書きされます。次に、この機能を使用した例を示します。

次の例では、一般セッション コマンド **remote-as 1** がピア セッション テンプレート SESSION-TEMPLATE-ONE に適用されます。

```
template peer-session SESSION-TEMPLATE-ONE
```

```
remote-as 1
exit peer-session
```

ピア セッション テンプレートは、一般的なセッション コマンドだけをサポートします。特定のアドレス ファミリ、または NLRI コンフィギュレーション モードだけのために設定される BGP ポリシー コンフィギュレーション コマンドは、ピア ポリシー テンプレートで設定されます。

ピア ポリシー テンプレート

ピア ポリシー テンプレートは、特定のアドレス ファミリおよび NLRI コンフィギュレーション モードで適用されるコマンドのコンフィギュレーションをグループ化し、適用するために使用されます。ピア ポリシー テンプレートの作成と設定は、ピア ポリシー コンフィギュレーション モードで行います。特定のアドレス ファミリ専用に設定される BGP ポリシー コマンドは、ピア ポリシー テンプレートで設定されます。ピア ポリシー テンプレートでは、次の BGP ポリシー コマンドがサポートされています。

- **advertisement-interval**
- **allowas-in**
- **as-override**
- **capability**
- **default-originate**
- **distribute-list**
- **dmzlink-bw**
- **exit-peer-policy**
- **filter-list**
- **inherit peer-policy**
- **maximum-prefix**
- **next-hop-self**
- **next-hop-unchanged**
- **prefix-list**
- **remove-private-as**
- **route-map**
- **route-reflector-client**
- **send-community**
- **send-label**
- **soft-reconfiguration**
- **unsuppress-map**

- **weight**

ピア ポリシー テンプレートは、特定のアドレス ファミリに属するネイバーに設定される BGP ポリシー コマンドの設定に使用されます。ピア セッション テンプレートと同様、ピア ポリシーテンプレートを一度設定しておくと、直接適用、または継承を通じて、多数のネイバーにピア ポリシーテンプレートを適用することができます。ピア ポリシー テンプレートの設定により、自律システム内のすべてのネイバーに適用される BGP ポリシー コマンドの設定が簡略化されます。

ピア セッション テンプレートと同様、ピア ポリシー テンプレートは継承をサポートしています。しかし、多少の違いはあります。直接適用されたピア ポリシー テンプレートは、最大 7 つのピア ポリシー テンプレートから設定を直接的または間接継承できます。したがって、合計 8 つのピア ポリシーテンプレートをネイバーまたはネイバー グループに適用できます。ルート マップと同じように、継承されたピア ポリシー テンプレートにはシーケンス番号が設定されます。また、ルート マップと同じように、継承されたピア ポリシー テンプレートは、最も低いシーケンス番号を持つ **inherit peer-policy** 文が最初に評価され、最も高いシーケンス番号のものが最後に評価されます。ただし、ピア ポリシー テンプレートはルート マップのように折りたたむことはできません。シーケンスはすべて評価されます。異なる値を使って、BGP ポリシー コマンドが再適用された場合は、シーケンス番号の小さいものから順に、前の値がすべて上書きされます。

直接適用されたピア ポリシーテンプレートと、シーケンス番号が最も大きい **inherit peer-policy** 文のプライオリティは常に最も高く、最後に適用されます。これ以降のピア テンプレートに再適用されるコマンドは、必ず、前の値を上書きします。この動作は、個々のポリシー コンフィギュレーション コマンドを繰り返さずとも、共通のポリシー コンフィギュレーションは大規模なネイバー グループに適用し、特定のポリシー コンフィギュレーションは特定のネイバーやネイバー グループだけに適用できるように設計されています。

ピア ポリシー テンプレートは、ポリシー コンフィギュレーション コマンドだけをサポートします。特定のアドレス ファミリ用に設定される BGP ポリシー コンフィギュレーション コマンドは、ピア ポリシー テンプレートで設定されます。

ピア ポリシー テンプレートの設定により、BGP 設定が簡略化され、柔軟性が向上します。特定のポリシーを 1 回設定すれば、何回も参照できます。ピア ポリシーは最大 8 レベルの継承をサポートするため、非常に具体的で複雑な BGP ポリシーも作成できます。

BGP ルート マップ ネクスト ホップ セルフ

BGP ルート マップ ネクスト ホップ セルフ機能は、bgp next-hop unchanged と bgp next-hop unchanged allpaths の設定を選択的にオーバーライドする方法を提供します。これらの設定はアドレス ファミリに対してグローバルに適用されます。ルートによっては、これは適切でない場合があります。たとえば、スタティック ルートは、自身をネクスト ホップとして再配布する必要がある一方で、接続ルート、および内部ボーダー ゲートウェイ プロトコル（IBGP）または外部ボーダー ゲートウェイ プロトコル（EBGP）を介して学習されたルートは、引き続きネクスト ホップを変更せずに再配布する場合があります。

BGP ルート マップ ネクスト ホップ セルフ機能は、bgp next-hop unchanged 設定と bgp next-hop unchanged allpaths 設定をオーバーライドする新しい ip next-hop self 設定を構成できるように、既存のルート マップ インフラストラクチャを変更します。

ip next-hop self 設定は、VPNv4 および VPNv6 アドレス ファミリにのみ適用されます。BGP 以外のプロトコルによって配布されるルートは影響を受けません。

新しい bgp route-map priority 設定を使用すると、bgp next-hop unchanged と bgp next-hop unchanged allpaths の設定よりもルート マップが優先されることを BGP に通知できます。bgp route-map priority 設定は、BGP にのみ影響します。bgp next-hop unchanged または bgp next-hop unchanged allpaths 設定を構成していない場合、bgp route-map priority 設定は効果がありません。

BGP の設定方法

BGP のデフォルト設定

下の表に、BGP のデフォルト設定を示します。

表 *7: BGP* のデフォルト設定

機能	デフォルト設定
集約アドレス	無効：未定義
AS パス アクセス リスト	未定義
自動サマリー	無効
最適パス	• ルータはルートを選択する場合に *as-path* を考慮し、外部 BGP ピアからの類似ルートは比較しません。 • ルータ ID の比較：無効
BGP コミュニティ リスト	• 番号：未定義。コミュニティ番号を示す特定の値を許可すると、許可されていないその他すべてのコミュニティ番号は、暗黙の拒否にデフォルト設定されます。 • フォーマット：シスコデフォルト フォーマット（32 ビット番号）
BGP 連合 ID/ピア	• ID：未設定 • ピア：識別なし
BGP 高速外部フォールオーバー	有効

機能	デフォルト設定
BGP ローカル初期設定	100。指定できる範囲は 0～4294967295 です（大きな値を推奨）。
BGP ネットワーク	指定なし。バックドア ルートのアドバタイズなし
BGP ルート ダンプニング	デフォルトでは、無効です。有効の場合は、次のようになります。 • 半減期は 15 分 • 再使用は 750（10 秒増分） • 抑制は 2000（10 秒増分） • 最大抑制時間は半減期の 4 倍（60 分）
BGP ルータ ID	ループバック インターフェイスに IP アドレスが設定されている場合は、ループバック インターフェイスの IP アドレス、またはルータの物理インターフェイスに対して設定された最大の IP アドレス
デフォルトの情報送信元（プロトコルまたはネットワーク再配布）	無効
デフォルト メトリック	自動メトリック変換（組み込み）
ディスタンス	• 外部ルート アドミニストレーティブ ディスタンス：20（有効値は 1 ～ 255） • 内部ルート アドミニストレーティブ ディスタンス：200（有効値は 1 ～ 255） • ローカル ルート アドミニストレーティブ ディスタンス：200（有効値は 1 ～ 255）
ディストリビュート リスト	• 入力（アップデート中に受信されたネットワークをフィルタリング）：無効 • 出力（アップデート中のネットワークのアドバタイズを抑制）：無効
内部ルート再配布	無効
IP プレフィックス リスト	未定義

機能	デフォルト設定
Multi Exit Discriminator（MED）	• 常に比較：無効。異なる自律システム内のネイバーからのパスに対して、MED を比較しません。 • 最適パスの比較：無効 • 最悪パスである MED の除外：無効 • 決定的な MED 比較：無効

機能	デフォルト設定
ネイバー	

機能	デフォルト設定
	• アドバタイズメント インターバル：外部ピアの場合は 30 秒、内部ピアの場合は 5 秒 • ロギング変更：有効 • 条件付きアドバタイズ：無効 • デフォルト送信元：ネイバーに送信されるデフォルト ルートはなし • 説明：なし • ディストリビュート リスト：未定義 • 外部 BGP マルチホップ：直接接続されたネイバーだけを許可 • フィルタ リスト：使用しない • 受信したプレフィックスの最大数：制限なし • ネクストホップ（BGP ネイバーのネクスト ホップとなるルータ）：無効 • パスワード：無効 • ピア グループ：定義なし、割り当てメンバーなし • プレフィックス リスト：指定なし • リモート AS（ネイバー BGP テーブルへのエントリ追加）：ピア定義なし • プライベート AS 番号の削除：無効 • ルート マップ：ピアへの適用なし • コミュニティ属性送信：ネイバーへの送信なし。 • シャットダウンまたはソフト再設定：無効 • タイマー：60 秒、ホールドタイム：180 秒 • アップデート送信元：最適ローカル アドレス

