Cisco IOS XE Amsterdam 17.3.x（Catalyst 9400 スイッチ）IP ルーティング コンフィギュレーション ガイド

初版：2020 年 7 月 31 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。

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

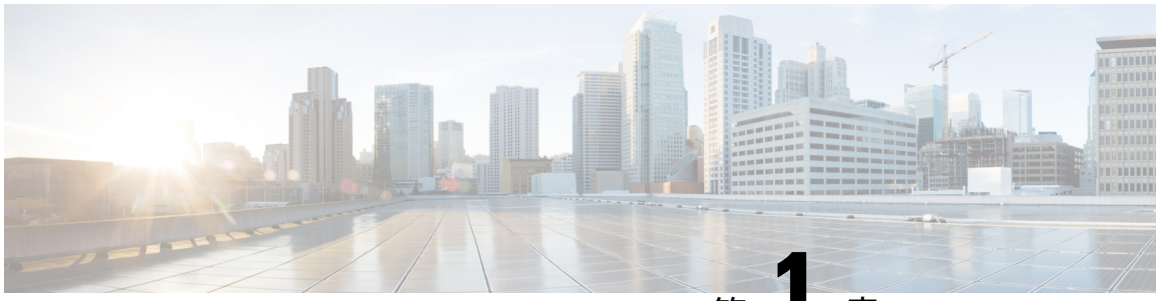
Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/c/en/us/about/legal/trademarks.html. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2020 Cisco Systems, Inc. All rights reserved.



第 1 章

MSDP の設定

- MSDP の設定について （1 ページ）
- MSDP の設定方法 （4 ページ）
- MSDP のモニタリングおよびメンテナンス （26 ページ）
- MSDP の設定例 （27 ページ）
- Multicast Source Discovery Protocol の機能情報 （29 ページ）

MSDP の設定について

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

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

MSDP の概要

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

PIM-SM ドメイン内の RP は、他のドメイン内の MSDP 対応デバイスと MSDP ピアリング関係にあります。ピアリング関係は TCP 接続を通じて発生します。主に、マルチキャスト グループを送信する送信元のリストを交換します。RP 間の TCP 接続は、基本的なルーティング システムによって実現されます。受信側の RP では、送信元リストを使用して送信元のパスが確立されます。

このトポロジの目的は、ドメインから、他のドメイン内のマルチキャスト送信元を検出することです。マルチキャスト送信元がレシーバーのあるドメインを対象としている場合、マルチキャストデータは PIM-SM の通常の送信元ツリー構築メカニズムを通じて配信されます。MSDP

は、グループを送信する送信元のアナウンスにも使用されます。これらのアナウンスは、ドメインの RP で発信する必要があります。

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

MSDP の動作

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

各 MSDP ピアは SA メッセージを発信元の RP から受信して転送し、ピア Reverse-Path Forwarding（RPF）フラッディングを実現します。MSDP デバイスは、BGP または MBGP ルーティング テーブルを調べ、どのピアが SA メッセージの発信元 RP へのネクスト ホップであるかを検出します。このようなピアは *RPF* ピアと呼ばれます。MSDP デバイスでは、RPF ピア以外のすべての MSDP ピアにメッセージが転送されます。BGP および MBGP がサポートされていない場合に MSDP を設定する方法については、デフォルトの MSDP ピアの設定 （4 ページ）を参照してください。

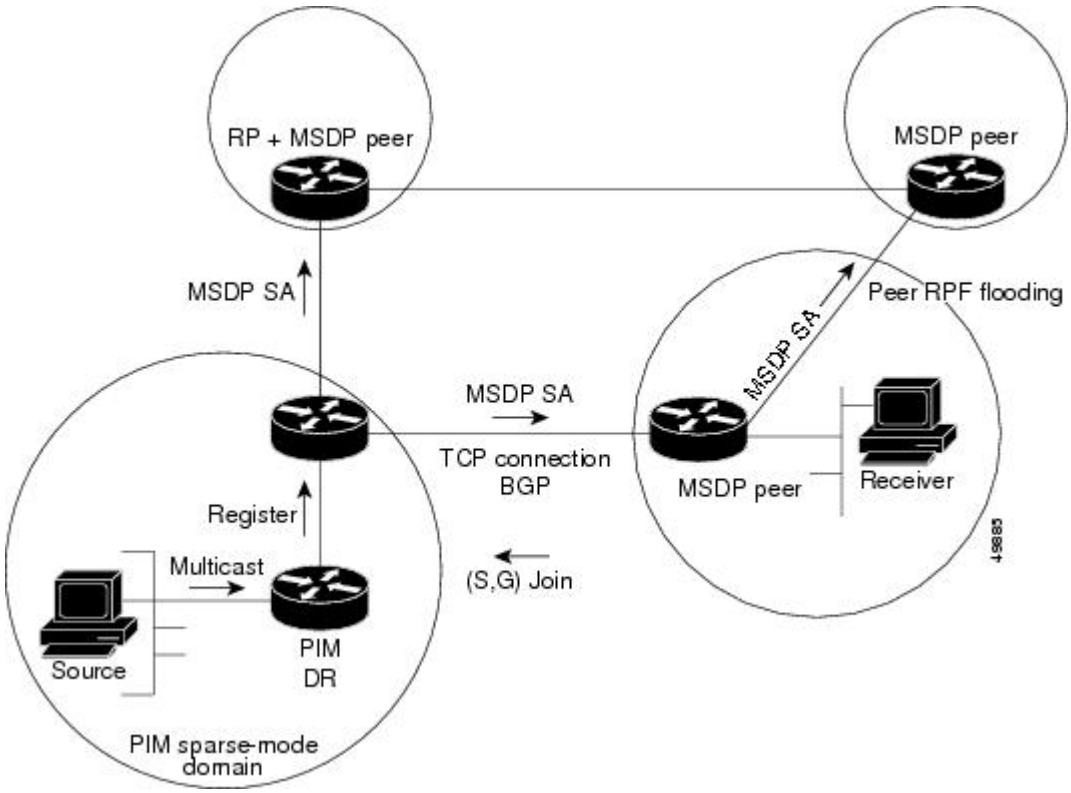
MSDP ピアは、非 RPF ピアから発信元 RP へ向かう同じ SA メッセージを受信すると、そのメッセージをドロップします。それ以外の場合、すべての MSDP ピアにメッセージが転送されます。

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

図 *1 : RP* ピア間で動作する *MSDP*

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

いる場合は、次のシーケンスが発生します。



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

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

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

MSDP の利点

MSDP には次の利点があります。

- 共有されたマルチキャスト配信ツリーが分割され、共有ツリーがドメインに対してローカルになるように設定できます。ローカル メンバーはローカル ツリーに加入します。共有ツリーへの Join メッセージはドメインから脱退する必要はありません。

- PIM SM ドメインは独自の RP だけを信頼するため、他のドメインの RP に対する信頼度が低下します。このため、送信元の情報がドメイン外部に漏れないようにでき、セキュリティが高まります。
- レシーバーだけが配置されているドメインは、グループ メンバーシップをグローバルにアドバタイズしなくても、データを受信できます。
- グローバルな送信元マルチキャスト ルーティング テーブル ステートが不要になり、メモリが削減されます。

MSDP の設定方法

MSDP のデフォルト設定

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

デフォルトの MSDP ピアの設定

始める前に

MSDP ピアを設定します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip msdp default-peer** *ip-address* | *name* [**prefix-list** *list*] 例： `Device(config)#ip msdp default-peer 10.1.1.1 prefix-list site-a`	すべての MSDP SA メッセージの受信元となるデフォルト ピアを定義します。 • *ip-address* | *name* には、MSDP デフォルト ピアの IP アドレスまたはドメイン ネーム システム（DNS）サーバ名を入力します。

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

	コマンドまたはアクション	目的
		• **permit** キーワードを指定すると、条件が一致した場合にアクセスが許可されます。 • *network length* には、許可または拒否されているネットワークの番号およびネットワーク マスク長（ビット単位）を指定します。
ステップ 5	**ip msdp description** {*peer-name* | *peer-address*} *text* 例： `Device(config)#ip msdp description peer-name site-b`	（任意）設定内で、または **show** コマンド出力内で簡単に識別できるように、指定されたピアの説明を設定します。 デフォルトでは、MSDP ピアに説明は関連付けられていません。
ステップ 6	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 7	**show running-config** 例： `Device#show running-config`	入力を確認します。
ステップ 8	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

SA ステートのキャッシング

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

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip msdp cache-sa-state** [**list** *access-list-number*] 例： `Device(config)#ip msdp cache-sa-state 100`	送信元とグループのペアのキャッシングをイネーブルにします（SA ステートを作成します）。アクセス リストを通過したこれらのペアがキャッシュに格納されます。 **list** *access-list-number* の範囲は 100 ～ 199 です。 （注） このコマンドの代わりに、**ip msdp sa-reques** グローバル コンフィギュレーション コマンドを使用できます。この代替コマンドを使用すると、グループの新しいメンバがアクティブになった場合に、SA 要求メッセージがデバイスから MSDP ピアに送信されます。
ステップ 4	**access-list** *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* 例： `Device(config)#access-list 100 permit ip 171.69.0.0 0.0.255.255 224.2.0.0 0.0.255.255`	IP 拡張アクセス リストを作成します。必要な回数だけこのコマンドを繰り返します。 • *access-list-number* の範囲は 100 ～ 199 です。ステップ 2 で作成した番号と同じ値を入力します。 • **deny** キーワードは、条件が一致した場合にアクセスを拒否します。**permit** キーワードは、条件が一致した場合にアクセスを許可します。

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

MSDP ピアからの送信元情報の要求

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

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** **例：** `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip msdp sa-request** {*ip-address* | *name*} **例：** `Device(config)#ip msdp sa-request 171.69.1.1`	指定された MSDP ピアに SA 要求メッセージを送信するようにデバイスを設定します。 *ip-address* | *name* を指定する場合は、グループの新しいメンバーがアクティブになるときにローカルデバイスの SA メッセージの要求元になる MSDP ピアの IP アドレス、または名前を入力します。 SA メッセージを送信する必要がある MSDP ピアごとに、このコマンドを繰り返します。
ステップ 4	**end** **例：** `Device(config)#end`	特権 EXEC モードに戻ります。

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

スイッチから発信される送信元情報の制御

デバイスから発信されるマルチキャスト送信元情報を制御できます。

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

詳細については、送信元の再配信 （10 ページ）およびSA 要求メッセージのフィルタリング （13 ページ）を参照してください。

送信元の再配信

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

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

手順

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

	コマンドまたはアクション	目的
	`Device#configure terminal`	
ステップ 3	**ip msdp redistribute** [**list** *access-list-name*] [**asn** *aspath-access-list-number*] [**route-map** *map*] 例： `Device(config)#ip msdp redistribute list 21`	SA メッセージに格納されてアドバタイズされる、マルチキャスト ルーティング テーブル内の（S,G）エントリを設定します。 デフォルトでは、ローカル ドメイン内の送信元だけがアドバタイズされます。 • （任意） **list** *access-list-name*：IP 標準または IP 拡張アクセスリストの名前または番号を入力します。標準アクセス リストの範囲は 1 ～ 99、拡張アクセス リストの範囲は 100 ～ 199 です。アクセス リストによって、アドバタイズされるローカルな送信元、および送信されるグループが制御されます。 • （任意） **asn** *aspath-access-list-number*：1 ～ 199 の範囲の IP 標準または IP 拡張アクセスリスト番号を入力します。このアクセスリスト番号は、**ip as-path access-list** コマンドでも設定する必要があります。 • （任意） **route-map** *map*：1 ～ 199 の範囲の IP 標準または IP 拡張アクセスリスト番号を入力します。このアクセスリスト番号は、**ip as-path access-list** コマンドでも設定する必要があります。 アクセスリストまたは自律システムパス アクセスリストに従って、デバイスが（S, G）ペアをアドバタイズします。
ステップ 4	次のいずれかを使用します。 • `access-list`*access-list-number* {`deny`|`permit`} *source* [*source-wildcard*]	IP 標準アクセス リストを作成します。必要な回数だけこのコマンドを繰り返します。 または

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

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

SA 要求メッセージのフィルタリング

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

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

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

これらのオプションのいずれかを設定するには、次の手順を実行します。

手順

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

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

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

スイッチで転送される送信元情報の制御

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

フィルタの使用法

フィルタを作成すると、次のいずれかの処理を実行できます。

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

フィルタを適用するには、次の手順を実行します。

手順

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

	コマンドまたはアクション	目的

ステップ 3	次のいずれかを使用します。
	• **ip msdp sa-filter out** {*ip-address name*}
	• **ip msdp sa-filter out** {*ip-address name*} `list` *access-list-number*
	• **ip msdp sa-filter out** {*ip-address name*} `route-map` *map-tag*
	例：
	`Device(config)#ip msdp sa-filter out switch.cisco.com`
	または
	`Device(config)#ip msdp sa-filter out list 100`
	または
	`Device(config)#ip msdp sa-filter out switch.cisco.com route-map 22`
		• 指定された MSDP ピアへの SA メッセージをフィルタリングします。
		• 指定したピアに対する IP 拡張アクセス リストを通過した SA メッセージのみを渡します。拡張アクセスリスト番号の範囲は 100 ～ 199 です。
		list と **route-map** の両方のキーワードを使用すると、すべての条件に一致しなければ、発信 SA メッセージ内のいずれの（S,G）ペアも通過できません。
		• 指定された MSDP ピアへのルートマップ *map-tag* で一致基準を満たす SA メッセージのみを渡します。
		すべての一致基準に当てはまる場合、ルート マップの **permit** がフィルタを通してルートを通過します。**deny** はルートをフィルタ処理します。

ステップ 4	**access-list** *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* 例： `Device(config)#access list 100 permit ip 194.1.22.0 1.1.1.1 194.3.44.0 1.1.1.1`	（任意）IP 拡張アクセス リストを作成します。必要な回数だけこのコマンドを繰り返します。 • *access-list-number* には、ステップ 2 で指定した番号を入力します。 • **deny** キーワードは、条件が一致した場合にアクセスを拒否します。**permit** キーワードは、条件が一致した場合にアクセスを許可します。 • *protocol* には、プロトコル名として **ip** を入力します。 • *source* には、パケットの送信元であるネットワークまたはホストの番号を入力します。 • *source-wildcard* には、送信元に適用するワイルドカード ビットをドット付き 10 進表記で入力します。無

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

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

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

TTL しきい値を確立するには、次の手順に従います。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip msdp ttl-threshold** {*ip-address* | *name*} *ttl* 例： `Device(config)#ip msdp ttl-threshold switch.cisco.com 0`	指定された MSDP ピア宛ての最初の SA メッセージにカプセル化されるマルチキャスト データを制限します。 • *ip-address* | *name* には、TTL の制限が適用される MSDP ピアの IP アドレスまたは名前を入力します。 • *ttl* には、TTL 値を入力します。デフォルトは 0 です。この場合、すべてのマルチキャスト データ パケットは、TTL がなくなるまでピアに転送されます。指定できる範囲は 0 ～ 255 です。
ステップ 4	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `Device#show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

スイッチで受信される送信元情報の制御

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

次のいずれかの処理を実行できます。

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

フィルタを適用するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	次のいずれかを使用します。 • **ip msdp sa-filter in** {*ip-address name*} • **ip msdp sa-filter in** {*ip-address name*} `list` *access-list-number* • **ip msdp sa-filter in** {*ip-address name*} `route-map` *map-tag* 例： `Device(config)#ip msdp sa-filter in switch.cisco.com`	• 指定された MSDP ピアへの SA メッセージをフィルタリングします。 • IP 拡張アクセス リストを通過する、指定されたピアからの SA メッセージのみを通過させます。拡張アクセス リスト *access-list-number* の範囲は 100 ～ 199 です。 **list** と **route-map** の両方のキーワードを使用すると、すべての条件に一致しなければ、発信 SA メッセージ内のいずれの（S,G）ペアも通過できません。 • ルート マップ *map-tag* 内の一致条件を満たす、指定された MSDP ピ

	コマンドまたはアクション	目的

	または
	`Device(config)#ip msdp sa-filter in list 100`
	または
	`Device(config)#ip msdp sa-filter in switch.cisco.com route-map 22`
		アからのSAメッセージのみを通過させます。
		すべての一致基準に当てはまる場合、ルートマップの **permit** がフィルタを通してルートを通過します。**deny** はルートをフィルタ処理します。

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

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

MSDP メッシュ グループの設定

MSDP メッシュ グループは、MSDP によって完全なメッシュ型に相互接続された MSDP スピーカーのグループです。メッシュ グループ内のピアから受信された SA メッセージは、同じメッシュ グループ内の他のピアに転送されません。したがって、SA メッセージのフラッディングが削減され、ピア RPF フラッディングが簡素化されます。ドメイン内に複数の RP がある場合は、**ip msdp mesh-group** グローバル コンフィギュレーション コマンドを使用します。特に、ドメインを越えて SA メッセージを送信する場合に使用します。単一のデバイスに複数のメッシュグループを（異なる名前で）設定できます。

メッシュ グループを作成するには、次の手順を実行します。

手順

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

	コマンドまたはアクション	目的
ステップ 3	**ip msdp mesh-group** *name* {*ip-address* | *name*} 例： `Devic(config)#ip msdp mesh-group 2 switch.cisco.com`	MSDP メッシュ グループを設定し、そのメッシュ グループに属する MSDP ピアを指定します。 デフォルトでは、MSDP ピアはメッシュ グループに属しません。 • *name* には、メッシュ グループの名前を入力します。 • *ip-address* | *name* には、メッシュ グループのメンバーになる MSDP ピアの IP アドレスまたは名前を入力します。 グループ内の MSDP ピアごとに、この手順を繰り返します。
ステップ 4	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `Device#show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

MSDP ピアのシャットダウン

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

ピアをシャットダウンするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip msdp shutdown** {*peer-name* | *peer address*} 例： `Device(config)#ip msdp shutdown switch.cisco.com`	設定情報を保持したまま、指定された MSDP ピアをシャットダウン状態にします。 *peer-name* | *peer address* を指定する場合は、シャットダウンする MSDP ピアの IP アドレスまたは名前を入力します。
ステップ 4	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `Device#show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

境界 PIM デンス モード領域の MSDP への包含

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

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

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

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip msdp border sa-address** *interface-id* 例： `Device(config)#ip msdp border sa-address 0/1`	DM 領域内のアクティブな送信元に関する SA メッセージを送信するように、DM 領域と SM 領域の境界スイッチを設定します。 *interface-id* には、SA メッセージ内の RP アドレスとして使用される、IP アドレスの配信元となるインターフェイスを指定します。 インターフェイスの IP アドレスは、SA メッセージ内の RP フィールド [Originator-ID] の値として使用されます。
ステップ 4	**ip msdp redistribute** [**list** *access-list-name*] [**asn** *aspath-access-list-number*] [**route-map** *map*]	SA メッセージに格納されてアドバタイズされる、マルチキャスト ルーティング テーブル内の（S,G）エントリを設定

	コマンドまたはアクション	目的
	例： `Device(config)#ip msdp redistribute list 100`	します。詳細については、送信元の再配信 （10 ページ）を参照してください。
ステップ 5	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 6	**show running-config** 例： `Device#show running-config`	入力を確認します。
ステップ 7	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

RP アドレス以外の発信元アドレスの設定

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

- MSDP メッシュグループ内の複数のデバイス上で、論理 RP を設定する場合。
- PIM SM ドメインと DM ドメインの境界となるデバイスがある場合。サイトの DM ドメインの境界となるデバイスがあり、SM がその外部で使用されている場合は、DM の送信元を外部に通知する必要があります。このデバイスは RP でないため、SA メッセージで使用される RP アドレスはありません。したがって、このコマンドではインターフェイスのアドレスを指定し、RP アドレスを提供します。

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

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip msdp originator-id** *interface-id* 例： `Device(config)#ip msdp originator-id 0/1`	発信元デバイスのインターフェイスのアドレスとなるように、SA メッセージ内の RP アドレスを設定します。 *interface-id* には、ローカルデバイスのインターフェイスを指定します。
ステップ 4	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `Device#show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

MSDP のモニタリングおよびメンテナンス

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

表 *1 : MSDP* のモニタおよびメンテナンスのためのコマンド

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

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

表 *2 : MSDP* 接続、統計情報、または *SA* キャッシュ エントリをクリアするためのコマンド

コマンド	目的
clear ip msdp peer *peer-address* | *name*	指定された MSDP ピアへの TCP 接続をクリアし、すべての MSDP メッセージ カウンタをリセットします。
clear ip msdp statistics [*peer-address* | *name*]	セッションをリセットせずに、1 つまたはすべての MSDP ピア統計情報カウンタをクリアします。
clear ip msdp sa-cache [*group-address* | *name*]	すべてのエントリの SA キャッシュ エントリ、特定のグループのすべての送信元、または特定の送信元とグループのペアのすべてのエントリをクリアします。

MSDP の設定例

デフォルト MSDP ピアの設定：例

次に、ルータ A およびルータ C の部分的な設定の例を示します。これらの ISP にはそれぞれに複数のカスタマー（カスタマーと同様）がおり、デフォルトのピアリング（BGP または MBGP なし）を使用しています。この場合、両方の ISP で類似した設定となります。つまり、

両方の ISP では、対応するプレフィックス リストで SA が許可されている場合、デフォルト ピアからの SA だけが受信されます。

ルータ A

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

ルータ C

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

SA ステートのキャッシング：例

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

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

MSDP ピアからの送信元情報の要求：例

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

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

スイッチから発信される送信元情報の制御：例

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

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

スイッチから転送される送信元情報の制御：例

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

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

スイッチで受信される送信元情報の制御：例

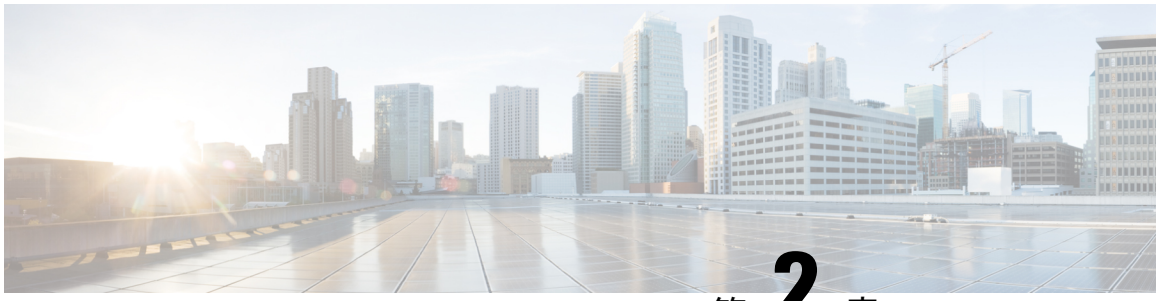
次に、*switch.cisco.com* という名前のピアからのすべての SA メッセージをフィルタリングする例を示します。

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

Multicast Source Discovery Protocol の機能情報

表 *3 : Multicast Source Discovery Protocol* の機能情報

リリース	機能情報
Cisco IOS XE Everest 16.6.1	この機能が導入されました



第 2 章

IP ユニキャスト ルーティングの設定

- IP ユニキャスト ルーティングの設定に関する情報 （31 ページ）
- IP ルーティングに関する情報 （31 ページ）
- IP ルーティングの設定方法 （37 ページ）
- IP アドレッシングの設定方法 （38 ページ）
- IP アドレスのモニタリングおよびメンテナンス （58 ページ）
- IP ユニキャスト ルーティングの設定方法 （59 ページ）
- IP ネットワークのモニタリングおよびメンテナンス （61 ページ）
- IP ユニキャスト ルーティングの機能情報 （61 ページ）

IP ユニキャスト ルーティングの設定に関する情報

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

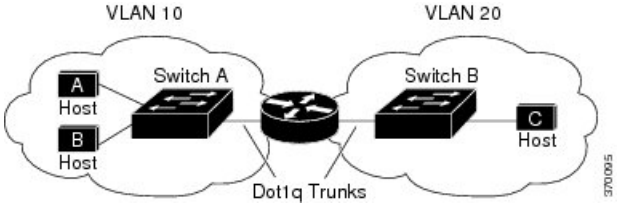
（注） IPv4 トラフィックに加えて、I6（IPv6）ユニキャストルーティングをイネーブルにし、IPv6 トラフィックを転送するようにインターフェイスを設定できます。

IP ルーティングに関する情報

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

図 *2:* ルーティング トポロジの例

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



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

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

ルーティング タイプ

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

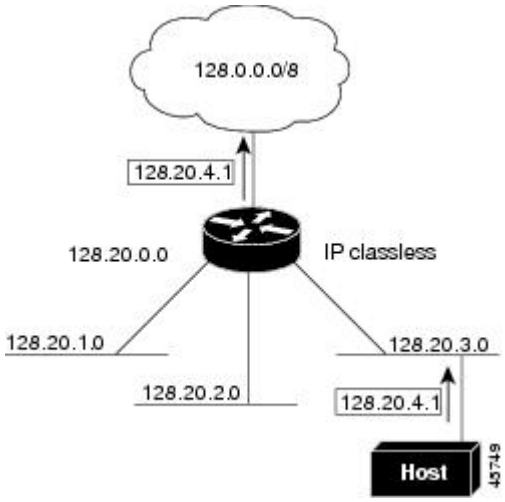
- デフォルト ルーティング
- 事前にプログラミングされているトラフィックのスタティック ルートの使用

クラスレス ルーティング

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

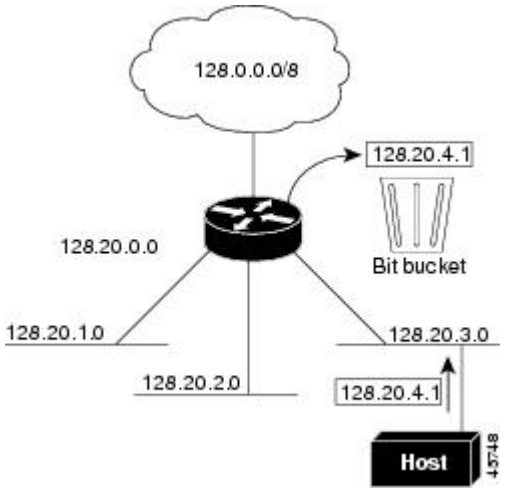
図では、クラスレスルーティングがイネーブルとなっています。ホストがパケットを 128.20.4.1 に送信すると、ルータはパケットを廃棄せずに、最適なスーパーネット ルートに転送します。クラスレス ルーティングがディセーブルの場合、デフォルト ルートがないネットワークのサブネット宛てパケットを受信したルータは、パケットを廃棄します。

図 *3 : IP* クラスレス ルーティングがイネーブルの場合



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

図 *4 : IP* クラスレス ルーティングがディセーブルの場合



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

アドレス解決

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

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

デバイスでは、次の形式のアドレス解決を行うことができます。

- ARP：IP アドレスを MAC アドレスと関連付けるために使用されます。ARP は IP アドレスを入力と解釈し、対応する MAC アドレスを学習します。次に、IP アドレス/MAC アドレス アソシエーションを ARP キャッシュにストアし、すぐに取り出せるようにします。その後、IPデータグラムがリンク層フレームにカプセル化され、ネットワークを通じて送信されます。イーサネット以外の IEEE 802 ネットワークにおける IP データグラムのカプセル化、および ARP 要求や応答については、サブネットワーク アクセス プロトコル（SNAP）で規定されています。
- プロキシ ARP：ルーティング テーブルを持たないホストで、他のネットワークまたはサブネット上のホストの MAC アドレスを学習できるようにします。デバイス（ルータ）が送信者と異なるインターフェイス上のホストに宛てたARP要求を受信した場合、そのルータに他のインターフェイスを経由してそのホストに至るすべてのルートが格納されていれば、ルータは自身のローカル データ リンク アドレスを示すプロキシARPパケットを生成します。ARP要求を送信したホストはルータにパケットを送信し、ルータはパケットを目的のホストに転送します。

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

プロシキ ARP

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

ICMP Router Discovery Protocol

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

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

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

UDP ブロードキャスト パケットおよびプロトコル

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

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

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

ブロードキャスト パケットの処理

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

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

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

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

IP ブロードキャストのフラッディング

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

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

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

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

- パケットの存続可能時間（TTL）値は 2 以上でなければなりません。

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

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

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

IP ルーティングの設定方法

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

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

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

（注） スイッチは、ユニキャスト ルーテッド トラフィックのトンネル インターフェイスをサポートしません。

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

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

ルーティングを設定するための主な手順は次のとおりです。

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

IP アドレッシングの設定方法

IP ルーティングを設定するには、レイヤ 3 ネットワーク インターフェイスに IP アドレスを割り当ててインターフェイスをイネーブルにし、IP を使用するインターフェイスを経由してホストとの通信を許可する必要があります。次の項では、さまざまな IP アドレス指定機能の設定方法について説明します。IP アドレスをインターフェイスに割り当てる手順は必須ですが、その他の手順は任意です。

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

IP アドレス指定のデフォルト設定

表 *4:* アドレス指定のデフォルト設定

機能	デフォルト設定
IP アドレス	未定義

機能	デフォルト設定
ARP	ARP キャッシュに永続的なエントリはありません カプセル化：標準イーサネット形式の ARP 14400 秒（4 時間）
IP ブロードキャスト アドレス	255.255.255.255（すべて 1）
IP クラスレス ルーティング	イネーブル
IP デフォルト ゲートウェイ	ディセーブル
IP ダイレクト ブロードキャスト	ディセーブル（すべての IP ダイレクト ブロードキャストがドロップされます）
IP ドメイン	ドメイン リスト：ドメイン名は未定義 ドメイン検索：イネーブル ドメイン名：イネーブル
IP 転送プロトコル	ヘルパー アドレスが定義されているか、またはユーザ データグラム プロトコル（UDP）フラッディングが設定されている場合、デフォルト ポートでは UDP 転送がイネーブルとなります ローカル ブロードキャスト：ディセーブル スパニングツリー プロトコル（STP）：ディセーブル ターボフラッディング：ディセーブル
IP ヘルパー アドレス	ディセーブル
IP ホスト	ディセーブル

機能	デフォルト設定
ICMP Router Discovery Protocol（IRDP）	ディセーブル イネーブルの場合のデフォルト： • ブロードキャスト IRDP アドバタイズメント • アドバタイズメント間の最大インターバル：600 秒 • アドバタイズ間の最小インターバル：最大インターバルの 0.75 倍 • プリファレンス：0
IP プロキシ ARP	イネーブル
IP ルーティング	ディセーブル
IP サブネットゼロ	ディセーブル