機能	デフォルト設定
	• バージョン：BGP バージョン 4 • 重み：BGP ピアによって学習されたルート：0、ローカル ルータから取得されたルート：32768
NSF[1] 認識	無効にされた NSF 認識は、グレースフルリスタートを有効にすることにより、 ライセンスを実行するスイッチ上で IPv4 に対して有効にできます。[2]。有効な場合、レイヤ 3 スイッチでは、ハードウェアやソフトウェアの変更中に、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。
ルート リフレクタ	未設定
同期化（BGP および IGP）	無効
テーブル マップ アップデート	無効
タイマー	キープアライブ：60秒、ホールドタイム：180秒

1 Nonstop Forwarding
2

BGP ルーティングの有効化

始める前に

（注） BGP を有効にするには、スイッチまたはスタックマスター上で IP サービス フィーチャ セットが稼働している

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`	IP ルーティングを有効にします。
ステップ 4	**router bgp** *autonomous-system* 例： `Device(config)# router bgp 45000`	BGP ルーティング プロセスを有効にして AS 番号を割り当て、ルータ コンフィギュレーション モードを開始します。指定できる AS 番号は 1～65535 です。64512～65535 は、プライベート AS 番号専用です。
ステップ 5	**network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*] 例： `Device(config)# network 10.108.0.0`	このASに対してローカルとなるようにネットワークを設定し、BGP テーブルにネットワークを格納します。
ステップ 6	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *number* 例： `Device(config)# neighbor 10.108.1.2 remote-as 65200`	BGP ネイバー テーブルに設定を追加し、IP アドレスによって識別されるネイバーが、指定された AS に属することを示します。 EBGP の場合、通常ネイバーは直接接続されており、IP アドレスは接続のもう一方の端におけるインターフェイスのアドレスです。 IBGPの場合、IPアドレスにはルータ インターフェイス内の任意のアドレスを指定できます。
ステップ 7	**neighbor** {*ip-address* | *peer-group-name*} **remove-private-as** 例： `Device(config)# neighbor 172.16.2.33 remove-private-as`	（任意）発信ルーティング アップデート内の AS パスからプライベート AS 番号を削除します。
ステップ 8	**synchronization** 例： `Device(config)# synchronization`	（任意）BGP と IGP の同期化を有効にします。

	コマンドまたはアクション	目的
ステップ 9	**auto-summary** 例： `Device(config)# auto-summary`	（任意）自動ネットワーク サマライズを有効にします。IGP から BGP にサブネットが再配布された場合、ネットワーク ルートだけが BGP テーブルに挿入されます。
ステップ 10	**bgp graceful-restart** 例： `Device(config)# bgp graceful-start`	（任意）NSF 認識をスイッチで有効にします。NSF 認識はデフォルトでは無効です。
ステップ 11	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 12	**show ip bgp network** *network-number* 例： `Device# show ip bgp network 10.108.0.0`	設定を確認します。
ステップ 13	**show ip bgp neighbor** 例： `Device# show ip bgp neighbor`	NSF 認識（グレースフル リスタート）がネイバーで有効にされていることを確認します。スイッチおよびネイバーでNSF認識が有効になっている場合、次のメッセージが表示されます。Graceful Restart Capability: advertised and received スイッチでNSF認識が有効になっていて、ネイバーで有効になっていない場合、次のメッセージが表示されます。Graceful Restart Capability: advertised
ステップ 14	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ルーティング ポリシー変更の管理

BGP ピアがルート リフレッシュ機能をサポートするかどうかを学習して、BGP セッションをリセットするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**show ip bgp neighbors** 例： `Device# show ip bgp neighbors`	ネイバーがルートリフレッシュ機能をサポートするかどうかを表示します。サポートされている場合は、ルータに関する次のメッセージが表示されます。 *Received route refresh capability from peer*
ステップ 2	**clear ip bgp** {***** | *address* | *peer-group-name*} 例： `Device# clear ip bgp *`	指定された接続上でルーティングテーブルをリセットします。 • すべての接続をリセットする場合は、アスタリスク（*）を入力します。 • 特定の接続をリセットする場合は、IPアドレスを入力します。 • ピア グループをリセットする場合は、ピア グループ名を入力します。
ステップ 3	**clear ip bgp** {***** | *address* | *peer-group-name*} **soft out** 例： `Device# clear ip bgp * soft out`	（任意）指定された接続上でインバウンドルーティングテーブルをリセットするには、アウトバウンドソフト リセットを実行します。このコマンドは、ルートリフレッシュがサポートされている場合に使用してください。 • すべての接続をリセットする場合は、アスタリスク（*）を入力します。 • 特定の接続をリセットする場合は、IPアドレスを入力します。 • ピア グループをリセットする場合は、ピア グループ名を入力します。
ステップ 4	**show ip bgp** 例： `Device# show ip bgp`	ルーティング テーブルおよびBGP ネイバーに関する情報をチェックし、リセットされたことを確認します。
ステップ 5	**show ip bgp neighbors** 例： `Device# show ip bgp neighbors`	ルーティング テーブルおよびBGP ネイバーに関する情報をチェックし、リセットされたことを確認します。

BGP 判断属性の設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router bgp** *autonomous-system* 例： `Device(config)# router bgp 4500`	BGP ルーティング プロセスを有効にして AS 番号を割り当て、ルータ コンフィギュレーション モードを開始します。
ステップ 4	**bgp best-path as-path ignore** 例： `Device(config-router)# bgp bestpath as-path ignore`	（任意）ルート選択中にASパス長を無視するようにルータを設定します。
ステップ 5	**neighbor** {*ip-address* | *peer-group-name*} **next-hop-self** 例： `Device(config-router)# neighbor 10.108.1.1 next-hop-self`	（任意）ネクストホップ アドレスの代わりに使用される特定の IP アドレスを入力し、ネイバーへの BGP アップデートに関するネクストホップの処理を無効にします。
ステップ 6	**neighbor** {*ip-address* | *peer-group-name*} **weight** *weight* 例： `Device(config-router)# neighbor 172.16.12.1 weight 50`	（任意）ネイバー接続に重みを割り当てます。指定できる値は0～65535です。最大の重みのルートを推奨します。別の BGP ピアから学習されたルートのデフォルトの重みは 0 です。ローカル ルータから送信されたルートのデフォルトの重みは32768です。
ステップ 7	**default-metric** *number* 例： `Device(config-router)# default-metric 300`	（任意）推奨パスを外部ネイバーに設定するようにMEDメトリックを設定します。MEDを持たないすべてのルータも、この値に設定されます。指定できる範囲は 1 ～ 4294967295 です。最小値を推奨します。

	コマンドまたはアクション	目的
ステップ 8	**bgp bestpath med missing-as-worst** 例： `Device(config-router)# bgp bestpath med missing-as-worst`	（任意）MED がない場合は無限の値が指定されていると見なし、MED 値を持たないパスが最も望ましくないパスになるように、スイッチを設定します。
ステップ 9	**bgp always-compare med** 例： `Device(config-router)# bgp always-compare-med`	（任意）異なるAS内のネイバーからのパスに対して、MEDを比較するようにスイッチを設定します。デフォルトでは、MED は同じ AS 内のパス間でだけ比較されます。
ステップ 10	**bgp bestpath med confed** 例： `Device(config-router)# bgp bestpath med confed`	（任意）連合内の異なるサブASによってアドバタイズされたパスから特定のパスを選択する場合に、MED を考慮するようにスイッチを設定します。
ステップ 11	**bgp deterministic med** 例： `Device(config-router)# bgp deterministic med`	（任意）同じAS内の異なるピアによってアドバタイズされたルートから選択する場合に、MED 変数を考慮するようにスイッチを設定します。
ステップ 12	**bgp default local-preference** *value* 例： `Device(config-router)# bgp default local-preference 200`	（任意）デフォルトのローカル プリファレンス値を変更します。指定できる範囲は 0 ～ 4294967295 で、デフォルト値は 100 です。最大のローカル プリファレンス値を推奨します。
ステップ 13	**maximum-paths** *number* 例： `Device(config-router)# maximum-paths 8`	（任意）IP ルーティング テーブルに追加するパスの数を設定します。デフォルトでは、最適パスだけがルーティング テーブルに追加されます。指定できる範囲は 1 ～ 16 です。複数の値を指定すると、パス間のロード バランシングが可能になります。スイッチ ソフトウェア では最大32の等コスト ルートが許可されていますが、スイッチ ハードウェア はルートあたり 17 パス以上は使用しません。
ステップ 14	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 15	**show ip bgp** 例： `Device# show ip bgp`	ルーティング テーブルおよびBGP ネイバーに関する情報をチェックし、リセットされたことを確認します。

	コマンドまたはアクション	目的
ステップ 16	**show ip bgp neighbors** 例： `Device# show ip bgp neighbors`	ルーティング テーブルおよびBGP ネイバーに関する情報をチェックし、リセットされたことを確認します。
ステップ 17	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ルート マップによる BGP フィルタリングの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] 例： `Device(config)# route-map set-peer-address permit 10`	ルートマップを作成し、ルートマップ コンフィギュレーション モードを開始します。
ステップ 4	**set ip next-hop** *ip-address* [*...ip-address*] [**peer-address**] 例： `Device(config)# set ip next-hop 10.1.1.3`	（任意）ネクストホップ処理を無効にするようにルート マップを設定します。 • インバウンド ルート マップの場合は、一致するルートのネクスト ホップをネイバー ピア アドレスに設定し、サードパーティのネクスト ホップを上書きします。 • BGP ピアのアウトバウンド ルート マップの場合は、ネクスト ホップをローカル ルータのピア アドレスに設定して、ネクスト ホップ計算を無効にします。