ネットワーク インターフェイスへの IP アドレスの割り当て

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

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

手順

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

	コマンドまたはアクション	目的
ステップ 3	**interface** *interface-id* 例： `Device(config)#interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 4	**no switchport** 例： `Device(config-if)#no switchport`	レイヤ 2 コンフィギュレーション モードからインターフェイスを削除します（物理インターフェイスの場合）。
ステップ 5	**ip address** *ip-address subnet-mask* 例： `Device(config-if)#ip address 10.1.5.1 255.255.255.0`	IP アドレスおよび IP サブネット マスクを設定します。
ステップ 6	**no shutdown** 例： `Device(config-if)#no shutdown`	物理インターフェイスをイネーブルにします。
ステップ 7	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 8	**show ip route** 例： `Device#show ip route`	入力を確認します。
ステップ 9	**show ip interface** [*interface-id*] 例： `Device#show ip interface gigabitethernet 1/0/1`	入力を確認します。
ステップ 10	**show running-config** 例： `Device#show running-config`	入力を確認します。
ステップ 11	**copy running-config startup-config** 例：	（任意）コンフィギュレーション ファイルに設定を保存します。

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

サブネット ゼロの使用

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

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

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip subnet-zero** 例： `Device(config)#ip subnet-zero`	インターフェイス アドレスおよびルーティングのアップデート時にサブネット ゼロの使用をイネーブルにします。
ステップ 4	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例：	入力を確認します。

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

クラスレス ルーティングのディセーブル化

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**no ip classless** 例： `Device(config)#no ip classless`	クラスレスルーティング動作をディセーブルにします。
ステップ 4	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `Device#show running-config`	入力を確認します。

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

アドレス解決方法の設定

アドレス解決を設定するために必要な作業は次のとおりです。

スタティック ARP キャッシュの定義

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**arp** *ip-address hardware-address type* 例： `Device(config)#ip 10.1.5.1 c2f3.220a.12f4 arpa`	ARP キャッシュ内で IP アドレスを MAC（ハードウェア）アドレスに関連付け、次に示すカプセル化タイプのいずれかを指定します。 • **arpa**：ARP カプセル化（イーサネット インターフェイス用）

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

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

ARP のカプセル化の設定

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

カプセル化タイプを無効にするには、**no arp arpa** または **no arp snap** インターフェイス コンフィギュレーション コマンドを使用します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config)#interface gigabitethernet 1/0/2`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 4	**arp** {**arpa** | **snap**} 例： `Device(config-if)#arp arpa`	ARP カプセル化方法を指定します。 • **arpa**：Address Resolution Protocol • **snap**：Subnetwork Address Protocol
ステップ 5	**end** 例：	特権 EXEC モードに戻ります。

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

プロキシ ARP のイネーブル化

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

手順

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

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

IP ルーティングがディセーブルの場合のルーティング支援機能

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

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

プロシキ ARP

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

デフォルト ゲートウェイ

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip default-gateway** *ip-address* 例： `Device(config)#ip default gateway 10.1.5.1`	デフォルト ゲートウェイ（ルータ）を設定します。
ステップ 4	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 5	**show ip redirects** 例： `Device#show ip redirects`	設定を確認するため、デフォルト ゲートウェイ ルータのアドレスを表示します。
ステップ 6	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ICMP Router Discovery Protocol（IRDP）

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

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

手順

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

	コマンドまたはアクション	目的
	`Device(config-if)#ip irdp holdtime 1000`	**maxadvertinterval** 値よりも大きな値（9000秒以下）を指定する必要があります。**maxadvertinterval**値を変更すると、この値も変更されます。
ステップ 7	**ip irdp maxadvertinterval** *seconds* 例： `Device(config-if)#ip irdp maxadvertinterval 650`	（任意）アドバタイズメントの IRDP 最大間隔を設定します。デフォルトは 600 秒です。
ステップ 8	**ip irdp minadvertinterval** *seconds* 例： `Device(config-if)#ip irdp minadvertinterval 500`	（任意）アドバタイズ間の IRDP の最小インターバルを設定します。デフォルト値は **maxadvertinterval** 値の 0.75 倍です。**maxadvertinterval** を変更すると、この値も新しいデフォルト値（**maxadvertinterval** の 0.75 倍）に変更されます。
ステップ 9	**ip irdp preference** *number* 例： `Device(config-if)#ip irdp preference 2`	（任意）デバイスの IRDP プリファレンス レベルを設定します。指定できる範囲は -231 ～ 231 です。デフォルトは 0 です。大きな値を設定すると、ルータのプリファレンス レベルも高くなります。
ステップ 10	**ip irdp address** *address* [*number*] 例： `Device(config-if)#ip irdp address 10.1.10.10`	（任意）プロキシアドバタイズを行うための IRDP アドレスとプリファレンスを設定します。
ステップ 11	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 12	**show ip irdp** 例： `Device#show ip irdp`	IRDP 値を表示し、設定を確認します。
ステップ 13	**copy running-config startup-config** 例：	（任意）コンフィギュレーション ファイルに設定を保存します。

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

ブロードキャスト パケットの処理方法の設定

これらの方式をイネーブルにするには、次に示す作業を実行します。

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

ダイレクト ブロードキャストから物理ブロードキャストへの変換のイネーブル化

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

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

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

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

手順

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

	コマンドまたはアクション	目的
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config)#interface gigabitethernet 1/0/2`	インターフェイス コンフィギュレーション モードを開始し、設定するインターフェイスを指定します。
ステップ 4	**ip network-broadcast** 例： `Device(config-if)#ip network-broadcast`	入力インターフェイスがネットワーク プレフィックスダイレクト ブロードキャストパケットを受信して受け入れることを可能にします。
ステップ 5	**exit** 例： `Device(config-if)#exit`	グローバル コンフィギュレーション モードに戻ります。
ステップ 6	**interface** *interface-id* 例： `Device(config)#interface gigabitethernet 1/0/3`	インターフェイス コンフィギュレーション モードを開始し、設定するインターフェイスを指定します。
ステップ 7	**ip directed-broadcast** [*access-list-number*] 例： `Device(config-if)#ip directed-broadcast 103`	インターフェイス上で、ダイレクトブロードキャストから物理ブロードキャストへの変換をイネーブルにします。転送するブロードキャストを制御するアクセス リストを指定できます。アクセス リストを指定すると、アクセス リストで許可されている IP パケットだけが変換可能になります。
ステップ 8	**exit** 例： `Device(config-if)#exit`	グローバル コンフィギュレーション モードに戻ります。
ステップ 9	**ip forward-protocol** {**udp** [*port*] | **nd** | **sdns**} 例：	ブロードキャストパケットを転送するとき、ルータによって転送されるプロトコルおよびポートを指定します。

	コマンドまたはアクション	目的
	`Device(config)#ip forward-protocol nd`	• **udp**：UPD データグラムを転送します。 port：（任意）転送される UDP サービスを制御する宛先ポートです。 • **nd**：ND データグラムを転送します。 • **sdns**：SDNS データグラムを転送します。
ステップ 10	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 11	**show ip interface** [*interface-id*] 例： `Device#show ip interface`	指定されたインターフェイスまたはすべてのインターフェイスの設定を確認します。
ステップ 12	**show running-config** 例： `Device#show running-config`	入力を確認します。
ステップ 13	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーションファイルに設定を保存します。

UDP ブロードキャスト パケットおよびプロトコルの転送

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

手順

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

	コマンドまたはアクション	目的
ステップ 8	**show ip interface** [*interface-id*] 例： `Device#show ip interface gigabitethernet 1/0/1`	指定されたインターフェイスまたはすべてのインターフェイスの設定を確認します。
ステップ 9	**show running-config** 例： `Device#show running-config`	入力を確認します。
ステップ 10	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーションファイルに設定を保存します。

IP ブロードキャスト アドレスの確立

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

手順

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

	コマンドまたはアクション	目的
ステップ 4	**ip broadcast-address** *ip-address* 例： `Device(config-if)#ip broadcast-address 128.1.255.255`	デフォルト値と異なるブロードキャスト アドレス（128.1.255.255 など）を入力します。
ステップ 5	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 6	**show ip interface** [*interface-id*] 例： `Device#show ip interface`	指定されたインターフェイスまたはすべてのインターフェイスのブロードキャスト アドレスを確認します。
ステップ 7	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

IP ブロードキャストのフラッディング

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip forward-protocol spanning-tree** 例： `Device(config)#ip forward-protocol spanning-tree`	ブリッジング スパニングツリー データベースを使用し、UDP データグラムをフラッディングします。

	コマンドまたはアクション	目的
ステップ 4	**ip forward-protocol turbo-flood** 例： `Device(config)#ip forward-protocol turbo-flood`	スパニングツリー データベースを使用し、UDP データグラムのフラッディングを高速化します。
ステップ 5	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 6	**show running-config** 例： `Device#show running-config`	入力を確認します。
ステップ 7	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

IP アドレスのモニタリングおよびメンテナンス

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

表 *5:* キャッシュ、テーブル、データベースをクリアするコマンド

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

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

表 *6:* キャッシュ、テーブル、データベースを表示するコマンド

コマンド	目的
show arp	ARP テーブル内のエントリを表示します。
show hosts	デフォルトのドメイン名、検索サービスの方式、サーバ ホスト名、およびキャッシュに格納されているホスト名とアドレスのリストを表示します。
show ip aliases	TCP ポートにマッピングされた IP アドレスを表示します（エイリアス）。
show ip arp	IP ARP キャッシュを表示します。
show ip interface [*interface-id*]	インターフェイスの IP ステータスを表示します。
show ip irdp	IRDP 値を表示します。
show ip masks *address*	ネットワーク アドレスに対して使用されるマスクおよび各マスクを使用するサブネット番号を表示します。
show ip redirects	デフォルト ゲートウェイのアドレスを表示します。
show ip route [*address* [*mask*]] | [*protocol*]	ルーティング テーブルの現在の状態を表示します。
show ip route summary	サマリー形式でルーティング テーブルの現在のステータスを表示します。

IP ユニキャスト ルーティングの設定方法

IP ユニキャスト ルーティングのイネーブル化

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip routing** 例： `Device(config)# ip routing`	IPルーティングをイネーブルにします。
ステップ 4	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `Device#show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

IP ルーティングの有効化の例

次に、IP ルーティングをイネーブルにする例を示します。

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

次の作業

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

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

IP ネットワークのモニタリングおよびメンテナンス

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

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

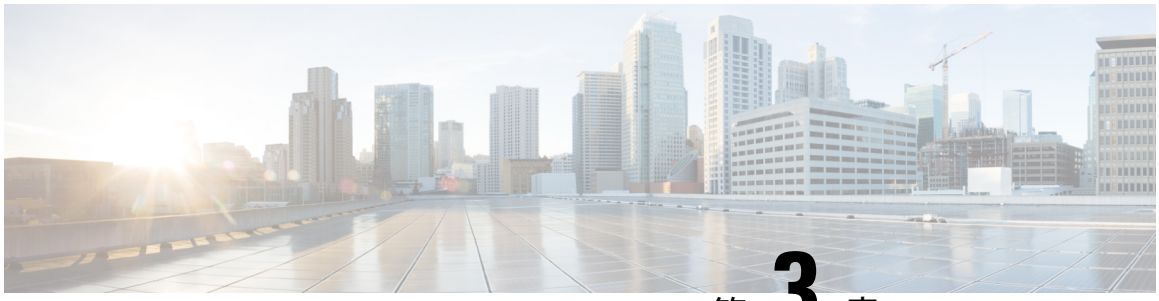
コマンド	目的
show ip route summary	サマリー形式でルーティング テーブルの現在のステータスを表示します。

IP ユニキャスト ルーティングの機能情報

表 *8 : IP* ユニキャスト ルーティングの機能情報

リリース	機能	機能情報
Cisco IOS XE Everest 16.6.1	IP ユニキャスト ルーティング	IP ユニキャストルーティングは、トラフィックをユニキャストアドレスに転送するルーティングプロセスです。ルータとレイヤ 3 スイッチは、事前にプログラムされたスタティックルートまたはデフォルトルートのいずれかを介してパケットをルーティングします。

リリース	機能	機能情報
Cisco IOS XE Amsterdam 17.3.1	新しいコマンドの **ip network-broadcast**	**ip network-broadcast** コマンドは、ネットワークプレフィックス ダイレクト ブロードキャスト パケットを受信して受け入れるために導入されました。



第 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 （63 ページ）
- IPv6 ユニキャストルーティングの設定方法 （68 ページ）
- IPv6 ユニキャスト ルーティングの設定例 （84 ページ）
- その他の参考資料 （87 ページ）
- 機能情報 （87 ページ）

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。

IPv6 の概要

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

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

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

IPv6 およびこの章のその他の機能については、

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

IPv6 のスタティック ルート

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

IPv6 のスタティック ルーティングの設定（CLI）

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

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

IPv6 ユニキャストのパス MTU ディスカバリ

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

ICMPv6

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

ネイバー探索

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

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

ネイバー探索スロットリングにより、IPv6 パケットをルーティングするためにネクスト ホップ転送情報を取得するプロセス中に、スイッチ CPU に不必要な負荷がかかりません。IPv6 パケットのネクストホップがスイッチによってアクティブに解決しようとしている同じネイバーである場合は、そのようなパケットが追加されると、スイッチはそのパケットをドロップします。このドロップにより、CPU に余分な負荷がかからないようになります。

DNS 設定の IPv6 ルータ アドバタイズメント オプション

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

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

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

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

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

デフォルト ルータ プリファレンス

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

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

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

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

IPv6 のポリシーベース ルーティング

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

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

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

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

PBR を使用すると、次の作業を実行できます。

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

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

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

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

サポートされていない IPv6 ユニキャスト ルーティング機能

スイッチは、次の IPv6 機能をサポートしません。

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

IPv6 機能の制限

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

機能の制限は次のとおりです。

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

IPv6 とスイッチ スタック

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

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

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

IPv6 アクティブスイッチおよびメンバーの機能は次のとおりです。

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

- ハードウェアへのルートのプログラミング

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

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

IPv6 のデフォルト設定

表 *9 : IPv6* のデフォルト設定

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

IPv6 ユニキャストルーティングの設定方法

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

IPv6 アドレッシングの設定と IPv6 ルーティングのイネーブル化

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

スイッチ上の IPv6 を設定する前に、次の注意事項に従ってください。

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

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

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

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

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

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： デバイス> **enable**	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： デバイス# **configure terminal**	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**sdm prefer dual-ipv4-and-ipv6** {**advanced** | **vlan**} 例： デバイス(config)# **sdm prefer dual-ipv4-and-ipv6 vlan**	IPv4 および IPv6 をサポートする SDM テンプレートを選択します。 • **advanced**：スイッチをアドバンスドテンプレートに設定します。 • **vlan**：ハードウェアでのルーティングをサポートしないスイッチでの VLAN 設定を最適化します。
ステップ 4	**end** 例： デバイス(config)# **end**	特権 EXEC モードに戻ります。
ステップ 5	**reload** 例： デバイス# **reload**	オペレーティング システムをリロードします。
ステップ 6	**configure terminal** 例： デバイス# **configure terminal**	スイッチのリロード後、グローバル コンフィギュレーション モードを開始します。
ステップ 7	**interface** *interface-id* 例： デバイス(config)# **interface gigabitethernet 1/0/1**	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。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

	コマンドまたはアクション	目的
		（SVI）、またはレイヤ 3 EtherChannel に設定できます。
ステップ 8	**no switchport** 例： `デバイス(config-if)# no switchport`	レイヤ 2 コンフィギュレーション モードからインターフェイスを削除します（物理インターフェイスの場合）。

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

ステップ 10	**exit** 例： `デバイス(config-if)# exit`	グローバル コンフィギュレーション モードに戻ります。

	コマンドまたはアクション	目的
ステップ 11	**ip routing** 例： `デバイス(config)# ip routing`	スイッチ上でIPルーティングをイネーブルにします。
ステップ 12	**ipv6 unicast-routing** 例： `デバイス(config)# ipv6 unicast-routing`	IPv6 ユニキャスト データ パケットの転送をイネーブルにします。
ステップ 13	**end** 例： `デバイス(config)# end`	特権 EXEC モードに戻ります。
ステップ 14	**show ipv6 interface** *interface-id* 例： `デバイス# show ipv6 interface gigabitethernet 1/0/1`	入力を確認します。
ステップ 15	**copy running-config startup-config** 例： `デバイス# copy running-config startup-config`	（任意）コンフィギュレーションファイルに設定を保存します。

IPv4 および IPv6 プロトコル スタックの設定

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

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

手順

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

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

再帰 DNS サーバ（RDNSS）の設定

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

始める前に

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface ethernet** *number* 例： `Device(config)# interface ethernet 3/3`	インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**ipv6 nd ra dns server** *ipv6-addr* [*rdnss-life* | **infinite**] **sequence** *sequence-num* 例： `Device(config-if)# ipv6 nd ra dns server 1::1 1000 sequence 0`	再帰 DNS サーバを設定します。サーバの有効期間と順序を指定できます。
ステップ 5	**show ipv6 nd ra dns server** [**interface** *interface*] 例： `Device(config-if)# show ipv6 nd ra dns server`	（任意）設定した RDNSS リストを表示します。
ステップ 6	**ipv6 nd ra dns server suppress** 例： `Device(config-if)# ipv6 nd ra dns server suppress`	（任意）設定したサーバ リストをディセーブルにします。

デフォルト ルータ プリファレンス（DRP）の設定

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

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

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

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

手順

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

IPv6 ICMP レート制限の設定

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

ICMP のレート制限パラメータを変更するには、次の手順を実行します。

手順

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

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

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

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

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

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

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

IPv6 のスタティック ルーティングの設定

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

スタティック IPv6 ルーティングを設定するには、次の手順を実行します。

始める前に

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

手順

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

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

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

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

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

インターフェイスでの IPv6 PBR の有効化

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

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

PBR for IPv6 を有効にするには、次の手順を実行します。

手順

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

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

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

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

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

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

ローカル PBR for IPv6 の有効化

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

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

手順

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

IPv6 の表示

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

表 *10 : IPv6* をモニタリングするコマンド

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

IPv6 ユニキャスト ルーティングの設定例

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

例：IPv4 および IPv6 プロトコル スタックの設定

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

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

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

例：RDNSS の設定

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

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

Device(config)# show ipv6 nd ra dns server

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

例：DNSSL の設定

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

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

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

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

例：デフォルト ルータ プリファレンスの設定

次に、インターフェイス上のルータに高い DRP を設定する例を示します。

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

例：IPv6 ICMP レート制限の設定

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

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

例：IPv6 のスタティック ルーティングの設定

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

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

例：インターフェイスでの PBR のイネーブル化

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

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

例：ローカル PBR for IPv6 の有効化

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

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

```
Device(config)# set ipv6 next-hop 2001:DB8:2003:1::95
Device(config)# ipv6 local policy route-map pbr-src-90
```

例：IPv6 の表示

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

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

その他の参考資料

標準および RFC

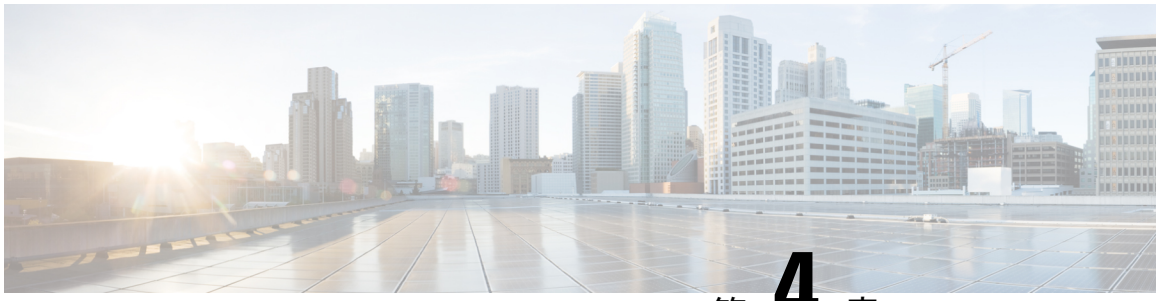
標準/RFC	タイトル
RFC 5453	予約済み *IPv6* インターフェイス識別子

機能情報

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

表 *11 : IPv6* ユニキャストおよびルーティングの機能情報

機能名	リリース	機能情報
IPv6 ユニキャストおよびルーティング	Cisco IOS XE Everest 16.6.1	ユニキャストおよびルーティング機能が IPv6 に対してサポートされました。
RFC 5453	Cisco IOS XE Gibraltar 16.11.1	RFC 5453 がサポートされています。
DNS 設定の IPv6 ルータ アドバタイズメント オプション	Cisco IOS XE Gibraltar 16.11.1	この機能が導入されました。



第 4 章

RIP の設定

- RIP 情報 （89 ページ）
- RIP の設定方法 （90 ページ）
- 例：IPv6 用の RIP の設定 （101 ページ）
- 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 （101 ページ）
- Routing Information Protocol に関する機能情報 （102 ページ）

RIP 情報

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

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

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

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

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

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

RIP for IPv6

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

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

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

サマリー アドレスおよびスプリット ホライズン

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

RIP の設定方法

RIP のデフォルト設定

表 *12 : RIP* のデフォルト設定

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

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

基本的な RIP パラメータの設定

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

手順

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

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

	コマンドまたはアクション	目的
		• *invalid*：ルートが無効と宣言されるまでの時間。デフォルト値は180秒です。 • *holddown*：ルートがルーティングテーブルから削除されるまでの時間。デフォルト値は 180 秒です。 • *flush*：ルーティング アップデートが延期される時間。デフォルトは 240 秒です。
ステップ 9	**version** {**1** | **2**} 例： `Device(config-router)# version 2`	（任意）RIP バージョン 1 または RIP バージョン 2 のパケットだけを送受信するようにスイッチを設定します。デフォルトの場合、スイッチではバージョン 1 および 2 を受信しますが、バージョン 1 だけを送信します。インターフェイスコマンド **ip rip** {**send** | **receive**} **version 1** | **2** | **1 2**} を使用し、インターフェイスでの送受信に使用するバージョンを制御することもできます。
ステップ 10	**no auto summary** 例： `Device(config-router)# no auto summary`	（任意）自動要約をディセーブルにします。デフォルトでは、クラスフル ネットワーク境界を通過するときにサブプレフィックスがサマライズされます。サマライズをディセーブルにし（RIP バージョン 2 だけ）、クラスフル ネットワーク境界にサブネットおよびホストルーティング情報をアドバタイズします。
ステップ 11	**output-delay** *delay* 例： `Device(config-router)# output-delay 8`	（任意）送信する RIP アップデートにパケット間遅延を追加します。デフォルトでは、複数のパケットからなる RIP アップデートのパケットに、パケット間遅延が追加されません。パケットを低速なデバイスに送信する場合は、8 ～ 50 ミリ秒のパケット間遅延を追加できます。
ステップ 12	**end** 例：	特権 EXEC モードに戻ります。

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

RIP 認証の設定

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

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

手順

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

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

IPv6 RIP の設定

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

IPv6 の RIP ルーティングを設定するには、次の手順を実行します。

始める前に

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

手順

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

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

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

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

サマリー アドレスおよびスプリット ホライズンの設定

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

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

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

手順

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

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

スプリット ホライズンの設定

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

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

手順

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

	コマンドまたはアクション	目的
	startup-config	

例：IPv6 用の RIP の設定

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

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

サマリーアドレスおよびスプリットホライズンの設定例

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

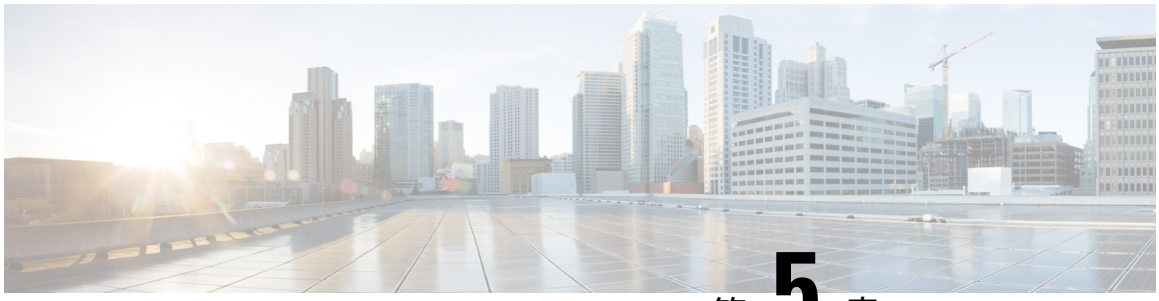
（注） スプリットホライズンが有効である場合、（**ip summary-address rip** ルータ コンフィギュレーション コマンドによって設定される）自動サマリーとインターフェイス サマリー アドレスはともにアドバタイズされません。

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

Routing Information Protocol に関する機能情報

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

リリース	機能情報
Cisco IOS XE Everest 16.6.1	この機能が導入されました。



第 5 章

OSPF の設定

- OSPF に関する情報 （103 ページ）
- OSPF の設定方法 （107 ページ）
- OSPF のモニタリング （122 ページ）
- OSPF の設定例 （123 ページ）
- OSPF の設定例 （123 ページ）
- 例：基本的な OSPF パラメータの設定 （123 ページ）
- OSPF の機能情報 （123 ページ）

OSPF に関する情報

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

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

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

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

『OSPF for IPv6』

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

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

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

OSPF NSF

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

- OSPF NSF 認識 （104 ページ）
- OSPF NSF 対応 （104 ページ）

OSPF NSF 認識

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

この機能をディセーブルにできません。

OSPF NSF 対応

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

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

OSPF エリア パラメータ

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

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

その他の OSPF パラメータ

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

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

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

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

- すべての OSPF **show** 特権 EXEC コマンドでの表示にドメインネームサーバ（DNS）名を使用すると、ルータ ID やネイバー ID を指定して表示する場合に比べ、ルータを簡単に特定できます。

- デフォルト メトリック：OSPF は、インターフェイスの帯域幅に従ってインターフェイスの OSPF メトリックを計算します。メトリックは、帯域幅で分割された *ref-bw* として計算されます。ここでの *ref* のデフォルト値は 10 で、帯域幅（*bw*）は **bandwidth** インターフェイス コンフィギュレーション コマンドによって指定されます。大きな帯域幅を持つ複数のリンクの場合は、大きな数値を指定し、これらのリンクのコストを区別できます。

- アドミニストレーティブディスタンスは、ルーティング情報送信元の信頼性を表す数値です。0 ～ 255 の整数を指定でき、値が大きいほど信頼性は低下します。アドミニストレーティブディスタンスが 255 の場合はルーティング情報の送信元をまったく信頼できないた

め、無視する必要があります。OSPF では、エリア内のルート（エリア内）、別のエリアへのルート（エリア間）、および再配信によって学習した別のルーティング ドメインからのルート（外部）の 3 つの異なるアドミニストレーティブ ディスタンスが使用されます。どのアドミニストレーティブ ディスタンスの値でも変更できます。

- 受動インターフェイス：イーサネット上の 2 つのデバイス間のインターフェイスは 1 つのネットワーク セグメントしか表しません。このため、OSPF が送信側インターフェイスに hello パケットを送信しないようにするには、送信側デバイスを受動インターフェイスに設定する必要があります。両方のデバイスは受信側インターフェイス宛ての hello パケットを使用することで、相互の識別を可能にします。
- ルート計算タイマー：OSPF がトポロジ変更を受信してから SPF 計算を開始するまでの遅延時間、および 2 つの SPF 計算の間のホールド タイムを設定できます。
- ネイバー変更ログ：OSPF ネイバー ステートが変更されたときに Syslog メッセージを送信するようにルータを設定し、ルータの変更を詳細に表示できます。

LSA グループ ペーシング

OSPF LSA グループ ペーシング機能を使用すると、OSPF LSA をグループ化し、リフレッシュ、チェックサム、エージング機能の同期を取って、ルータをより効率的に使用できるようになります。デフォルトでこの機能はイネーブルとなっています。デフォルトのペーシング インターバルは 4 分間です。通常は、このパラメータを変更する必要はありません。最適なグループ ペーシング インターバルは、ルータがリフレッシュ、チェックサム、エージングを行う LSA 数に反比例します。たとえば、データベース内に約 10000 個の LSA が格納されている場合は、ペーシング インターバルを短くすると便利です。小さなデータベース（40 ～ 100 LSA）を使用する場合は、ペーシング インターバルを長くし、10 ～ 20 分に設定してください。

ループバック インターフェイス

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

OSPF の設定方法

OSPF のデフォルト設定

表 *14 : OSPF* のデフォルト設定

機能	デフォルト設定
インターフェイス パラメータ	コスト： 再送信インターバル：5 秒 送信遅延：1 秒 プライオリティ：1 hello インターバル：10 秒 デッド インターバル：hello インターバルの 4 倍 認証なし パスワードの指定なし MD5 認証はディセーブル
エリア	認証タイプ：0（認証なし） デフォルト コスト：1 範囲：ディセーブル スタブ：スタブ エリアは未定義 NSSA：NSSA エリアは未定義
自動コスト	100 Mb/s
デフォルト情報送信元	ディセーブルイネーブルの場合、デフォルトのメトリック設定は10で、外部ルートタイプのデフォルトはタイプ 2 です。
デフォルト メトリック	各ルーティング プロトコルに適切な、組み込みの自動メトリック変換
距離 OSPF	dist1（エリア内のすべてのルート）：110。 dist2（エリア間のすべてのルート）：110。および dist3（他のルーティング ドメインからのルート）：110。

機能	デフォルト設定
OSPF データベース フィルタ	ディセーブルすべての発信 LSA がインターフェイスにフラッディングされます。
IP OSPF 名検索	ディセーブル
隣接関係変更ログ	イネーブル
ネイバー	指定なし
ネイバー データベース フィルタ	ディセーブルすべての発信 LSA はネイバーにフラッディングされます。
ネットワーク エリア	ディセーブル
ルータ ID	OSPF ルーティング プロセスは未定義
サマリー アドレス	ディセーブル
タイマー LSA グループのペーシング	240 秒
タイマー Shortest Path First（SPF）	spf 遅延：50 ミリ秒、spf ホールド時間：200 ミリ秒
仮想リンク	エリア ID またはルータ ID は未定義 hello インターバル：10 秒 再送信インターバル：5 秒 送信遅延：1 秒 デッド インターバル：40 秒 認証キー：キーは未定義 メッセージ ダイジェスト キー（MD5）：キーは未定義

基本的な OSPF パラメータの設定

OSPF をイネーブルにするには、OSPF ルーティング プロセスを作成し、そのルーティング プロセスに関連付けられる IP アドレスの範囲を指定し、その範囲に関連付けられるエリア ID を割り当てます。