	コマンドまたはアクション	目的
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show route-map** [*map-name*] 例： `Device# show route-map`	設定を確認するため、設定されたすべてのルートマップ、または指定されたルートマップだけを表示します。
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ネイバーによる BGP フィルタリングの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router bgp** *autonomous-system* 例： `Device(config)# router bgp 109`	BGP ルーティング プロセスを有効にして AS 番号を割り当て、ルータ コンフィギュレーション モードを開始します。
ステップ 4	**neighbor** {*ip-address* | *peer-group name*} **distribute-list** {**access-list-number** | *name*} {**in** | **out**} 例：	（任意）アクセス リストの指定に従って、ネイバーに対して送受信される BGP ルーティング アップデートをフィルタリングします。

	コマンドまたはアクション	目的
	`Device(config-router)# neighbor 172.16.4.1 distribute-list 39 in`	（注） **neighbor prefix-list** ルータ コンフィギュレーション コマンドを使用して、アップデートをフィルタリングすることもできますが、両方のコマンドを使用して同じBGPピアを設定することはできません。
ステップ 5	**neighbor** {*ip-address* | *peer-group name*} **route-map** *map-tag* {**in** | **out**} 例： `Device(config-router)# neighbor 172.16.70.24 route-map internal-map in`	（任意）ルート マップを適用し、着信または発信ルートをフィルタリングします。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show ip bgp neighbors** 例： `Device# show ip bgp neighbors`	設定を確認します。
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

アクセス リストおよびネイバーによる BGP フィルタリングの設定

BGP 自律システム パスに基づいて着信および発信の両方のアップデートにアクセス リスト フィルタを指定して、フィルタリングすることもできます。各フィルタは、正規表現を使用するアクセス リストです。この方法を使用するには、自律システム パスのアクセス リストを定義し、特定のネイバーとの間のアップデートに適用します。

手順

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

	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 as-path access-list** *access-list-number* {**permit** | **deny**} *as-regular-expressions* 例： `Device(config)# ip as-path access-list 1 deny _65535_`	BGP-related アクセス リストを定義します。
ステップ 4	**router bgp** *autonomous-system* 例： `Device(config)# router bgp 110`	BGP ルータ コンフィギュレーション モードを開始します。
ステップ 5	**neighbor** {*ip-address* | *peer-group name*} **filter-list** {*access-list-number* | *name*} {**in** | **out** | **weight** *weight*} 例： `Device(config-router)# neighbor 172.16.1.1 filter-list 1 out`	アクセス リストに基づいて、BGP フィルタを確立します。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show ip bgp neighbors** [**paths** *regular-expression*] 例： `Device# show ip bgp neighbors`	設定を確認します。
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

BGP フィルタリング用のプレフィックス リストの設定

コンフィギュレーションエントリを削除する場合は、シーケンス番号を指定する必要はありません。**Show** コマンドの出力には、シーケンス番号が含まれます。

コマンド内でプレフィックス リストを使用する場合は、あらかじめプレフィックス リストを設定しておく必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip prefix-list** *list-name* [**seq** *seq-value*] **deny** | **permit** *network/len* [**ge** *ge-value*] [**le** *le-value*] 例： `Device(config)# ip prefix-list BLUE permit 172.16.1.0/24`	一致条件に合わせてアクセスを **deny** または **permit** するプレフィックスリストを作成します。シーケンス番号を指定することもできます。少なくとも 1 つの **permit** または **deny** 句を入力する必要があります。 • *network/len* は、ネットワーク番号およびネットワーク マスクの長さ（ビット単位）です。 •（任意）**ge** および **le** の値は、一致させるプレフィックス長を指定します。指定する *ge-value* および l*e-value* は次の条件を満たしている必要があります。*len* < *ge-value* < *le-value* < 32
ステップ 4	**ip prefix-list** *list-name* **seq** *seq-value* **deny** | **permit** *network/len* [**ge** *ge-value*] [**le** *le-value*] 例： `Device(config)# ip prefix-list BLUE seq 10 permit 172.24.1.0/24`	（任意）プレフィックス リストにエントリを追加し、そのエントリにシーケンス番号を割り当てます。
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show ip prefix list** [**detail** | **summary**] *name* [*network/len*] [**seq** *seq-num*] [**longer**] [**first-match**] 例： `Device# show ip prefix list summary test`	プレフィックス リストまたはプレフィックス リストエントリに関する情報を表示して、設定を確認します。

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

BGP コミュニティ フィルタリングの設定

デフォルトでは、COMMUNITIES 属性はネイバーに送信されません。COMMUNITIES 属性が特定の IP アドレスのネイバーに送信されるように指定するには、**neighbor send-community** ルータ コンフィギュレーション コマンドを使用します。

手順の概要

1. **enable**
2. **configure terminal**
3. **ip community-list** *community-list-number* {**permit** | **deny**} *community-number*
4. **router bgp** *autonomous-system*
5. **neighbor** {*ip-address* | *peer-group name*} **send-community**
6. **set comm-list** *list-num* **delete**
7. **exit**
8. **ip bgp-community new-format**
9. **end**
10. **show ip bgp community**
11. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip community-list** *community-list-number* {**permit**	**deny**} *community-number* 例：	コミュニティ リストを作成し、番号を割り当てます。

	コマンドまたはアクション	目的
	`Device(config)# ip community-list 1 permit 50000:10`	• *community-list-number* は 1 ～ 99 の整数です。この値は、コミュニティの1つ以上の許可または拒否グループを識別します。 • *community-number* は、**set community** ルート マップ コンフィギュレーション コマンドで設定される番号です。
ステップ 4	**router bgp** *autonomous-system* 例： `Device(config)# router bgp 108`	BGP ルータ コンフィギュレーション モードを開始します。
ステップ 5	**neighbor** {*ip-address* | *peer-group name*} **send-community** 例： `Device(config-router)# neighbor 172.16.70.23 send-community`	この IP アドレスのネイバーに送信する COMMUNITIES 属性を指定します。
ステップ 6	**set comm-list** *list-num* **delete** 例： `Device(config-router)# set comm-list 500 delete`	（任意）ルート マップで指定された標準または拡張コミュニティ リストと一致する着信または発信アップデートのコミュニティ属性から、コミュニティを削除します。
ステップ 7	**exit** 例： `Device(config-router)# end`	グローバル コンフィギュレーション モードに戻ります。
ステップ 8	**ip bgp-community new-format** 例： `Device(config)# ip bgp-community new format`	（任意）AA:NN の形式で、BGP コミュニティを表示、解析します。 BGP コミュニティは、2つの部分からなる2バイト長形式で表示されます。シスコのデフォルトのコミュニティ形式は、NNAA です。BGP に関する最新の RFC では、コミュニティは AA:NN の形式をとります。最初の部分はAS番号で、その次の部分は 2 バイトの数値です。
ステップ 9	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

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

BGP ネイバーおよびピア グループの設定

各ネイバーに設定オプションを割り当てるには、ネイバーの IP アドレスを使用し、次に示すルータ コンフィギュレーション コマンドのいずれかを指定します。ピア グループにオプションを割り当てるには、ピア グループ名を使用し、いずれかのコマンドを指定します。**neighbor shutdown** ルータ コンフィギュレーション コマンドを使用して、コンフィギュレーション情報を削除せずに、BGP ピア、またはピアグループを削除することができます。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router bgp** *autonomous-system*	BGP ルータ コンフィギュレーション モードを開始します。
ステップ 4	**neighbor** *peer-group-name* **peer-group**	BGP ピア グループを作成します。
ステップ 5	**neighbor** *ip-address* **peer-group** *peer-group-name*	BGP ネイバーをピア グループのメンバにします。
ステップ 6	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *number*	BGP ネイバーを指定します。**remote-as** *number* を使用してピアグループが設定されていない場合は、このコマンドを使用し、EBGP ネイバーを含むピアグループを作成します。指定できる範囲は 1 ～ 65535 です。

	コマンドまたはアクション	目的
ステップ 7	**neighbor** {*ip-address* | *peer-group-name*} **description** *text*	（任意）ネイバーに説明を関連付けます。
ステップ 8	**neighbor** {*ip-address* | *peer-group-name*} **default-originate** [**route-map** *map-name*]	（任意）BGP スピーカー（ローカル ルータ）にネイバーへのデフォルト ルート 0.0.0.0 の送信を許可して、このルートがデフォルト ルートとして使用されるようにします。
ステップ 9	**neighbor** {*ip-address* | *peer-group-name*} **send-community**	（任意）この IP アドレスのネイバーに送信する COMMUNITIES 属性を指定します。
ステップ 10	**neighbor** {*ip-address* | *peer-group-name*} **update-source** *interface*	（任意）内部 BGP セッションに、TCP 接続に関するすべての操作インターフェイスの使用を許可します。
ステップ 11	**neighbor** {*ip-address* | *peer-group-name*} **ebgp-multihop**	（任意）ネイバーがセグメントに直接接続されていない場合でも、BGP セッションを使用可能にします。マルチホップ ピア アドレスへの唯一のルートがデフォルトルート（0.0.0.0）の場合、マルチホップ セッションは確立されません。
ステップ 12	**neighbor** {*ip-address* | *peer-group-name*} **local-as** *number*	（任意）ローカル AS として使用する AS 番号を指定します。指定できる範囲は 1 ～ 65535 です。
ステップ 13	**neighbor** {*ip-address* | *peer-group-name*} **advertisement-interval** *seconds*	（任意）BGP ルーティング アップデートを送信する最小間隔を設定します。
ステップ 14	**neighbor** {*ip-address* | *peer-group-name*} **maximum-prefix** *maximum* [*threshold*]	（任意）ネイバーから受信できるプレフィックス数を制御します。指定できる範囲は 1 ～ 4294967295 です。*threshold*（任意）は、警告メッセージが生成される基準となる最大値（パーセンテージ）です。デフォルトは 75% です。
ステップ 15	**neighbor** {*ip-address* | *peer-group-name*} **next-hop-self**	（任意）ネイバー宛ての BGP アップデートに関して、ネクストホップでの処理を無効にします。
ステップ 16	neighbor {*ip-address* | *peer-group-name*} password *string*	（任意）TCP 接続での MD5 認証を BGP ピアに設定します。両方の BGP ピアに同じパスワードを設定する必要があります。そうしないと、BGP ピア間に接続が作成されません。
ステップ 17	**neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}	（任意）着信または発信ルートにルート マップを適用します。
ステップ 18	**neighbor** {*ip-address* | *peer-group-name*} **send-community**	（任意）この IP アドレスのネイバーに送信する COMMUNITIES 属性を指定します。

	コマンドまたはアクション	目的
ステップ 19	**neighbor** {*ip-address* | *peer-group-name*} **timers** *keepalive holdtime*	（任意）ネイバーまたはピア グループ用のタイマーを設定します。 • *keepalive* インターバルは、キープアライブ メッセージがピアに送信される間隔です。指定できる範囲は 1 ～ 4294967295 秒です。デフォルト値は 60 秒です。 • *holdtime* は、キープアライブ メッセージを受信しなかった場合、ピアが非アクティブと宣言されるまでのインターバルです。指定できる範囲は 1 ～ 4294967295 秒です。デフォルト値は 180 秒です。
ステップ 20	**neighbor** {*ip-address* | *peer-group-name*} **weight** *weight*	（任意）ネイバーからのすべてのルートに関する重みを指定します。
ステップ 21	**neighbor** {*ip-address* | *peer-group-name*} **distribute-list** {*access-list-number* | *name*} {**in** | **out**}	（任意）アクセス リストの指定に従って、ネイバーに対して送受信される BGP ルーティング アップデートをフィルタリングします。
ステップ 22	**neighbor** {*ip-address* | *peer-group-name*} **filter-list** *access-list-number* {**in** | **out** | **weight** *weight*}	（任意）BGP フィルタを確立します。
ステップ 23	**neighbor** {*ip-address* | *peer-group-name*} **version** *value*	（任意）ネイバーと通信するときに使用する BGP バージョンを指定します。
ステップ 24	**neighbor** {*ip-address* | *peer-group-name*} **soft-reconfiguration inbound**	（任意）受信したアップデートのストアを開始するようにソフトウェアを設定します。
ステップ 25	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 26	**show ip bgp neighbors**	設定を確認します。
ステップ 27	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ルーティング テーブルでの集約アドレスの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router bgp** *autonomous-system* 例： `Device(config)# router bgp 106`	BGP ルータ コンフィギュレーション モードを開始します。
ステップ 4	**aggregate-address** *address mask* 例： `Device(config-router)# aggregate-address 10.0.0.0 255.0.0.0`	BGP ルーティング テーブル内に集約エントリを作成します。集約ルートはASからのルートとしてアドバタイズされます。情報が失われた可能性があることを示すため、アトミック集約属性が設定されます。
ステップ 5	**aggregate-address** *address mask* **as-set** 例： `Device(config-router)# aggregate-address 10.0.0.0 255.0.0.0 as-set`	（任意）AS 設定パス情報を生成します。このコマンドは、この前のコマンドと同じルールに従う集約エントリを作成します。ただし、アドバタイズされるパスは、すべてのパスに含まれる全要素で構成されるAS_SETです。多くのパスを集約するときは、このキーワードを使用しないでください。このルートは絶えず取り消され、アップデートされます。
ステップ 6	**aggregate-address** *address-mask* **summary-only** 例： `Device(config-router)# aggregate-address 10.0.0.0 255.0.0.0 summary-only`	（任意）サマリー アドレスだけをアドバタイズします。
ステップ 7	**aggregate-address** *address mask* **suppress-map** *map-name* 例： `Device(config-router)# aggregate-address 10.0.0.0 255.0.0.0 suppress-map map1`	（任意）選択された、より具体的なルートを抑制します。

	コマンドまたはアクション	目的
ステップ 8	**aggregate-address** *address mask* **advertise-map** *map-name* 例： `Device(config-router)# aggregate-address 10.0.0.0 255.0.0.0 advertise-map map2`	（任意）ルート マップによって指定された設定に基づいて集約を生成します。
ステップ 9	**aggregate-address** *address mask* **attribute-map** *map-name* 例： `Device(config-router)# aggregate-address 10.0.0.0 255.0.0.0 attribute-map map3`	（任意）ルート マップで指定された属性を持つ集約を生成します。
ステップ 10	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 11	**show ip bgp neighbors** [**advertised-routes**] 例： `Device# show ip bgp neighbors`	設定を確認します。
ステップ 12	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ルーティング ドメイン連合の設定

自律システムのグループの自律システム番号として機能する連合 ID を指定する必要があります。

手順

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

	コマンドまたはアクション	目的
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router bgp** *autonomous-system* 例： `Device(config)# router bgp 100`	BGP ルータ コンフィギュレーション モードを開始します。
ステップ 4	**bgp confederation identifier** *autonomous-system* 例： `Device(config)# bgp confederation identifier 50007`	BGP 連合 ID を設定します。
ステップ 5	**bgp confederation peers** *autonomous-system* [*autonomous-system ...*] 例： `Device(config)# bgp confederation peers 51000 51001 51002`	連合に属する AS、および特殊な EBGP ピアとして処理する AS を指定します。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show ip bgp neighbor** 例： `Device# show ip bgp neighbor`	設定を確認します。
ステップ 8	**show ip bgp network** 例： `Device# show ip bgp network`	設定を確認します。
ステップ 9	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

BGP ルート リフレクタの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router bgp** *autonomous-system* 例： `Device(config)# router bgp 101`	BGP ルータ コンフィギュレーション モードを開始します。
ステップ 4	**neighbor** {*ip-address* | *peer-group-name*} **route-reflector-client** 例： `Device(config-router)# neighbor 172.16.70.24 route-reflector-client`	ローカル ルータを BGP ルート リフレクタとして、指定されたネイバーをクライアントとして、それぞれ設定します。
ステップ 5	**bgp cluster-id** *cluster-id* 例： `Device(config-router)# bgp cluster-id 10.0.1.2`	（任意）クラスタに複数のルート リフレクタが存在する場合、クラスタ ID を設定します。
ステップ 6	**no bgp client-to-client reflection** 例： `Device(config-router)# no bgp client-to-client reflection`	（任意）クライアント間のルート反映を無効にします。デフォルトでは、ルート リフレクタ クライアントからのルートは、他のクライアントに反映されます。ただし、クライアントが完全メッシュ構造の場合、ルート リフレクタはルートをクライアントに反映させる必要がありません。
ステップ 7	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
ステップ 8	**show ip bgp** 例： `Device# show ip bgp`	設定を確認します。送信元 ID およびクラスタリスト属性を表示します。
ステップ 9	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ルート ダンプニングの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router bgp** *autonomous-system* 例： `Device(config)# router bgp 100`	BGP ルータ コンフィギュレーション モードを開始します。
ステップ 4	**bgp dampening** 例： `Device(config-router)# bgp dampening`	BGP ルート ダンプニングを有効にします。
ステップ 5	**bgp dampening** *half-life reuse suppress max-suppress* [**route-map** *map*] 例： `Device(config-router)# bgp dampening 30 1500 10000 120`	（任意）ルート ダンプニング係数のデフォルト値を変更します。

	コマンドまたはアクション	目的
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show ip bgp flap-statistics** [{ **regexp** *regexp*} | { **filter-list** *list*} | {*address mask* [**longer-prefix**]}] 例： `Device# show ip bgp flap-statistics`	（任意）フラッピングしているすべてのパスのフラップを監視します。ルートの抑制が終了し、安定状態になると、統計情報が削除されます。
ステップ 8	**show ip bgp dampened-paths** 例： `Device# show pi bgp dampened-paths`	（任意）抑制されるまでの時間を含めて、ダンプニングされたルートを表示します。
ステップ 9	**clear ip bgp flap-statistics** [{ **regexp** *regexp*} | { **filter-list** *list*} | {*address mask* [**longer-prefix**]} 例： `Device# clear ip bgp flap-statistics`	（任意）BGP フラップ統計情報を消去して、ルートがダンプニングされる可能性を小さくします。
ステップ 10	**clear ip bgp dampening** 例： `Device# clear ip bgp dampening`	（任意）ルートダンプニング情報を消去して、ルートの抑制を解除します。
ステップ 11	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

BGP ルートの条件付き注入

標準のルート集約を通じて選択された具体性にかけるプレフィックスではなく、より具体的なプレフィックスを BGP ルーティング テーブルに注入するには、この作業を実行します。より具体的なプレフィックスを使用すると、集約されたルートを使う場合よりも、よりきめ細かなトラフィック エンジニアリングや管理制御を行うことができます。