手順

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

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

IPv6 OSPF の設定

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

IPv6 の OSPF ルーティングを設定するには、次の手順を実行します。

始める前に

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

次の注意事項に従ってください。

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

手順

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

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

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

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

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

OSPF インターフェイスの設定

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

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

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** interface-id 例： `Device(config)#interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 4	**ip ospf** cost 例： `Device(config-if)#ip ospf 8`	（任意）インターフェイスでパケットを送信するコストを明示的に指定します。
ステップ 5	**ip ospf retransmit-interval** seconds 例： `Device(config-if)#ip ospf transmit-interval 10`	（任意）LSA 送信間隔を秒数で指定します。指定できる範囲は 1 ～ 65535 秒です。デフォルト値は 5 秒です。
ステップ 6	**ip ospf transmit-delay** seconds 例： `Device(config-if)#ip ospf transmit-delay 2`	（任意）リンク ステートアップデート パケットを送信するまでの予測待機時間を秒数で設定します。指定できる範囲は 1 ～ 65535 秒です。デフォルト値は 1 秒です。

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

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

OSPF エリア パラメータの設定

始める前に

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

手順

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

	コマンドまたはアクション	目的
		アでなく通常のエリアに取り込む場合に使用します。 • **default-information-originate**：LSA タイプ 7 を NSSA に取り込めるようにする場合に、ABR で選択します。 • **no-redistribution**：サマリー LSA を NSSA に送信しない場合に選択します。
ステップ 8	**area** *area-id* **range** *address mask* 例： `Device(config-router)#area 1 range 255.240.0.0`	（任意）単一のルートをアドバタイズするアドレス範囲を指定します。このコマンドは、ABR に対してだけ使用します。
ステップ 9	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 10	**show ip ospf** [*process-id*] 例： `Device#show ip ospf`	設定を確認するため、一般的な OSPF ルーティング プロセスまたは特定のプロセス ID に関する情報を表示します。
ステップ 11	**show ip ospf** [*process-id* [*area-id*]] **database** 例： `Device#show ip osfp database`	特定のルータの OSPF データベースに関連する情報のリストを表示します。
ステップ 12	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

その他の OSPF パラメータの設定

手順

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

	コマンドまたはアクション	目的
ステップ 8	**ip auto-cost reference-bandwidth** *ref-bw* 例： `Device(config)#ip auto-cost reference-bandwidth 5`	（任意）単一のルートをアドバタイズするアドレス範囲を指定します。このコマンドは、ABR に対してだけ使用します。
ステップ 9	**distance ospf** {[**inter-area** *dist1*] [**inter-area** *dist2*] [**external** *dist3*]} 例： `Device(config)#distance ospf inter-area 150`	（任意）OSPF の距離の値を変更します。各タイプのルートのデフォルト距離は 110 です。有効値は 1 ～ 255 です。
ステップ 10	**passive-interface** *type number* 例： `Device(config)#passive-interface gigabitethernet 1/0/6`	（任意）指定されたインターフェイス経由の hello パケットの送信を抑制します。
ステップ 11	**timers throttle spf** *spf-delay spf-holdtime spf-wait* 例： `Device(config)#timers throttle spf 200 100 100`	（任意）ルート計算タイマーを設定します。 • *spf-delay*：SPF 計算の変更を受信する間の遅延。指定できる範囲は 1 ～ 600000 ミリ秒です。 • *spf-holdtime*：最初と 2 番目の SPF 計算の間の遅延。指定できる範囲は 1 ～ 600000 ミリ秒です。 • *spf-wait*：SPF 計算の最大待機時間（ミリ秒）。指定できる範囲は 1 ～ 600000 ミリ秒です。
ステップ 12	**ospf log-adj-changes** 例： `Device(config)#ospf log-adj-changes`	（任意）ネイバーステートが変更されたとき、syslog メッセージを送信します。
ステップ 13	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。

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

LSA グループ ペーシングの変更

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router ospf** *process-id* 例： `Device(config)#router ospf 25`	OSPF ルーティングをイネーブルにし、ルータ コンフィギュレーション モードを開始します。
ステップ 4	**timers lsa-group-pacing** *seconds* 例： `Device(config-router)#timers lsa-group-pacing 15`	LSA の グループ ペーシングを変更します。
ステップ 5	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。

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

ループバック インターフェイスの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface loopback 0** 例： `Device(config)#interface loopback 0`	ループバック インターフェイスを作成し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**ip address** address mask 例： `Device(config-if)#ip address 10.1.1.5 255.255.240.0`	このインターフェイスに IP アドレスを割り当てます。
ステップ 5	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。

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

OSPF のモニタリング

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

表 *15 : IP OSPF* 統計情報の表示コマンド

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

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

OSPF の設定例

OSPF の設定例

例：基本的な OSPF パラメータの設定

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

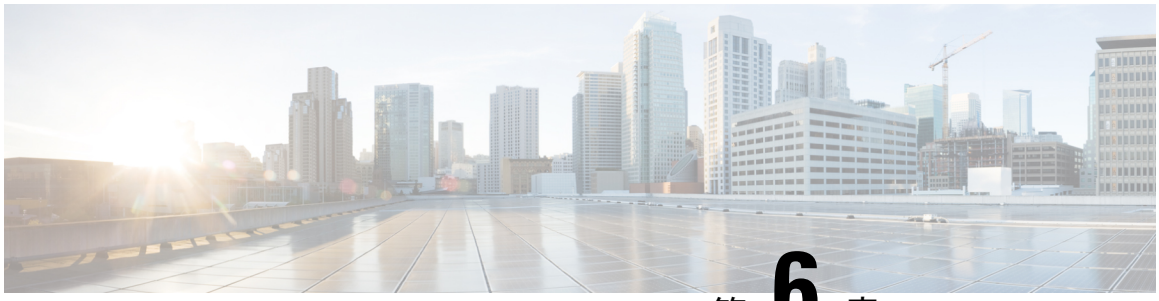
```
Device(config)#router ospf 109
Device(config-router)#network 131.108.0.0 255.255.255.0 area 24
```

OSPF の機能情報

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

表 *16 : OSPF* の機能情報

リリース	機能情報
Cisco IOS XE Everest 16.6.1	この機能が導入されました



第 6 章

OSPF NSR の設定

- OSPF ノンストップルーティングに関する制約事項 （125 ページ）
- OSPF ノンストップルーティングに関する情報 （125 ページ）
- OSPF ノンストップ ルーティングの設定方法 （126 ページ）
- OSPF ノンストップ ルーティングの設定例 （127 ページ）
- OSPF ノンストップルーティングの機能情報 （128 ページ）

OSPF ノンストップルーティングに関する制約事項

- OSPF ノンストップルーティングでは、動作の特定の段階で OSPF に使用されるメモリを大幅に増やすことができます。CPU使用率も増やすことができます。ルータのメモリ容量を認識し、OSPF ノンストップルーティングの考えられるメモリ要件を見積もっておく必要があります。

  詳細については、「OSPF ノンストップルーティングの設定」を参照してください。メモリと CPU が制約を受けるデバイスでは、代わりに OSPF ノンストップフォワーディング（NSF）の使用を検討する場合があります。詳細については、OSPF RFC 3623 グレースフル リスタート ヘルパー モードを参照してください。

- アクティブルートプロセッサ（RP）からスタンバイ RP への切り替えは、ハードウェアプラットフォームによって数秒かかることがあります。この間、OSPF は hello パケットを送信できません。そのため、短い OSPF dead 間隔を使用する設定では切り替えで隣接関係を維持できない可能性があります。

OSPF ノンストップルーティングに関する情報

OSPF ノンストップルーティング機能を使用すると、冗長ルートプロセッサ（RP）を持つデバイスが計画内外の RP の切り替えで Open Shortest Path First（OSPF）ステートと隣接関係を維持することができます。OSPF ステートは、アクティブ RP からスタンバイ RP で OSPF からステート情報のチェックポイントを実行することによって維持されます。スタンバイ RP への切り替え後、OSPF はチェックポイントされた情報を使用して中断することなく動作を継続します。

OSPF ノンストップルーティングは OSPF ノンストップ フォワーディング（NSF）と同様の機能を提供しますが、しくみは異なります。NSF では、新しいアクティブ スタンバイ RP の OSPF にステート情報はありません。OSPF は OSPF プロトコルの拡張を使用して、隣接する OSPF デバイスからステートを回復します。リカバリが機能するためには、ネイバーが NSF プロトコル拡張をサポートし、再起動するデバイスの「ヘルパー」として積極的に動作する必要があります。ネイバーはまた、プロトコル ステートのリカバリが行われる間、再起動するデバイスにデータ トラフィックを転送し続ける必要もあります。

一方、ノンストップルーティングでは、切り替えを実行するデバイスはデバイスステートを内部的に保持し、ほとんどの場合、ネイバーは切り替えを認識しません。隣接デバイスからのサポートが必要ないため、ノンストップルーティングは NSF を使用できない状況で使用できます。たとえば、一部のネイバーが NSF プロトコル拡張を実装していないネットワーク、または NSF を当てにできなくなるリカバリ中にネットワークトポロジを変更するネットワークでは、NSF の代わりにノンストップルーティングを使用します。

OSPF ノンストップ ルーティングの設定方法

ここでは、OSPFノンストップルーティングの設定について説明します。

OSPF ノンストップルーティングの設定

OSPF ノンストップルーティングを設定するには、次の手順を実行します。

（注） ノンストップルーティングをサポートしないデバイスは、**nsr**（OSPFv3）コマンドを受け入れません。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router ospf** *process-id* 例： `Device(config)# router ospf 109`	OSPF 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。

	コマンドまたはアクション	目的
ステップ 4	**nsr** 例： `Device(config-router)# nsr`	ノンストップルーティングを設定します。
ステップ 5	**end** 例： `Device(config-router)# end`	ルータ コンフィギュレーション モードを終了して、特権 EXEC モードに戻ります。
ステップ 6	**show ip ospf** ［*process-id* ］ **nsr** [**objects** | **statistics**］ 例： `Device# show ip ospf 109 nsr`	OSPF ノンストップルーティングのステータス情報を表示します。

OSPF ノンストップ ルーティングの設定例

例：OSPF ノンストップルーティングの設定

次に、OSPF NSR の設定方法を示す出力例を示します。

```
Device> enable
Device# configure terminal
Device(config)# router ospf 1
Device(config-router)# nsr
Device(config-router)# end
Device# show ip ospf 1 nsr
Standby RP
 Operating in duplex mode
 Redundancy state: STANDBY HOT
 Peer redundancy state: ACTIVE
 ISSU negotation complete
 ISSU versions compatible
Routing Process "ospf 1" with ID 10.1.1.100
 NSR configured
 Checkpoint message sequence number: 3290
 Standby synchronization state: synchronized
 Bulk sync operations:  1
 Last sync start time:  15:22:48.971 UTC Fri Jan 14 2011
 Last sync finish time: 15:22:48.971 UTC Fri Jan 14 2011
 Last sync lost time:   -
 Last sync reset time:  -
 LSA Count: 2, Checksum Sum 0x00008AB4
```

出力には、OSPF ノンストップルーティングが設定されていること、スタンバイ RP 上で OSPF が完全に同期されていて、アクティブな RP に障害が発生したり切り替えが手動で実行されても操作を続行する準備ができていることが示されています。

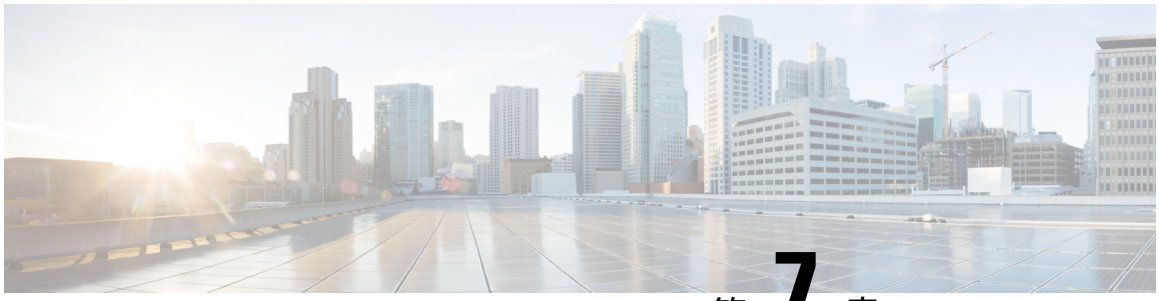
OSPF ノンストップルーティングの機能情報

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

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

リリース	機能	機能情報
Cisco IOS XE Amsterdam 17.3.1	OSPF ノンストップルーティング	OSPF ノンストップルーティング機能を使用すると、冗長ルートプロセッサを持つデバイスが計画内外の RP スイッチオーバーで OSPF ステートと隣接関係を維持することができます。

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



第 7 章

OSPFv3 NSR の設定

- OSPFv3 ノンストップルーティングに関する情報 （129 ページ）
- OSPFv3 ノンストップルーティングの設定方法 （130 ページ）
- OSPFv3 ノンストップルーティングの設定例 （133 ページ）
- トラブルシューティングのヒント （135 ページ）
- その他の参考資料 （136 ページ）
- OSPFv3 ノンストップルーティングの機能情報 （137 ページ）

OSPFv3 ノンストップルーティングに関する情報

OSPFv3 ノンストップルーティング機能を使用すると、冗長ルートプロセッサ（RP）を持つデバイスが計画内外の RP スイッチオーバーで Open Shortest Path First（OSPF）ステートと隣接関係を維持することができます。この機能は、アクティブ RP からスタンバイ RP への OSPFv3 情報をチェックポイントすることによって実現します。切り替えが発生し、スタンバイ RP が新しいアクティブ RP になると、このチェックポイントされた情報を使用して中断することなく動作が継続されます。

OSPFv3 ノンストップルーティングは OSPFv3 グレースフルリスタート機能と同様の機能を提供しますが、異なる方法で動作します。グレースフル リスタートでは、新しいアクティブ スタンバイ RP の OSPFv3 に最初はステート情報がないため、OSPFv3 プロトコルの拡張を使用して隣接する OSPFv3 デバイスからステートを回復します。これを機能させるには、ネイバーがグレースフル リスタート プロトコル拡張をサポートし、再起動するデバイスのヘルパーとして機能できる必要があります。また、このリカバリの実行中、再起動するデバイスへのデータトラフィックの転送を継続する必要があります。

一方、ノンストップルーティングでは、切り替えを実行するデバイスはデバイスステートを内部的に保持し、ほとんどの場合、ネイバーは切り替えが発生したことを認識しません。隣接デバイスからのサポートが必要ないため、ノンストップルーティングはグレースフルリスタートを使用できない状況で使用できます。たとえば、一部のネイバーがグレースフル リスタート プロトコル拡張を実装していないネットワーク、またはリカバリ中にネットワークトポロジを変更するネットワークでは、グレースフルリスタートを当てにすることができません。

（注） ノンストップルーティングを有効にすると、OSPF の応答性と拡張性が低下します。パフォーマンスの低下は、スタンバイ RP にデータをチェックポイントするのに OSPF が CPU とメモリを使用するために発生します。

OSPFv3 ノンストップルーティングの設定方法

ここでは、OSPFv3 を設定する方法と、アドレスファミリの OSPFv3 ノンストップルーティングを有効または無効にする方法について説明します。

OSPFv3 ノンストップルーティングの設定

（注） ノンストップルーティングをサポートしないデバイスは、**nsr**（OSPFv3）コマンドを受け入れません。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router ospfv3** *process-id* 例： `Device(config)# router ospfv3 109`	ルータ コンフィギュレーション モードを開始して、OSPFv3 ルーティングプロセスを設定します。
ステップ 4	**nsr** 例： `Device(config-router)# nsr`	ノンストップルーティングを設定します。
ステップ 5	**end** 例： `Device(config-router)# end`	ルータ コンフィギュレーション モードを終了して、特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
ステップ 6	**show ospfv3** [*process-id*] [*address-family*] **nsr** 例： `Device# show ospfv3 109 nsr`	OSPFv3 ノンストップルーティングのステータス情報を表示します。

アドレスファミリの OSPFv3 ノンストップルーティングの有効化

アドレスファミリの OSPFv3 ノンストップルーティングを有効にするには、次の手順を実行します。

（注） ノンストップルーティングをサポートしないデバイスは、**nsr**（OSPFv3）コマンドを受け入れません。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router ospfv3** *process-id* 例： `Device(config)# router ospfv3 109`	ルータ コンフィギュレーション モードを開始して、OSPFv3 ルーティングプロセスを設定します。
ステップ 4	**address-family** {**ipv4** | **ipv6**} **unicast** [**vrf** *vrf-name*] 例： `Device(config-router)# address-family ipv4 unicast`	OSPFv3 ルータ コンフィギュレーション モードで、IPv4 または IPv6 アドレスファミリ コンフィギュレーション モードを開始します。
ステップ 5	**nsr** 例： `Device(config-router-af)# nsr`	設定済みのアドレスファミリのノンストップルーティングを有効にします。

	コマンドまたはアクション	目的
ステップ 6	**end** 例： `Device(config-router)# end`	ルータ コンフィギュレーション モードを終了して、特権 EXEC モードに戻ります。

アドレスファミリの OSPFv3 ノンストップルーティングの無効化

アドレスファミリの OSPFv3 ノンストップルーティングを無効にするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router ospfv3** *process-id* 例： `Device(config)# router ospfv3 109`	ルータ コンフィギュレーション モードを開始して、OSPFv3 ルーティングプロセスを設定します。
ステップ 4	**address-family** {**ipv4** | **ipv6**} **unicast** [**vrf** *vrf-name*] 例： `Device(config-router)# address-family ipv6 unicast`	OSPFv3 ルータ コンフィギュレーション モードで、IPv4 または IPv6 アドレス ファミリ コンフィギュレーション モードを開始します。
ステップ 5	**nsr** [**disable**] 例： `Device(config-router-af)# nsr disable`	設定済みのアドレスファミリのノンストップルーティングを無効にします。
ステップ 6	**end** 例： `Device(config-router)# end`	ルータ コンフィギュレーション モードを終了して、特権 EXEC モードに戻ります。

OSPFv3 ノンストップルーティングの設定例

例：OSPFv3 ノンストップルーティングの設定

次に、OSPFv3 ノンストップルーティングを設定し、それが有効になっていることを確認する例を示します。

```
Device(config)# router ospfv3 1
Device(config-router)# nsr
Device(config-router)# end
Device# show ospfv3 1
  OSPFv3 1 address-family ipv4
  Router ID 10.0.0.1
  Supports NSSA (compatible with RFC 3101)
  Event-log enabled, Maximum number of events: 1000, Mode: cyclic
  It is an area border and autonomous system boundary router
  Redistributing External Routes from,
  Router is not originating router-LSAs with maximum metric
  Initial SPF schedule delay 5000 msecs
  Minimum hold time between two consecutive SPFs 10000 msecs
  Maximum wait time between two consecutive SPFs 10000 msecs
  Minimum LSA interval 5 secs
  Minimum LSA arrival 1000 msecs
  LSA group pacing timer 240 secs
  Interface flood pacing timer 33 msecs
  Retransmission pacing timer 66 msecs
  Retransmission limit dc 24 non-dc 24
  Number of external LSA 0. Checksum Sum 0x000000
  Number of areas in this router is 3. 2 normal 0 stub 1 nssa
  Non-Stop Routing enabled
  Graceful restart helper support enabled
  Reference bandwidth unit is 100 mbps
  RFC1583 compatibility enabled
     Area BACKBONE(0) (Inactive)
         Number of interfaces in this area is 1
         SPF algorithm executed 3 times
         Number of LSA 6. Checksum Sum 0x03C938
         Number of DCbitless LSA 0
         Number of indication LSA 0
         Number of DoNotAge LSA 0
         Flood list length 0
     Area 1
         Number of interfaces in this area is 3
         SPF algorithm executed 3 times
         Number of LSA 6. Checksum Sum 0x024041
         Number of DCbitless LSA 0
         Number of indication LSA 0
         Number of DoNotAge LSA 0
         Flood list length 0
     Area 3
         Number of interfaces in this area is 1
         It is a NSSA area
         Perform type-7/type-5 LSA translation
         SPF algorithm executed 4 times
         Number of LSA 5. Checksum Sum 0x024910
         Number of DCbitless LSA 0
         Number of indication LSA 0
         Number of DoNotAge LSA 0
```

```
        Flood list length 0

  OSPFv3 1 address-family ipv6
  Router ID 10.0.0.1
  Supports NSSA (compatible with RFC 3101)
  Event-log enabled, Maximum number of events: 1000, Mode: cyclic
  It is an area border and autonomous system boundary router
  Redistributing External Routes from,
     ospf 2
  Router is not originating router-LSAs with maximum metric
  Initial SPF schedule delay 5000 msecs
  Minimum hold time between two consecutive SPFs 10000 msecs
  Maximum wait time between two consecutive SPFs 10000 msecs
  Minimum LSA interval 5 secs
  Minimum LSA arrival 1000 msecs
  LSA group pacing timer 240 secs
  Interface flood pacing timer 33 msecs
  Retransmission pacing timer 66 msecs
  Retransmission limit dc 24 non-dc 24
  Number of external LSA 0. Checksum Sum 0x000000
  Number of areas in this router is 3. 2 normal 0 stub 1 nssa
  Non-Stop Routing enabled
  Graceful restart helper support enabled
  Reference bandwidth unit is 100 mbps
  RFC1583 compatibility enabled
     Area BACKBONE(0) (Inactive)
         Number of interfaces in this area is 2
         SPF algorithm executed 2 times
         Number of LSA 6. Checksum Sum 0x02BAB7
         Number of DCbitless LSA 0
         Number of indication LSA 0
         Number of DoNotAge LSA 0
         Flood list length 0
     Area 1
         Number of interfaces in this area is 4
         SPF algorithm executed 2 times
         Number of LSA 7. Checksum Sum 0x04FF3A
         Number of DCbitless LSA 0
         Number of indication LSA 0
         Number of DoNotAge LSA 0
         Flood list length 0
     Area 3
         Number of interfaces in this area is 1
         It is a NSSA area
         Perform type-7/type-5 LSA translation
         SPF algorithm executed 3 times
         Number of LSA 5. Checksum Sum 0x011014
         Number of DCbitless LSA 0
         Number of indication LSA 0
         Number of DoNotAge LSA 0
         Flood list length 0
```

例：OSPFv3 ノンストップルーティングのステータスの確認

次に、OSPFv3 ノンストップルーティングのステータスを確認する例を示します。

```
Device# show ospfv3 1 nsr
  Active RP
  Operating in duplex mode
  Redundancy state: ACTIVE
  Peer redundancy state: STANDBY HOT
  Checkpoint peer ready
```

```
Checkpoint messages enabled
ISSU negotiation complete
ISSU versions compatible


        OSPFv3 1 address-family ipv4 (router-id 10.0.0.1)
NSR configured
Checkpoint message sequence number: 29
Standby synchronization state: synchronized
Bulk sync operations:  1
Next sync check time:  12:00:14.956 PDT Wed Jun 6 2012
LSA Count: 17, Checksum Sum 0x00085289


        OSPFv3 1 address-family ipv6 (router-id 10.0.0.1)
NSR configured
Checkpoint message sequence number: 32
Standby synchronization state: synchronized
Bulk sync operations:  1
Next sync check time:  12:00:48.537 PDT Wed Jun 6 2012
LSA Count: 18, Checksum Sum 0x0008CA05
```

出力には、OSPFv3 ノンストップルーティングが設定されていること、スタンバイ RP 上で OSPFv3 が完全に同期されていて、アクティブな RP に障害が発生したり切り替えが手動で実行されても操作を続行する準備ができていることが示されています。

トラブルシューティングのヒント

OSPFv3 ノンストップルーティングにより、OSPFv3 デバイスプロセスで使用されるメモリの量を増加できます。NSR なしで OSPFv3 が現在使用しているメモリの量を確認するには、**show processes** および **show processes memory** コマンドを使用します。

```
Device# show processes
 | include OSPFv3
 276 Mwe  133BE14         1900       1792    1060 8904/12000  0 OSPFv3-1 Router
 296 Mwe  133A824           10        971      10 8640/12000  0 OSPFv3-1 Hello
```

プロセス 276 は、確認する必要がある OSPFv3 デバイス プロセスです。このプロセスの現在のメモリ使用量を表示するには、**show processes memory** コマンドを使用します。

```
Device# show processes memory 276
Process ID: 276
Process Name: OSPFv3-1 Router
Total Memory Held: 4454800 bytes
```

この例では、OSPFv3 は 4,454,800 バイト、つまり約 4.5 メガバイト（MB）を使用しています。OSPFv3 ノンストップルーティングは短期間にこの倍のメモリを消費する場合があるため、OSPFv3 ノンストップルーティングをイネーブルにする前に、デバイスに少なくとも 5 MB の空きメモリがあることを確認してください。

その他の参考資料

標準

標準	タイトル
この機能でサポートされる新規の標準または変更された標準はありません。また、既存の標準のサポートは変更されていません。	—

MIB

MIB	MIB のリンク
この機能によってサポートされる新しい MIB または変更された MIB はありません。またこの機能による既存 MIB のサポートに変更はありません。	選択したプラットフォーム、Cisco ソフトウェアリリース、およびフィーチャセットの MIB を検索してダウンロードする場合は、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

RFC

RFC	タイトル
RFC 5187	『*OSPFv3 Graceful Restart*』

シスコのテクニカル サポート

説明	リンク
★枠で囲まれた Technical Assistance の場合★右の URL にアクセスして、シスコのテクニカルサポートを最大限に活用してください。これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。この Web サイト上のツールにアクセスする際は、Cisco.com のログイン ID およびパスワードが必要です。	http://www.cisco.com/cisco/web/support/index.html

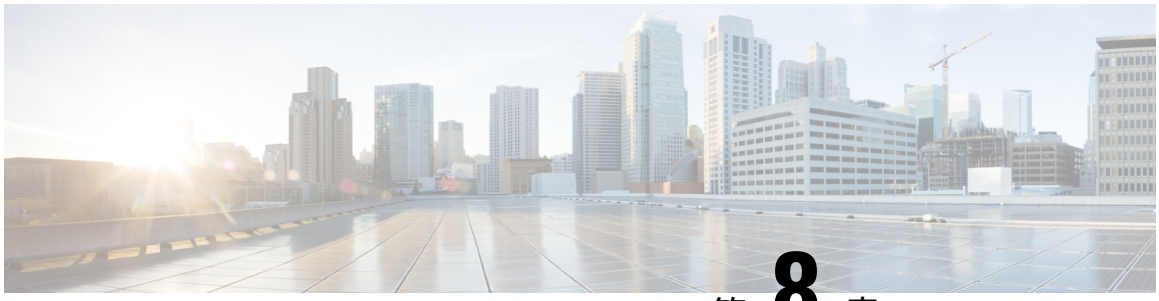
OSPFv3 ノンストップルーティングの機能情報

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

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

リリース	機能	機能情報
Cisco IOS XE Amsterdam 17.3.1	OSPFv3 ノンストップルーティング	OSPFv3 ノンストップルーティング機能により、冗長な RP を装備したルータが、計画されたおよび計画されていない RP スイッチオーバーで、OSPFv3 の状態と隣接関係を維持できるようになります。

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



第 8 章

OSPFv2 ループフリー代替 IP Fast Reroute の設定

OSPFv2 ループフリー代替（LFA）IP Fast Reroute（IP FRR）機能では、プライマリのネクストホップで障害が発生したときに、事前に計算された代替のネクストホップを使用して障害を軽減します。プレフィックスごとのLFAパスを設定し、プライマリネイバー以外のネクストホップにトラフィックをリダイレクトすることができます。他のルータが障害を知ることなく転送の決定が行われ、サービスが復元されます。

- OSPFv2 ループフリー代替 IP Fast Reroute の前提条件 （139 ページ）
- OSPFv2 ループフリー代替 IP Fast Reroute に関する制約事項 （139 ページ）
- OSPFv2 ループフリー代替 IP Fast Reroute に関する情報 （140 ページ）
- OSPFv2 ループフリー代替 IP Fast Reroute の設定方法 （143 ページ）
- OSPFv2 ループフリー代替 IP Fast Reroute の設定例 （147 ページ）
- OSPFv2 ループフリー代替 IP Fast Reroute の機能情報 （148 ページ）

OSPFv2 ループフリー代替 IP Fast Reroute の前提条件

Open Shortest Path First（OSPF）は、フォワーディング プレーンでこの機能をサポートするプラットフォームでのみ IP FRR をサポートします。プラットフォーム サポートについては、Cisco Feature Navigator（http://www.cisco.com/go/cfn）を参照してください。Cisco.com のアカウントは必要ありません。

OSPFv2 ループフリー代替 IP Fast Reroute に関する制約事項

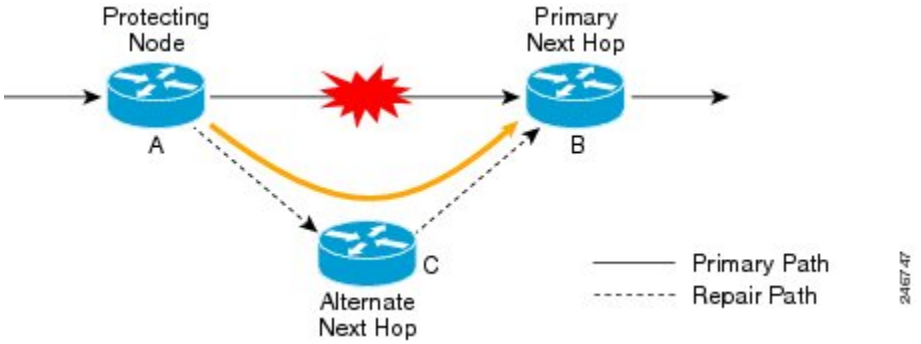
- IPv6 LFA IP FRRはサポートされていません。
- LFA IP FRR は、マルチプロトコル ラベル スイッチング（MPLS）としてのプライマリパスまたはバックアップパスではサポートされていません。

- LFA IP FRR は、等コストマルチパス（ECMP）としてのプライマリパスまたはバックアップパスではサポートされていません。
- LFA IP FRR は、OSPFv2 VRF-Lite ではサポートされていません。
- LFA IP FRR は、network-advantage ライセンスレベルでのみ使用できます。
- プライマリパスとしての Generic Routing Encapsulation（GRE）トンネルはサポートされていません。
- CPU 使用率が高い場合、コンバージェンス時間が長くなる可能性があります。
- コンバージェンス時間はプライマリリンクステータスの検出に依存するため、スイッチ仮想インターフェイス（SVI）やポートチャネルなどの論理インターフェイスの場合に物理リンクがダウンすると、コンバージェンス時間は長くなると予想されます。