始める前に

この作業は、BGP ピアに対して、IGP がすでに設定されていることを前提にしています。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **bgp inject-map** *inject-map-name* **exist-map** *exist-map-name* [**copy-attributes**]
5. **exit**
6. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
7. **match ip address** {*access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...* | *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]}
8. **match ip route-source** {*access-list-number* | *access-list-name*} [*access-list-number...*| *access-list-name...*]
9. **exit**
10. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
11. **set ip address** {*access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...* | *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]}
12. **set community** {*community-number* [**additive**] [*well-known-community*] | **none**}
13. **exit**
14. **ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network/length* | **permit** *network/length*} [**ge** *ge-value*] [**le** *le-value*]
15. 作成される各プレフィックス リストについて、ステップ 14 を繰り返します。
16. **exit**
17. **show ip bgp injected-paths**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router bgp** *autonomous-system-number* 例： `Device(config)# router bgp 40000`	指定したルーティング プロセスのルータ コンフィギュレーション モードを開始します。
ステップ 4	**bgp inject-map** *inject-map-name* **exist-map** *exist-map-name* [**copy-attributes**] 例：	条件付きルート注入のために、注入マップと存在マップを指定します。

	コマンドまたはアクション	目的
	`Device(config-router)# bgp inject-map ORIGINATE exist-map LEARNED_PATH`	• 注入したルートが集約ルートの属性を継承することを指定するには、**copy-attributes** キーワードを使用します。
ステップ 5	**exit** 例： `Device(config-router)# exit`	ルータ コンフィギュレーション モードを終了し、グローバル コンフィギュレーション モードに戻ります。
ステップ 6	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] 例： `Device(config)# route-map LEARNED_PATH permit 10`	ルート マップを設定し、ルート マップ コンフィギュレーション モードを開始します。
ステップ 7	**match ip address** {*access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...* | *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]} 例： `Device(config-route-map)# match ip address prefix-list SOURCE`	より具体的なルートの注入先となる集約ルートを指定します。 • この例では、ルートのソースの再配布に、プレフィックス リスト SOURCE が使用されています。
ステップ 8	**match ip route-source** {*access-list-number* | *access-list-name*} [*access-list-number...*| *access-list-name...*] 例： `Device(config-route-map)# match ip route-source prefix-list ROUTE_SOURCE`	ルートのソースを再配布するための一致条件を指定します。 • この例では、ルートのソースの再配布に、プレフィックス リスト ROUTE_SOURCE が使用されています。 （注） ルート ソースは、**neighbor remote-as** コマンドで設定されたネイバー アドレスです。より具体的なルートの注入先とな注入る集約ルートを指定します。
ステップ 9	**exit** 例： `Device(config-route-map)# exit`	ルート マップ コンフィギュレーション モードを終了して、グローバル コンフィギュレーション モードを開始します。
ステップ 10	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] 例： `Device(config)# route-map ORIGINATE permit 10`	ルート マップを設定し、ルート マップ コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 11	**set ip address** {*access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...* | *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]} 例： `Device(config-route-map)# set ip address prefix-list ORIGINATED_ROUTES`	注入されるルートを指定します。 • この例では、ルートのソースの再配布に、プレフィックス リスト originated_routes が使用されています。
ステップ 12	**set community** {*community-number* [**additive**] [*well-known-community*] | **none**} 例： `Device(config-route-map)# set community 14616:555 additive`	注入されたルートの BGP コミュニティ属性を設定します。
ステップ 13	**exit** 例： `Device(config-route-map)# exit`	ルート マップ コンフィギュレーション モードを終了して、グローバル コンフィギュレーション モードを開始します。
ステップ 14	**ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network/length* | **permit** *network/length*} [**ge** *ge-value*] [**le** *le-value*] 例： `Device(config)# ip prefix-list SOURCE permit 10.1.1.0/24`	プレフィックス リストを設定します。 • この例では、プレフィックス リスト SOURCE は、ネットワーク 10.1.1.0/24 からのルートを許可するように設定されています。
ステップ 15	作成される各プレフィックス リストについて、ステップ 14 を繰り返します。	--
ステップ 16	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 17	**show ip bgp injected-paths** 例： `Device# show ip bgp injected-paths`	（任意）注入されたパスに関する情報を表示します。

ピア セッション テンプレートの設定

次の作業では、ピアセッションテンプレートを作成し、設定します。

基本的なピア セッション テンプレートの設定

一般的な BGP ルーティング セッション コマンドを使って、この次に説明する 2 つの作業のうち 1 つを使用して、多数のネイバーに適用できる基本的なピア セッション テンプレートを作成するには、この作業を実行します。

（注） ステップ 5 と 6 のコマンドは任意で、サポートされている一般的なセッション コマンドのいずれとでも置き換えが可能です。

（注） ピア セッション テンプレートには、次の制約事項が適用されます。

- ピアセッションテンプレートが直接継承できるセッションテンプレートは1つだけです。また、継承されたセッション テンプレートはそれぞれ、間接継承されたセッション テンプレートを1つ含むことができます。したがって、ネイバー、またはネイバー グループの設定には、直接適用されたピア セッション テンプレートを 1 個だけと、間接継承されたピア セッション テンプレートを 7 個使用できます。
- BGP ネイバーを、ピア グループとピア テンプレートの両方と連動するようには設定できません。BGP ネイバーは、1 つのピア グループだけに属するように設定するか、またはピア テンプレートだけからポリシーを継承するように設定できます。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **template peer-session** *session-template-name*
5. **remote-as** *autonomous-system-number*
6. **timers** *keepalive-interval hold-time*
7. **end**
8. **show ip bgp template peer-session** [*session-template-name*]

手順の詳細

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

	コマンドまたはアクション	目的
	`Device# configure terminal`	
ステップ 3	**router bgp** *autonomous-system-number* 例： `Device(config)# router bgp 101`	ルータ コンフィギュレーション モードを開始して、BGP ルーティング プロセスを作成します。
ステップ 4	**template peer-session** *session-template-name* 例： `Device(config-router)# template peer-session INTERNAL-BGP`	セッション テンプレート コンフィギュレーション モードを開始して、ピア セッション テンプレートを作成します。
ステップ 5	**remote-as** *autonomous-system-number* 例： `Device(config-router-stmp)# remote-as 202`	（任意）指定された自律システムでリモート ネイバーとのピアリングを設定します。 （注） ここでは、サポートされている一般セッション コマンドならどれでも使用できます。サポートされているコマンドの一覧については、「制限事項」の項を参照してください。
ステップ 6	**timers** *keepalive-interval hold-time* 例： `Device(config-router-stmp)# timers 30 300`	（任意）BGP キープアライブとホールドタイマーを設定します。 • ホールドタイムは、少なくともキープアライブタイムの 2 倍の長さが必要です。 （注） ここでは、サポートされている一般セッション コマンドならどれでも使用できます。サポートされているコマンドの一覧については、「制限事項」の項を参照してください。
ステップ 7	**end** 例： `Device(config-router)# end`	セッション テンプレート コンフィギュレーション モードを終了して、特権 EXEC モードに戻ります。
ステップ 8	**show ip bgp template peer-session** [*session-template-name*] 例： `Device# show ip bgp template peer-session`	ローカルに設定されたピア セッション テンプレートを表示します。 • *session-template-name* 引数を使用して、ピア ポリシー テンプレートが 1 つだけ表示されるように、出力をフィルタ処理できます。また、この

	コマンドまたはアクション	目的
		コマンドは、標準出力修飾子すべてをサポートしています。

inherit peer-session コマンドを使用したピア セッション テンプレートの継承の設定

この作業は、**inherit peer-session** コマンドを使用して、ピア セッション テンプレートの継承を設定します。これは、ピア セッション テンプレートを作成、設定し、別のピア セッション テンプレートからコンフィギュレーションを継承できるようにします。

（注） ステップ 5 と 6 のコマンドは任意で、サポートされている一般的なセッション コマンドのいずれとでも置き換えが可能です。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **template peer-session** *session-template-name*
5. **description** *text-string*
6. **update-source** *interface-type interface-number*
7. **inherit peer-session** *session-template-name*
8. **end**
9. **show ip bgp template peer-session** [*session-template-name*]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router bgp** *autonomous-system-number* 例： `Device(config)# router bgp 101`	ルータ コンフィギュレーション モードを開始して、BGP ルーティング プロセスを作成します。

	コマンドまたはアクション	目的
ステップ 4	**template peer-session** *session-template-name* 例： `Device(config-router)# template peer-session CORE1`	セッション テンプレート コンフィギュレーション モードを開始して、ピア セッション テンプレートを作成します。
ステップ 5	**description** *text-string* 例： `Device(config-router-stmp)# description CORE-123`	（任意）説明を設定します。 • text-string には最大 80 文字を使用できます。 （注） ここでは、サポートされている一般セッション コマンドならどれでも使用できます。サポートされているコマンドの一覧については、「制限事項」の項を参照してください。
ステップ 6	**update-source** *interface-type interface-number* 例： `Device(config-router-stmp)# update-source loopback 1`	（任意）ルーティング テーブル アップデートを受信するための特定のソース、またはインターフェイスを選択するようにルータを設定します。 • この例では、ループバック インターフェイスを使用します。このコンフィギュレーションの利点は、ループバック インターフェイスはフラッピングしているインターフェイスの影響を受けにくいところにあります。 （注） ここでは、サポートされている一般セッション コマンドならどれでも使用できます。サポートされているコマンドの一覧については、「制限事項」の項を参照してください。
ステップ 7	**inherit peer-session** *session-template-name* 例： `Device(config-router-stmp)# inherit peer-session INTERNAL-BGP`	別のピア セッション テンプレートのコンフィギュレーションを継承するように、このピアセッション テンプレートを設定します。 • この例では、INTERNAL-BGPからコンフィギュレーションを継承するようにピア セッション テンプレートを設定しています。このテンプレートはネイバーに適用可能で、コンフィギュレーション INTERNAL-BGP は間接的に適用されます。その他のピア セッション テンプレートは直接適用できません。ただし、直接継承されたテンプレートは最高 7 個の間接継承されたピア セッション テンプレートを持つことができます。