OSPFv2 ループフリー代替 IP Fast Reroute に関する情報

LFA 修復パス

リンクに障害が発生した場合に OSPFv2 LFA IP FRR 機能がトラフィックを再ルーティングする方法を次の図に示します。保護ルータはプレフィックス単位の修復パスを事前に計算し、グローバルルーティング情報ベース（RIB）にこれらをインストールします。保護されたプライマリ パスで障害が発生すると、保護ルータはライブ トラフィックをプライマリ パスから格納された修復パスに転送します。このとき、他のルータはネットワーク トポロジを再計算する必要がなく、ネットワーク トポロジが変更されたことを認識する必要もありません。



LFA 修復パス属性

プライマリ パスで障害が発生すると、多数のパスが修復の候補になります。OSPFv2 LFA IP FRR 機能のデフォルト選択ポリシーでは次の順序で属性の優先順位が付けられています。

1. srlg
2. primary-path

3. interface-disjoint
4. lowest-metric
5. linecard-disjoint
6. node-protecting
7. broadcast-interface-disjoint

評価によって候補が選択されない場合、修復パスは暗黙的なロードバランシングによって選択されます。これは、修復パスの選択がプレフィックスによって変わることを意味します。

show ip ospf fast-reroute コマンドを使用すると、現在の設定を表示できます。

fast-reroute tie-break コマンドを使用すると、候補から選択するために、次のセクションで説明する 1 つ以上の修復パス属性を設定できます。

共有リスク リンク グループ

共有リスク リンク グループ（SRLG）は、同時に障害が発生する可能性が高い修復パスおよび保護されたプライマリ パスのネクストホップ インターフェイスのグループです。OSPFv2 LFA IP FRR 機能では、コンピューティングルータでローカルに設定された SRLG のみがサポートされます。単一の物理インターフェイス上の VLAN は SRLG の例です。物理インターフェイスで障害が発生すると、すべての VLAN インターフェイスが同時にエラーになります。デフォルトの修復パス属性では、ある VLAN のプライマリ パスが別の VLAN 上の修復パスによって保護される可能性があります。srlg 属性を設定すると、LFA 修復パスがプライマリ パスと同じ SRLG ID を共有しないように指定することができます。インターフェイスを SRLG に割り当てるには、**srlg** コマンドを使用します。

インターフェイスの保護

ポイントツーポイント インターフェイスには、プライマリ ゲートウェイで障害が発生した場合、再ルーティングのための代替のネクスト ホップはありません。interface-disjoint 属性を設定すると、このような修復パスの選択を防ぐことができるため、インターフェイスが保護されます。

ブロードキャスト インターフェイス保護

LFA 修復パスは、修復パスと保護されたプライマリ パスが異なるネクストホップ インターフェイスを使用するときにリンクを保護します。ただし、ブロードキャスト インターフェイスでは、LFA 修復パスがプライマリ パスと同じインターフェイスを介して計算されても、ネクストホップ ゲートウェイが異なる場合、ノードは保護されますがリンクは保護されないことがあります。broadcast-interface-disjoint 属性を設定すると、プライマリ パスがポイントするブロードキャスト ネットワークを修復パスが経由しない（つまり、インターフェイスと、これに接続されるブロードキャスト ネットワークを使用できない）ように指定することができます。

このタイブレーカーを必要とするネットワークトポロジについては、RFC 5286 の『*Basic Specification for IP Fast Reroute: Loop-Free Alternates*』にある「Broadcast and Non-Broadcast Multi-Access (NBMA) Links」を参照してください。

ノード保護

デフォルトの修復パス属性では、プライマリ パスのネクスト ホップであるルータは保護されないことがあります。ノード保護属性を設定すると、修復パスがプライマリパス ゲートウェイ ルータをバイパスするように指定することができます。

ダウンストリーム パス

高レベルのネットワーク障害や複数の同時ネットワーク障害が発生すると、代替パスを介して送信されるトラフィックはOSPFがプライマリ パスを再計算するまでループする可能性があります。downstream 属性を設定して、保護された宛先への修復パスのメトリックが保護ノードの宛先へのメトリックより小さくなる必要があるように指定することができます。これによりトラフィックが失われる可能性がありますが、ループは防止されます。

ラインカード **Disjoint** インターフェイス

ラインカードにラインカードの活性挿抜（OIR）などの問題がある場合、同じラインカード上のすべてのインターフェイスで同時に障害が発生するため、ラインカードインターフェイスは SRLG と似ています。linecard-disjoint 属性を設定すると、LFA 修復パスがプライマリパスのライン カードのものとは異なるインターフェイスを使用するように指定することができます。

メトリック

LFA 修復パスは最も効率的な候補である必要はありません。高レベルのネットワーク障害に対する保護機能を提供する場合、高コストな修理パスがより魅力的と考えられることがあります。メトリック属性を設定すると、最小のメトリックを持つ修復パス ポリシーを指定することができます。

等コスト マルチパス プライマリ パス

プライマリ最短パス優先（SPF）修復時に検出される等コスト マルチパス パス（ECMP）は、トラフィックが任意の単一リンクの容量を超過することがわかっているネットワーク設計では望ましくないことがあります。primary-path 属性を設定して ECMP セットから LFA 修復パスを指定したり、secondary-path 属性を設定して ECMP セットからでない LFA 修復パスを指定したりすることができます。

修復パスの候補リスト

OSPF は修復パスを計算するとき、メモリを節約するため、すべての候補パスのうちベスト パスのみをローカル RIB に保持します。**fast-reroute keep-all-paths** コマンドを使用すると、考えられたすべての修復パス候補のリストを作成できます。この情報はトラブルシューティングに役立つ可能性がありますが、メモリ消費が大幅に増加することがあるため、テストとデバッグを目的として使用する必要があります。

OSPFv2 ループフリー代替 IP Fast Reroute の設定方法

プレフィックスごとの OSPFv2 ループフリー代替 IP Fast Reroute の有効化

プレフィックスごとの OSPFv2 ループフリー代替 IP Fast Reroute を有効化して、OSPF エリアでのプレフィックス優先度を選択するには、次のタスクを実行します。

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	**router ospf** *process-id* 例： `Device(config)# router ospf 10`	OSPF ルーティングをイネーブルにして、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	**fast-reroute per-prefix enable prefix-priority** *priority-level* 例： `Device (config-router)# fast-reroute per-prefix enable prefix-priority low`	修復パス計算をイネーブルにし、修理パスのプライオリティ レベルを選択します。 • プライオリティを低くすると、すべてのプレフィックスの保護の基準が同じになります。プライオリティを高くすると、プライオリティの高いプレフィックスのみが保護されます。
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。

LFA IP FRR によるプレフィックス保護の指定

どのプレフィックスをLFA IP FRRで保護するかを指定するには、次の作業を実行します。ルートマップで指定されたプレフィックスだけが保護されます。

（注） ルートマップでは **match tag**、**match route-type**、**match ip address prefix-list** の 3 つの match キーワードだけが認識されます。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**route-map** *map-tag* [permit | deny] [*sequence-number*] 例： `Device(config)# route-map OSPF-PREFIX-PRIORITY`	ルート マップ コンフィギュレーション モードを開始し、マップ名を指定します。
ステップ 4	**match tag** *tag-name* 例： `Device(config-route-map)# match tag 886`	照合されるプレフィックスを指定します。 • タグと一致するプレフィックスだけが保護されます。
ステップ 5	**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 6	**router ospf** *process-id* 例： `Device(config)# router ospf 10`	OSPF ルーティングをイネーブルにして、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 7	**prefix-priority** *priority-level* **route-map** *map-tag* 例：	修復パスの優先度レベルを設定し、プレフィックスを定義するルート 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)# prefix-priority high route-map OSPF-PREFIX-PRIORITY`	
ステップ 8	**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。

修復パスの選択ポリシーの設定

タイブレーキング状態を指定して修復パス選択ポリシーを設定するには、次の作業を実行します。タイブレーキング属性の詳細については、「*LFA* 修復パス属性」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 10`	OSPF ルーティングをイネーブルにして、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	**fast-reroute per-prefix tie-break** *attribute* **[required] index** *index-level* 例： `Device(config-router)# fast-reroute per-prefix tie-break srlg required index 10`	タイブレーキング状態を指定して優先度レベルを設定することにより、修復パス選択ポリシーを設定します。
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。

考慮する修復パス リストの作成

LFA IP FRR に対して検討されるパスのリストを作成するには、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	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router ospf** *process-id* 例： `Device(config)# router ospf 10`	OSPF ルーティングをイネーブルにして、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	**fast-reroute keep-all-paths** 例： `Device(config-router)# fast-reroute keep-all-paths`	LFA FRR に対して検討されるパスのリストを作成するよう指定します。
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。

インターフェイスのネクスト ホップとしての使用の禁止

インターフェイスが修復パスでネクストホップとして使用されるのを禁止するには、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	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	**interface** *type number* 例： `Device(config)# interface Ethernet 1/0`	指定したインターフェイスのインターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**ip ospf fast-reroute per-prefix candidate disable** 例： `Device(config-if)# ip ospf fast-reroute per-prefix candidate disable`	インターフェイスが修復パスでネクストホップとして使用されるのを禁止します。
ステップ 5	**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。

OSPFv2 ループフリー代替 IP Fast Reroute の設定例

例：プレフィックスごとの LFA IP FRR のイネーブル化

次に、プレフィックスごとの OSPFv2 LFA IP FRR をイネーブル化して、OSPF エリアでのプレフィックス優先度を選択する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# router ospf 10
Device(config-router)# fast-reroute per-prefix enable prefix-priority low
Device(config-router)# end
```

例：プレフィックス保護優先度の指定

次に、どのプレフィックスを LFA FRR で保護するかを指定する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# router ospf 10
Device(config-router)# prefix-priority high route-map OSPF-PREFIX-PRIORITY
Device(config-router)# fast-reroute per-prefix enable prefix-priority high
Device(config-router)# network 192.0.2.1 255.255.255.0 area 0
Device(config-router)# route-map OSPF-PREFIX-PRIORITY permit 10
Device(config-router)# match tag 866
Device(config-router)# end
```

例：修復パスの選択ポリシーの設定

次に、タイブレーキング属性として、SRLG、ラインカード障害、およびダウンストリームを設定し、それらの優先度インデックスを設定する修復パス選択ポリシーを設定する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# router ospf 10
Device(config-router)# fast-reroute per-prefix enable prefix-priority low
Device(config-router)# fast-reroute per-prefix tie-break srlg required index 10
Device(config-router)# fast-reroute per-prefix tie-break linecard-disjoint index 15
Device(config-router)# fast-reroute per-prefix tie-break downstream index 20
Device(config-router)# network 192.0.2.1 255.255.255.0 area 0
Device(config-router)# end
```

例：修復パスの選択の監視

次に、修復パスの選択を記録する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# router ospf 10
Device(config-router)# fast-reroute per-prefix enable prefix-priority low
Device(config-router)# fast-reroute keep-all-paths
Device(config-router)# network 192.0.2.1 255.255.255.0 area 0
Device(config-router)# end
```

例：インターフェイスの保護インターフェイス化の禁止

次に、インターフェイスの保護インターフェイス化を禁止する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# interface Ethernet 0/0
Device(config-if)# ip address 192.0.2.1 255.255.255.0
Device(config-if)# ip ospf fast-reroute per-prefix candidate disable
Device(config-if)# end
```

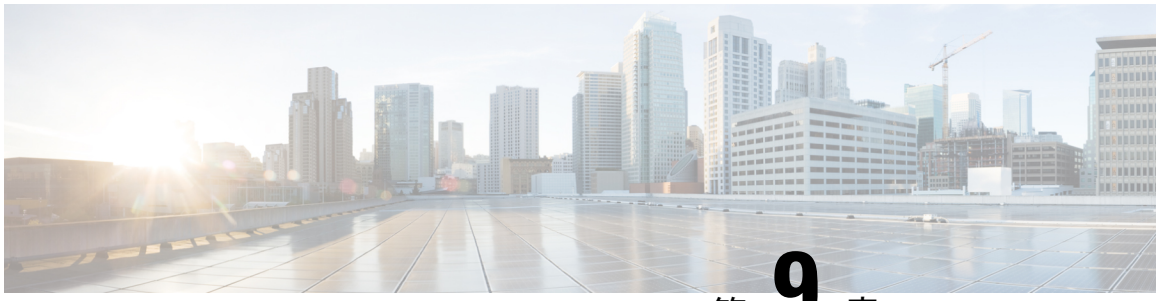
OSPFv2 ループフリー代替 IP Fast Reroute の機能情報

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

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

表 *17 : OSPFv2* ループフリー代替 *IP Fast Reroute* の機能情報

機能名	リリース	機能情報
OSPFv2 ループフリー代替 IP Fast Reroute	Cisco IOS XE Amsterdam 17.3.1	OSPFv2 ループフリー代替 IP Fast Reroute 機能では、プライマリのネクストホップで障害が発生したときに、事前に計算された代替のネクストホップを使用して障害を軽減します。



第 9 章

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

• OSPFv3 高速コンバージェンス：LSA および SPF スロットリング （151 ページ）

OSPFv3 高速コンバージェンス：LSA および SPF スロットリング

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

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

高速コンバージェンス：LSA および SPF スロットリング

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

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

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

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

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router ospfv3** [*process-id*] 例： `Device(config)# router ospfv3 1`	IPv4 または IPv6 アドレス ファミリの OSPFv3 ルータ コンフィギュレーション モードをイネーブルにします。
ステップ 4	**timers lsa arrival** *milliseconds* 例： `Device(config-rtr)# timers lsa arrival 300`	ソフトウェアが OSPFv3 ネイバーから同じ LSA を受け入れる最小間隔を設定します。
ステップ 5	**timers pacing flood** *milliseconds* 例： `Device(config-rtr)# timers pacing flood 30`	LSA フラッド パケット ペーシングを設定します。
ステップ 6	**timers pacing lsa-group** *seconds* 例： `Device(config-router)# timers pacing lsa-group 300`	OSPFv3 LSA を収集してグループ化し、リフレッシュ、チェックサム、またはエージングを行う間隔を変更します。
ステップ 7	**timers pacing retransmission** *milliseconds* 例： `Device(config-router)# timers pacing retransmission 100`	IPv4 OSPFv3 での LSA 再送信パケット ペーシングを設定します。

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

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ipv6 router ospf** *process-id* 例： `Device(config)# ipv6 router ospf 1`	OSPFv3 ルータ コンフィギュレーション モードをイネーブルにします。
ステップ 4	**timers throttle spf** *spf-start spf-hold spf-max-wait* 例： `Device(config-rtr)# timers throttle spf 200 200 200`	SPF スロットリングをオンにします。
ステップ 5	**timers throttle lsa** *start-interval hold-interval max-interval* 例： `Device(config-rtr)# timers throttle lsa 300 300 300`	OSPFv3 LSA 生成に対するレート制限値を設定します。
ステップ 6	**timers lsa arrival** *milliseconds* 例： `Device(config-rtr)# timers lsa arrival 300`	ソフトウェアが OSPFv3 ネイバーから同じ LSA を受け入れる最小間隔を設定します。
ステップ 7	**timers pacing flood** *milliseconds* 例： `Device(config-rtr)# timers pacing flood 30`	LSA フラッド パケット ペーシングを設定します。

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

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

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