	コマンドまたはアクション	目的
ステップ 8	**end** 例： `Device(config-router)# end`	セッション テンプレート コンフィギュレーション モードを終了して、特権 EXEC モードを開始します。
ステップ 9	**show ip bgp template peer-session** [*session-template-name*] 例： `Device# show ip bgp template peer-session`	ローカルに設定されたピア セッション テンプレートを表示します。 • オプションの *session-template-name* 引数を使用して、ピア ポリシー テンプレートが 1 つだけ表示されるように、出力をフィルタ処理できます。また、このコマンドは、標準出力修飾子すべてをサポートしています。

neighbor inherit peer-session コマンドを使用したピア セッション テンプレートの継承の設定

この作業では、**neighbor inherit peer-session** コマンドを使用して、ピア セッション テンプレートをネイバーに送信し、指定されたピア セッション テンプレートからコンフィギュレーションを継承させるようにデバイスを設定します。次の手順に従って、ピア セッション テンプレート コンフィギュレーションをネイバーに送信し、継承させます。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
5. **neighbor** *ip-address* **inherit peer-session** *session-template-name*
6. **end**
7. **show ip bgp template peer-session** [*session-template-name*]

手順の詳細

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

	コマンドまたはアクション	目的
ステップ 3	**router bgp** *autonomous-system-number* 例： `Device(config)# router bgp 101`	ルータ コンフィギュレーション モードを開始して、BGP ルーティング プロセスを作成します。
ステップ 4	**neighbor** *ip-address* **remote-as** *autonomous-system-number* 例： `Device(config-router)# neighbor 172.16.0.1 remote-as 202`	指定されたネイバーを使ってピアリング セッションを設定します。 • 手順 5 の neighbor inherit 文を動作させるには、**remote-as** 文を明示的に使用する必要があります。ピアリングが設定されていない場合、手順 5 で指定されたネイバーはセッション テンプレートを受け付けません。
ステップ 5	**neighbor** *ip-address* **inherit peer-session** *session-template-name* 例： `Device(config-router)# neighbor 172.16.0.1 inherit peer-session CORE1`	ネイバーがコンフィギュレーションを継承できるように、このネイバーにピア セッション テンプレートを送信します。 • この例では、ピア セッション テンプレート CORE1 を 172.16.0.1 ネイバーに送信し、継承させるようにデバイスを設定しています。このテンプレートはネイバーに適用できます。また、別のピア セッション テンプレートが CORE1 で間接継承された場合、間接継承されたコンフィギュレーションも適用されます。その他のピア セッション テンプレートは直接適用できません。ただし、直接継承されたテンプレートも、さらに最高 7 個の間接継承されたピア セッション テンプレートを継承することができます。
ステップ 6	**end** 例： `Device(config-router)# end`	ルータ コンフィギュレーション モードを終了して、特権 EXEC モードを開始します。
ステップ 7	**show ip bgp template peer-session** [*session-template-name*] 例： `Device# show ip bgp template peer-session`	ローカルに設定されたピア セッション テンプレートを表示します。 • オプションの *session-template-name* 引数を使用して、ピア ポリシー テンプレートが 1 つだけ表示されるように、出力をフィルタ処理できます。また、このコマンドは、標準出力修飾子すべてをサポートしています。

ピア ポリシー テンプレートの設定

次の作業では、ピアポリシーテンプレートを作成し、設定します。

基本的なピア ポリシー テンプレートの設定

BGP ポリシー コンフィギュレーション コマンドを使って、この次に説明する 2 つの作業のうち 1 つを使用して、多数のネイバーに適用できる基本的なピア ポリシー テンプレートを作成するには、この作業を実行します。

（注） ステップ 5 ～ 7 のコマンドは任意で、サポートされている BGP ポリシー コンフィギュレーション コマンドのいずれとでも置き換えが可能です。

（注） ピア ポリシー テンプレートには、次の制約事項が適用されます。

- ピア ポリシー テンプレートは、直接的、または間接的に、最高 8 個のピア ポリシー テンプレートを継承できます。
- BGP ネイバーを、ピア グループとピア テンプレートの両方と連動するようには設定できません。BGP ネイバーは、1 つのピア グループだけに属するように設定するか、またはピア テンプレートだけからポリシーを継承するように設定できます。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **template peer-policy** *policy-template-name*
5. **maximum-prefix** *prefix-limit* [*threshold*] [**restart** *restart-interval* | **warning-only**]
6. **weight** *weight-value*
7. **prefix-list** *prefix-list-name* {**in** | **out**}
8. **end**

手順の詳細

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

	コマンドまたはアクション	目的
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router bgp** *autonomous-system-number* 例： `Device(config)# router bgp 45000`	ルータ コンフィギュレーション モードを開始して、BGP ルーティング プロセスを作成します。
ステップ 4	**template peer-policy** *policy-template-name* 例： `Device(config-router)# template peer-policy GLOBAL`	ポリシーテンプレート コンフィギュレーション モードを開始し、ピア ポリシー テンプレートを作成します。
ステップ 5	**maximum-prefix** *prefix-limit* [*threshold*] [**restart** *restart-interval* | **warning-only**] 例： `Device(config-router-ptmp)# maximum-prefix 10000`	（任意）このピアがネイバーから受け入れるプレフィックスの最大数を設定します。 （注） ここでは、サポートされている BGP ポリシー コンフィギュレーション コマンドならどれでも使用できます。サポートされているコマンドの一覧については、「ピア ポリシー テンプレート」の項を参照してください。
ステップ 6	**weight** *weight-value* 例： `Device(config-router-ptmp)# weight 300`	（任意）このネイバーから送信されるルートのデフォルトの重みを設定します。 （注） ここでは、サポートされている BGP ポリシー コンフィギュレーション コマンドならどれでも使用できます。サポートされているコマンドの一覧については、「ピア ポリシー テンプレート」の項を参照してください。
ステップ 7	**prefix-list** *prefix-list-name* {**in** | **out**} 例： `Device(config-router-ptmp)# prefix-list NO-MARKETING in`	（任意）ルータにより受信、またはルータから送信されるプレフィックスをフィルタします。 • この例のプレフィックス リストは、インバウンド内部アドレスをフィルタします。

	コマンドまたはアクション	目的
		（注） ここでは、サポートされている BGP ポリシー コンフィギュレーション コマンドならどれでも使用できます。サポートされているコマンドの一覧については、「ピア ポリシー テンプレート」の項を参照してください。
ステップ 8	**end** 例： `Device(config-router-ptmp)# end`	ポリシーテンプレートコンフィギュレーションモードを終了して、特権 EXEC モードに戻ります。

inherit peer-policy コマンドを使用したピア ポリシー テンプレートの継承の設定

この作業は、**inherit peer-policy** コマンドを使用して、ピア ポリシー テンプレートの継承を設定します。これは、ピア ポリシー テンプレートを作成、設定し、別のピア ポリシー テンプレートからコンフィギュレーションを継承できるようにします。

（注） ステップ 5 と 6 のコマンドは任意で、サポートされている BGP ポリシーコンフィギュレーション コマンドのいずれとでも置き換えが可能です。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **template peer-policy** *policy-template-name*
5. **route-map** *map-name* {**in**| **out**}
6. **inherit peer-policy** *policy-template-name sequence-number*
7. **end**
8. **show ip bgp template peer-policy** [*policy-template-name*[**detail**]]

手順の詳細

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

	コマンドまたはアクション	目的
	`Device# configure terminal`	
ステップ 3	**router bgp** *autonomous-system-number* 例： `Device(config)# router bgp 45000`	ルータ コンフィギュレーション モードを開始して、BGP ルーティング プロセスを作成します。
ステップ 4	**template peer-policy** *policy-template-name* 例： `Device(config-router)# template peer-policy NETWORK1`	ポリシーテンプレート コンフィギュレーション モードを開始し、ピア ポリシー テンプレートを作成します。
ステップ 5	**route-map** *map-name* {**in**| **out**} 例： `Device(config-router-ptmp)# route-map ROUTE in`	（任意）指定されたルート マップをインバウンド ルート、またはアウトバウンド ルートに適用します。 （注） ここでは、サポートされている BGP ポリシー コンフィギュレーション コマンドならどれでも使用できます。
ステップ 6	**inherit peer-policy** *policy-template-name sequence-number* 例： `Device(config-router-ptmp)# inherit peer-policy GLOBAL 10`	別のピア ポリシーテンプレートのコンフィギュレーションを継承するように、このピア ポリシー テンプレートを設定します。 • *sequence-number* 引数は、ピア ポリシー テンプレートの評価順序を設定します。ルート マップのシーケンス番号と同様、最も小さいシーケンス番号が最初に評価されます。 • この例では、GLOBAL からコンフィギュレーションを継承するようにピア ポリシー テンプレートを設定しています。これらの手順で作成されたテンプレートをネイバーに適用すると、コンフィギュレーション GLOBAL も間接継承され、適用されます。GLOBAL からはさらに最高 6 個のピア ポリシー テンプレートが間接継承され、合計 8 個のピア ポリシー テンプレートが直接適用、および間接継承されます。 • 他のテンプレートで、これより小さいシーケンス番号が設定されていなければ、この例のこのテンプレートが最初に評価されます。
ステップ 7	**end** 例：	ポリシーテンプレート コンフィギュレーション モードを終了して、特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
	`Device(config-router-ptmp)# end`	
ステップ 8	**show ip bgp template peer-policy** [*policy-template-name*[**detail**]] 例： `Device# show ip bgp template peer-policy NETWORK1 detail`	ローカルに設定されたピア ポリシー テンプレートを表示します。 • *policy-template-name* 引数を使用して、ピア ポリシー テンプレートが 1 つだけ表示されるように、出力をフィルタ処理できます。また、このコマンドは、標準出力修飾子すべてをサポートしています。 • 詳細なポリシー情報を表示するには、**detail** キーワードを使用します。

例

次の例は、**show ip bgp template peer-policy** コマンドに **detail** キーワードを付けた場合の出力で、NETWORK1 というポリシーの詳細が表示されています。この例の出力からは、GLOBAL テンプレートが継承されたことがわかります。ルート マップおよびプレフィックス リスト コンフィギュレーションの詳細も表示されています。

```
Device# show ip bgp template peer-policy NETWORK1 detail
Template:NETWORK1, index:2.
Local policies:0x1, Inherited polices:0x80840
This template inherits:
  GLOBAL, index:1, seq_no:10, flags:0x1
Locally configured policies:
  route-map ROUTE in
Inherited policies:
  prefix-list NO-MARKETING in
  weight 300
  maximum-prefix 10000
 Template:NETWORK1 <detail>
Locally configured policies:
  route-map ROUTE in
route-map ROUTE, permit, sequence 10
  Match clauses:
    ip address prefix-lists: DEFAULT
ip prefix-list DEFAULT: 1 entries
   seq 5 permit 10.1.1.0/24
  Set clauses:
  Policy routing matches: 0 packets, 0 bytes
Inherited policies:
  prefix-list NO-MARKETING in
ip prefix-list NO-MARKETING: 1 entries
   seq 5 deny 10.2.2.0/24
```

neighbor inherit peer-policy コマンドを使用したピア ポリシー テンプレートの継承の設定

この作業では、**neighbor inherit peer-policy** コマンドを使用して、ピアポリシーテンプレートをネイバーに送信し、継承させるようにデバイスを設定します。次の手順に従って、ピア ポリシー テンプレート コンフィギュレーションをネイバーに送信し、継承させます。

BGP ネイバーが複数レベルのピア テンプレートを使用する場合、ネイバーに適用されているポリシーを判断するのが難しいことがあります。**show ip bgp neighbors** コマンドの **policy** および **detail** キーワードは、指定されたネイバーに継承されたポリシーおよび直接設定されたポリシーを表示します。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
5. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
6. **neighbor** *ip-address* **inherit peer-policy** *policy-template-name*
7. **end**
8. **show ip bgp neighbors** [*ip-address*[**policy** [**detail**]]]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router bgp** *autonomous-system-number* 例： `Device(config)# router bgp 45000`	ルータ コンフィギュレーション モードを開始して、BGP ルーティング プロセスを作成します。
ステップ 4	**neighbor** *ip-address* **remote-as** *autonomous-system-number* 例： `Device(config-router)# neighbor 192.168.1.2 remote-as 40000`	指定されたネイバーを使ってピアリング セッションを設定します。 • 手順 6 の **neighbor inherit** 文を動作させるには、**remote-as** 文を明示的に使用する必要があります。ピアリングが設定されていない場合、手順 6 で指定されたネイバーはセッション テンプレートを受け付けません。

	コマンドまたはアクション	目的
ステップ 5	**address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*] 例： `Device(config-router)# address-family ipv4 unicast`	アドレス ファミリ固有のコマンドコンフィギュレーションを使用するようにネイバーを設定するために、アドレス ファミリ コンフィギュレーション モードを開始します。
ステップ 6	**neighbor** *ip-address* **inherit peer-policy** *policy-template-name* 例： `Device(config-router-af)# neighbor 192.168.1.2 inherit peer-policy GLOBAL`	ネイバーが設定を継承できるように、ピア ポリシー テンプレートをこのネイバーに送信します。 • この例では、ピア ポリシー テンプレート GLOBAL を 192.168.1.2 ネイバーに送信し、継承させるようにルータを設定しています。このテンプレートはネイバーに適用できます。また、別のピア ポリシー テンプレートが GLOBAL から間接継承された場合、間接継承されたコンフィギュレーションも適用されます。GLOBAL からは、さらに最高 7 個のピア ポリシー テンプレートを間接継承できます。
ステップ 7	**end** 例： `Device(config-router-af)# end`	アドレス ファミリ コンフィギュレーション モードを終了して、特権 EXEC モードに戻ります。
ステップ 8	**show ip bgp neighbors** [*ip-address*[**policy** [**detail**]]] 例： `Device# show ip bgp neighbors 192.168.1.2 policy`	ローカルに設定されたピア ポリシー テンプレートを表示します。 • *policy-template-name* 引数を使用して、ピア ポリシー テンプレートが 1 つだけ表示されるように、出力をフィルタ処理できます。また、このコマンドは、標準出力修飾子すべてをサポートしています。 • このネイバーに適用されているポリシーをアドレス ファミリごとに表示するには、**policy** キーワードを使用します。 • 詳細なポリシー情報を表示するには、**detail** キーワードを使用します。

例

次の出力例に表示されているのは、192.168.1.2 にあるネイバーに適用されたポリシーです。この出力には、継承されたポリシーと、このネイバーデバイスで設定されたポリシーの両方が表示されています。継承されたポリシーは、ピア グループ、またはピア ポリシー テンプレートからネイバーが継承したポリシーです。

```
Device# show ip bgp neighbors 192.168.1.2 policy
Neighbor: 192.168.1.2, Address-Family: IPv4 Unicast
Locally configured policies:
 route-map ROUTE in
Inherited polices:
 prefix-list NO-MARKETING in
 route-map ROUTE in
 weight 300
 maximum-prefix 10000
```

BGP ルートマップの next-hop self の設定

ip next-hop self 設定を追加し、bgp next-hop unchanged 設定と bgp next-hop unchanged allpaths 設定をオーバーライドして、既存のルート マップを変更するには、この作業を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag* **permit** *sequence-number*
4. **match source-protocol** *source-protocol*
5. **set ip next-hop self**
6. **exit**
7. **route-map** *map-tag* **permit** *sequence-number*
8. **match route-type internal**
9. **match route-type external**
10. **match source-protocol** *source-protocol*
11. **exit**
12. **router bgp** *autonomous-system-number*
13. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
14. **address-family vpnv4**
15. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**
16. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **next-hop unchanged allpaths**
17. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-map** *map-name* **out**
18. **exit**
19. **address-family ipv4** [**unicast** | **multicast**| **vrf** *vrf-name*]
20. **bgp route-map priority**
21. **redistribute** *protocol*
22. **redistribute** *protocol*
23. **exit-address-family**
24. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**route-map** *map-tag* **permit** *sequence-number* 例： `Device(config)# route-map static-nexthop-rewrite permit 10`	ルーティング プロトコル間でルートを再配布する条件を定義し、ルートマップ コンフィギュレーション モードを開始します。
ステップ 4	**match source-protocol** *source-protocol* 例： `Device(config-route-map)# match source-protocol static`	送信元プロトコルに基づいて、Enhanced Interior Gateway Routing Protocol（EIGRP）の外部ルートを照合します。
ステップ 5	**set ip next-hop self** 例： `Device(config-route-map)# set ip next-hop self`	自身をネクスト ホップとするようにローカル ルート（BGP の場合のみ）を設定します。
ステップ 6	**exit** 例： `Device(config-route-map)# exit`	ルート マップ コンフィギュレーション モードを終了し、グローバル コンフィギュレーション モードを開始します。
ステップ 7	**route-map** *map-tag* **permit** *sequence-number* 例： `Device(config)# route-map static-nexthop-rewrite permit 20`	ルーティング プロトコル間でルートを再配布する条件を定義し、ルートマップ コンフィギュレーション モードを開始します。
ステップ 8	**match route-type internal** 例： `Device(config-route-map)# match route-type internal`	指定されたタイプのルートを再配布します。

	コマンドまたはアクション	目的
ステップ 9	**match route-type external** 例： `Device(config-route-map)# match route-type external`	指定されたタイプのルートを再配布します。
ステップ 10	**match source-protocol** *source-protocol* 例： `Device(config-route-map)# match source-protocol connected`	送信元プロトコルに基づいて、Enhanced Interior Gateway Routing Protocol（EIGRP）の外部ルートを照合します。
ステップ 11	**exit** 例： `Device(config-route-map)# exit`	ルート マップ コンフィギュレーション モードを終了し、グローバル コンフィギュレーション モードを開始します。
ステップ 12	**router bgp** *autonomous-system-number* 例： `Device(config)# router bgp 45000`	ルータ コンフィギュレーション モードを開始して、BGP ルーティング プロセスを作成します。
ステップ 13	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *autonomous-system-number* 例： `Device(config-router)# neighbor 172.16.232.50 remote-as 65001`	BGP ネイバー テーブルまたはマルチプロトコル BGP ネイバー テーブルにエントリを追加します。
ステップ 14	**address-family vpnv4** 例： `Device(config-router)# address-family vpnv4`	VPNv4 アドレス ファミリを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。
ステップ 15	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate** 例： `Device(config-router-af)# neighbor 172.16.232.50 activate`	ボーダー ゲートウェイ プロトコル（BGP）ネイバーとの情報交換を有効にします。
ステップ 16	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **next-hop unchanged allpaths** 例： `Device(config-router-af)# neighbor 172.16.232.50 next-hop unchanged allpaths`	マルチホップとして設定されている外部 EBGP ピアで、ネクスト ホップを変更せずに伝播できるようにします。

	コマンドまたはアクション	目的
ステップ 17	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-map** *map-name* **out** 例： `Device(config-router-af)# neighbor 172.16.232.50 route-map static-nexthop-rewrite out`	発信ルートにルート マップを適用します。
ステップ 18	**exit** 例： `Device(config-router-af)# exit`	アドレス ファミリ コンフィギュレーション モードを終了して、ルータ コンフィギュレーション モードを開始します。
ステップ 19	**address-family ipv4** [**unicast** | **multicast**| **vrf** *vrf-name*] 例： `Device(config-router)# address-family ipv4 unicast vrf inside`	IPv4 アドレス ファミリを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。
ステップ 20	**bgp route-map priority** 例： `Device(config-router-af)# bgp route-map priority`	ローカル BGP ルーティング プロセスについてルート マップを優先することを設定します。
ステップ 21	**redistribute** *protocol* 例： `Device(config-router-af)# redistribute static`	ルートを 1 つのルーティング ドメインから他のルーティング ドメインに再配布します。
ステップ 22	**redistribute** *protocol* 例： `Device(config-router-af)# redistribute connected`	ルートを 1 つのルーティング ドメインから他のルーティング ドメインに再配布します。
ステップ 23	**exit-address-family** 例： `Device(config-router-af)# exit address-family`	アドレス ファミリ コンフィギュレーション モードを終了し、ルータ コンフィギュレーション モードを開始します。
ステップ 24	**end** 例： `Device(config-router)# end`	ルータ コンフィギュレーション モードを終了して、特権 EXEC モードを開始します。

BGP の設定例

例：条件付き BGP ルートの挿入の設定

次の出力例は、**show ip bgp injected-paths** コマンドを入力したときに表示される出力に類似しています。

```
Device# show ip bgp injected-paths

BGP table version is 11, local router ID is 10.0.0.1
Status codes:s suppressed, d damped, h history, * valid, > best, i -
internal
Origin codes:i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 172.16.0.0       10.0.0.2                               0 ?
*> 172.17.0.0/16    10.0.0.2                               0 ?
```

例：ピア セッション テンプレートの設定

次の例は、セッション テンプレート コンフィギュレーション モードで、INTERNAL-BGP という名前のピア セッション テンプレートを作成します。

```
router bgp 45000
 template peer-session INTERNAL-BGP
 remote-as 50000
 timers 30 300
 exit-peer-session
```

次の例は、ピア セッション テンプレート CORE1 を作成します。この例は、INTERNAL-BGP というピア セッション テンプレートのコンフィギュレーションを継承します。

```
router bgp 45000
 template peer-session CORE1
 description CORE-123
 update-source loopback 1
 inherit peer-session INTERNAL-BGP
 exit-peer-session
```

次の例は、CORE1 ピア セッション テンプレートを継承するように、192.168.3.2 ネイバーを設定します。192.168.3.2 ネイバーも、ピア セッション テンプレート INTERNAL-BGP から間接的にコンフィギュレーションを継承します。neighbor inherit 文を動作させるには、**remote-as** 文を明示的に使用する必要があります。ピアリングが設定されていない場合、指定されたネイバーはセッション テンプレートを受け付けません。

```
router bgp 45000
 neighbor 192.168.3.2 remote-as 50000
 neighbor 192.168.3.2 inherit peer-session CORE1
```

例：ピア ポリシー テンプレートの設定

次の例は、GLOBAL という名前のピア ポリシー テンプレートを作成し、ポリシー テンプレート コンフィギュレーション モードを開始します。

```
router bgp 45000
 template peer-policy GLOBAL
  weight 1000
  maximum-prefix 5000
  prefix-list NO_SALES in
  exit-peer-policy
```

次の例は、PRIMARY-IN という名前のピア ポリシー テンプレートを作成し、ポリシー テンプレート コンフィギュレーション モードを開始します。

```
router bgp 45000
 template peer-policy PRIMARY-IN
  prefix-list ALLOW-PRIMARY-A in
  route-map SET-LOCAL in
  weight 2345
  default-originate
  exit-peer-policy
```

次の例は、ピア ポリシー テンプレート CUSTOMER-A を作成します。このピア ポリシー テンプレートは、PRIMARY-IN および GLOBAL という名前のピア ポリシー テンプレートからコンフィギュレーションを継承するように設定されています。

```
router bgp 45000
 template peer-policy CUSTOMER-A
  route-map SET-COMMUNITY in
  filter-list 20 in
  inherit peer-policy PRIMARY-IN 20
  inherit peer-policy GLOBAL 10
  exit-peer-policy
```

次の例は、アドレス ファミリ モードでピア ポリシー テンプレート CUSTOMER-A を継承するように 192.168.2.2 ネイバーを設定します。この例は上の例の続きと仮定しており、上のピア ポリシー テンプレート CUSTOMER-A は PRIMARY-IN および GLOBAL という名前のテンプレートからコンフィギュレーションを継承しているため、192.168.2.2 ネイバーもピア ポリシー テンプレート PRIMARY-IN および GLOBAL から間接継承します。

```
router bgp 45000
 neighbor 192.168.2.2 remote-as 50000
 address-family ipv4 unicast
  neighbor 192.168.2.2 inherit peer-policy CUSTOMER-A
  end
```

例：BGP ルート マップの next-hop self の設定

この項では、BGP ルート マップの next-hop self を設定する方法の例を示します。

この例では、bgp next-hop unchanged と bgp next-hop unchanged allpaths の設定をオーバーライドするネットワークを照合するルート マップを設定します。次に、next-hop self を設定します。その後、指定したアドレス ファミリに対して bgp route-map priority を設定して、指定済みの

ルート マップが bgp next-hop unchanged と bgp next-hop unchanged allpaths の設定よりも優先されるようにします。この設定により、スタティック ルートは自身をネクスト ホップとして再配布されますが、接続されたルートおよび IBGP または EBGP を介して学習されたルートは引き続きネクスト ホップを変更せずに再配布されます。

```
route-map static-nexthop-rewrite permit 10
 match source-protocol static
 set ip next-hop self
route-map static-nexthop-rewrite permit 20
 match route-type internal
 match route-type external
 match source-protocol connected
!
router bgp 65000
 neighbor 172.16.232.50 remote-as 65001
 address-family vpnv4
   neighbor 172.16.232.50 activate
   neighbor 172.16.232.50 next-hop unchanged allpaths
   neighbor 172.16.232.50 route-map static-nexthop-rewrite out
 exit-address-family
 address-family ipv4 unicast vrf inside
   bgp route-map priority
   redistribute static
   redistribute connected
 exit-address-family
end
```

BGP のモニタリングおよびメンテナンス

特定のキャッシュ、テーブル、またはデータベースのすべての内容を削除できます。この作業は、特定の構造の内容が無効になった場合、または無効である疑いがある場合に必要となります。

BGP ルーティング テーブル、キャッシュ、データベースの内容など、特定の統計情報を表示できます。さらに、リソースの利用率を取得したり、ネットワーク問題を解決するための情報を使用することもできます。さらに、ノードの到達可能性に関する情報を表示し、デバイスのパケットが経由するネットワーク内のルーティング パスを検出することもできます。

下の図に、BGP を消去および表示するために使用する特権 EXEC コマンドを示します。

表 *8 : IP BGP* の *clear* および *show* コマンド

clear ip bgp *address*	特定の BGP 接続をリセットします。
clear ip bgp *	すべての BGP 接続をリセットします。
clear ip bgp peer-group *tag*	BGP ピア グループのすべてのメンバを削除します。

show ip bgp *prefix*	プレフィックスがアドバタイズされるピア グループ、またはピア グループに含まれないピアを表示します。ネクスト ホップやローカル プレフィックスなどのプレフィックス属性も表示されます。
show ip bgp cidr-only	サブネットおよびスーパーネット ネットワーク マスクを含むすべてのBGPルートを表示します。
show ip bgp community [*community-number*] [**exact**]	指定されたコミュニティに属するルートを表示します。
show ip bgp community-list *community-list-number* [**exact-match**]	コミュニティ リストで許可されたルートを表示します。
show ip bgp filter-list *access-list-number*	指定された AS パス アクセス リストによって照合されたルートを表示します。
show ip bgp inconsistent-as	送信元のASと矛盾するルートを表示します。
show ip bgp regexp *regular-expression*	コマンドラインに入力された特定の正規表現と一致する AS パスを持つルートを表示します。
show ip bgp	BGP ルーティング テーブルの内容を表示します。
show ip bgp neighbors [*address*]	各ネイバーとのBGP接続およびTCP接続に関する詳細情報を表示します。
show ip bgp neighbors [*address*] [**advertised-routes** | **dampened-routes** | **flap-statistics** | **paths** *regular-expression* | **received-routes** | **routes**]	特定の BGP ネイバーから取得されたルートを表示します。
show ip bgp paths	データベース内のすべてのBGPパスを表示します。
show ip bgp peer-group [*tag*] [**summary**]	BGP ピア グループに関する情報を表示します。
show ip bgp summary	BGP 接続すべての状況を表示します。

bgp log-neighbor changes コマンドは、デフォルトでは有効です。そのため、BGP ネイバーのリセット、起動、またはダウン時に生成されるメッセージをログに記録できます。