```
Device# show ipv6 ospf

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

その他の参考資料

関連資料

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

標準および RFC

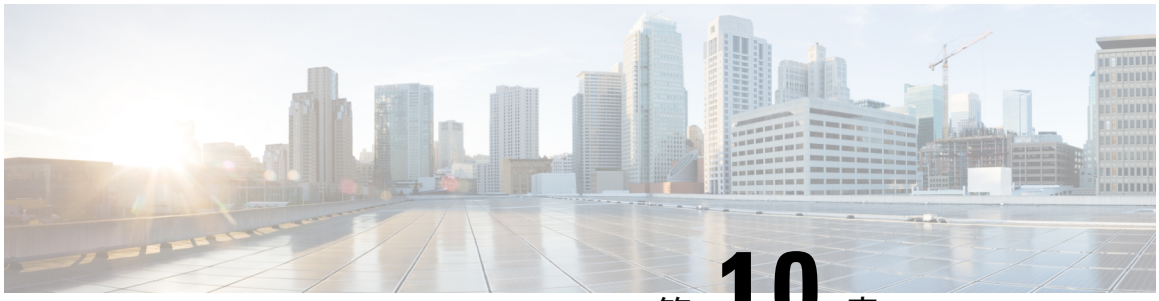
標準/RFC	タイトル
IPv6 に関する RFC	IPv6 RFCs

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

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

表 *18 : OSPFv3* 高速コンバージェンス：*LSA* および *SPF* スロットリングの機能情報

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



第 10 章

IPsec を使用した OSPFv3 認証サポートの設定

- IPsec を使用した OSPFv3 認証サポートに関する情報 （157 ページ）
- IPsec を使用した OSPFv3 認証サポートの設定方法 （159 ページ）
- OSPFv3 IPSec ESP 暗号化および認証の設定方法 （161 ページ）
- IPsec を使用した OSPFv3 認証サポートの設定例 （164 ページ）
- OSPFv3 IPSec ESP 暗号化および認証の設定例 （164 ページ）
- IPsec を使用した OSPFv3 認証サポートの機能履歴と機能情報 （165 ページ）

IPsec を使用した OSPFv3 認証サポートに関する情報

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

IPsec を使用した OSPFv3 認証サポートの概要

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

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

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

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

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

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

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

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

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

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

OSPFv3 仮想リンク

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

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

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

IPsec を使用した OSPFv3 認証サポートの設定方法

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

インターフェイスでの認証の定義

インターフェイスで認証を定義するには、次の手順を実行します。

始める前に

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

手順

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

	コマンドまたはアクション	目的
	`Device> enable`	プロンプトが表示されたらパスワードを入力します。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *type number* 例： `Device(config)# interface ethernet 1/0/1`	インターフェイスを設定します。

ステップ 4	次のいずれかを選択します。
	• **ospfv3 authentication** {{ **ipsec spi** *spi* {**md5** | **sha1**}} | { *key-encryption-type key* } | **null**}
	• **ipv6 ospf authentication** {**null** | **ipsec spi** *spi authentication-algorithm* [*key-encryption-type*] [*key*]}
	例：
	`Device(config-if)# ospfv3 authentication md5 0 275761340947681324733020312097 27`
	または
	`Device(config-if)# ipv6 ospf authentication ipsec spi 500 md5 1234567890abcdef1234567890abcdef`
		インターフェイスの認証タイプを指定します。

OSPFv3 エリア内の認証の定義

OSPFv3 エリア内で認証を定義するには、次の手順を実行します。

手順

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

	コマンドまたはアクション	目的
ステップ 3	**ipv6 router ospf** *process-id* 例： `Device(config)# ipv6 router ospf 1`	OSPFv3 ルータ コンフィギュレーション モードをイネーブルにします。
ステップ 4	**area** *area-id* **authentication ipsec spi** *spi authentication-algorithm* [*key-encryption-type*] *key* 例： `Device(config-router)# area 1 authentication ipsec spi 678 md5 1234567890ABCDEF1234567890ABCDEF`	OSPFv3 エリア内の認証をイネーブルにします。

OSPFv3 IPSec ESP 暗号化および認証の設定方法

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

インターフェイスでの暗号化の定義

インターフェイスで暗号化を定義するには、次の手順を実行します。

始める前に

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

手順

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

	コマンドまたはアクション	目的

ステップ 4	次のいずれかを選択します。
	• **ospfv3 authentication** { **ipsec spi** *spi* **esp** *encryption-algorithm key-encryption-type key authentication-algorithm key-encryption-type key* | **null**}
	• **ipv6 ospf authentication** { **ipsec spi** *spi* **esp** {*encryption-algorithm* [*key-encryption-type*] *key* | **null**} *authentication-algorithm* [*key-encryption-type*] *key*] | **null**}
	例：
	`Device(config-if)# ospfv3 encryption ipsec spi 1001 esp null md5 0 275761340947681324733020312097 27`
	または
	`Device(config-if)# ipv6 ospf encryption ipsec spi 1001 esp null sha1 123456789A123456789B123456789C123456789D`
		インターフェイスに暗号化タイプを指定します。

OSPFv3 エリア内の暗号化の定義

OSPFv3 エリアで暗号化を定義するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ipv6 router ospf** *process-id* 例： `Device(config)# ipv6 router ospf 1`	OSPFv3 ルータ コンフィギュレーション モードをイネーブルにします。
ステップ 4	**area** *area-id* **encryption ipsec spi** *spi* **esp** {*encryption-algorithm* [*key-encryption-type*] *key* | **null**} *authentication-algorithm* [*key-encryption-type*] *key*	OSPFv3 エリア内の暗号化をイネーブルにします。

	コマンドまたはアクション	目的
	例： `Device(config-router)# area 1 encryption ipsec spi 500 esp null md5 1aaa2bbb3ccc4ddd5eee6fff7aaa8bbb`	

OSPFv3 エリア内の仮想リンクに対する認証および暗号化の定義

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ipv6 router ospf** *process-id* 例： `Device(config)# ipv6 router ospf 1`	OSPFv3 ルータ コンフィギュレーション モードをイネーブルにします。
ステップ 4	**area** *area-id* **virtual-link** *router-id* **authentication ipsec spi** *spi authentication-algorithm* [*key-encryption-type*] *key* 例： `Device(config-router)# area 1 virtual-link 10.0.0.1 authentication ipsec spi 940 md5 1234567890ABCDEF1234567890ABCDEF`	OSPFv3 エリア内の仮想リンクに対して認証をイネーブルにします。
ステップ 5	**area** *area-id* **virtual-link** *router-id* **authentication ipsec spi** *spi* **esp** {*encryption-algorithm* [*key-encryption-type*] *key* | **null**} *authentication-algorithm* [*key-encryption-type*] *key* 例： `Device(config-router)# area 1 virtual-link 10.1.0.1 hello-interval 2 dead-interval 10 encryption ipsec`	OSPFv3 エリア内の仮想リンクに対して暗号化をイネーブルにします。

	コマンドまたはアクション	目的
	spi 3944 esp null sha1 123456789A123456789B123456789C123456789D	

IPsec を使用した OSPFv3 認証サポートの設定例

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

例：インターフェイスでの認証の定義

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

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

例：OSPFv3 エリア内の認証の定義

次に、OSPFv3 エリア 0 で認証を定義する例を示します。

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

OSPFv3 IPSec ESP 暗号化および認証の設定例

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

例：OSPFv3 エリアでの暗号化の確認

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

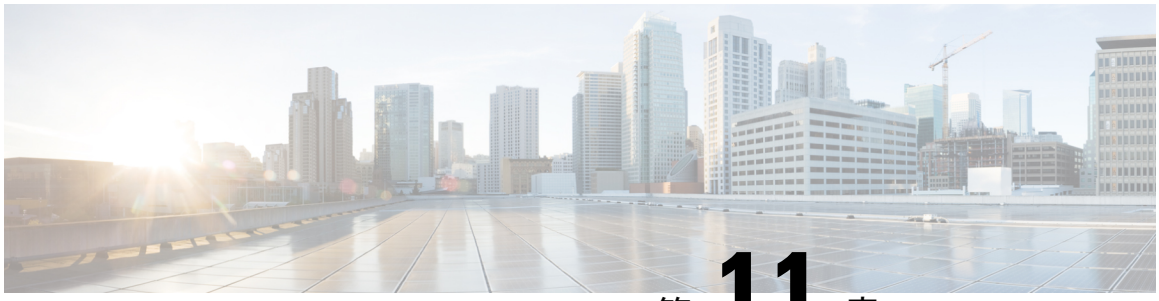
```
Device> enable
Device# show ipv6 ospf interface

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

IPsec を使用した OSPFv3 認証サポートの機能履歴と機能情報

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

表 *19 : IPsec* を使用した *OSPFv3* 認証サポートの機能履歴

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



第 11 章

OSPFv3 認証トレーラの設定

- OSPFv3 認証トレーラに関する情報 （167 ページ）
- OSPFv3 認証トレーラの設定方法 （168 ページ）
- OSPFv3 認証トレーラの設定例 （170 ページ）
- OSPFv3 認証トレーラに関する追加情報 （172 ページ）
- OSPFv3 認証トレーラの機能情報 （172 ページ）

OSPFv3 認証トレーラに関する情報

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

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

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

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

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

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

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

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

OSPFv3 認証トレーラの設定方法

OSPFv3 認証トレーラを設定するには、次の手順を実行します。

始める前に

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *type number* 例： `Device(config)# interface GigabitEthernet 2/0/1`	インターフェイス タイプおよび番号を指定します。
ステップ 4	**ospfv3** [*pid*] [**ipv4** | **ipv6**] **authentication** { **key-chain** *chain-name* | **null**} 例： `Device(config-if)# ospfv3 1 ipv6 authentication key-chain ospf-1`	OSPFv3 インターフェイスの認証タイプを指定します。
ステップ 5	**router ospfv3** [*process-id*] 例： `Device(config-if)# router ospfv3 1`	OSPFv3 ルータ コンフィギュレーション モードを開始します。
ステップ 6	**address-family ipv6 unicast** 例： `Device(config-router)# address-family ipv6 unicast`	OSPFv3 プロセスに IPv6 アドレス ファミリを設定し、IPv6 アドレス ファミリ コンフィギュレーション モードを開始します。
ステップ 7	**area** *area-id* **authentication** {**key-chain** *chain-name* | **null**} 例： `Device(config-router-af)# area 1 authentication key-chain ospf-chain-1`	OSPFv3 エリア内のすべてのインターフェイスの認証トレーラを設定します。
ステップ 8	**area** *area-id* **virtual-link** *router-id* **authentication key-chain** *chain-name* 例： `Device(config-router-af)# area 1 virtual-link 1.1.1.1 authentication key-chain ospf-chain-1`	仮想リンクの認証を設定します。
ステップ 9	**area** *area-id* **sham-link** *source-address destination-address* **authentication key-chain** *chain-name*	模造リンクの認証を設定します。

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

OSPFv3 認証トレーラの設定例

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

例：OSPFv3 認証トレーラの設定

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

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

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

例：OSPFv3 認証トレーラの確認

次に、**show ospfv3** コマンドの出力例を示します

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

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

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

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

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

OSPFv3 認証トレーラに関する追加情報

関連資料

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

標準および RFC

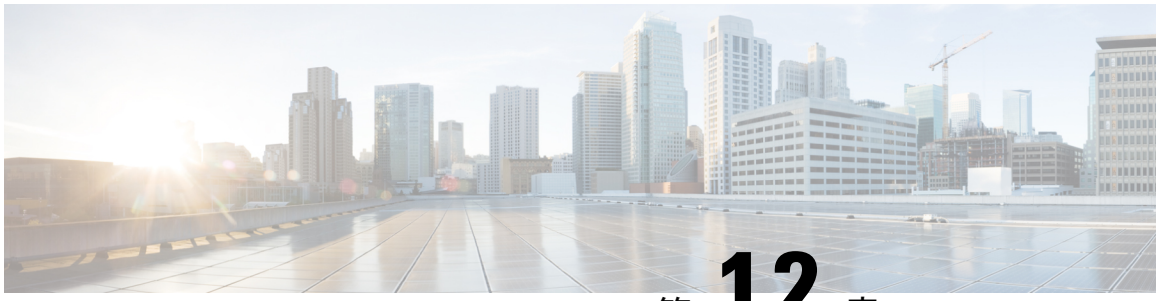
標準/RFC	マニュアル タイトル
RFC 7166	OSPFv3 認証トレーラのサポートに関する RFC
RFC 6506	OSPFv3 認証トレーラのサポートに関する RFC
RFC 4552	OSPFv3 の認証/機密性に関する RFC

OSPFv3 認証トレーラの機能情報

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

表 *20 : OSPFv3* 認証トレーラの機能情報

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



第 12 章

OSPFv3 のルート再配布数制限の設定

- OSPFv3 のルート再配布数の制限に関する制約事項 （173 ページ）
- OSPFv3 のルート再配布数制限の前提条件 （173 ページ）
- OSPFv3 のルート再配布数制限について （173 ページ）
- OSPFv3 のルート再配布数制限を設定する方法 （174 ページ）
- OSPFv3 のルート再配布数制限の設定例 （176 ページ）
- OSPFv3 のルート再配布数制限のモニタリング （177 ページ）
- その他の参考資料 （178 ページ）
- OSPFv3 のルート再配布数制限の機能情報 （178 ページ）

OSPFv3 のルート再配布数の制限に関する制約事項

この機能は、IPv6 アドレスファミリについてのみサポートされています。

OSPFv3 のルート再配布数制限の前提条件

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

OSPFv3 のルート再配布数制限について

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

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

OSPFv3 のルート再配布数制限を設定する方法

ここでは、OSPFv3 のルート再配布数制限の設定について説明します。

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

OSPFv3 のルート再配布数の制限

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

手順

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

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

OSPFv3 へのルートの再配布数に関する警告メッセージの要求

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

手順

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

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

OSPFv3 のルート再配布数制限の設定例

ここでは、OSPFv3 のルート再配布数制限の設定例を示します。

例：OSPFv3 のルート再配布数の制限

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

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

例：ルートの再配布数に関する警告メッセージの要求

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

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

OSPFv3 のルート再配布数制限のモニタリング

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

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

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

その他の参考資料

関連資料

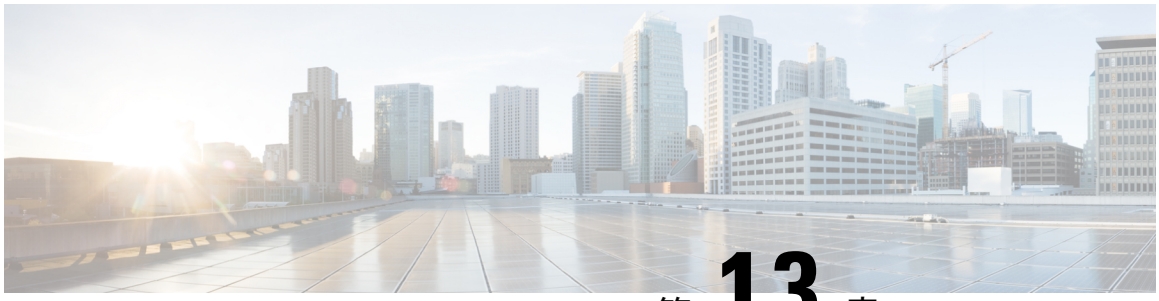
関連項目	マニュアル タイトル
この章で使用するコマンドの完全な構文および使用方法の詳細。	次のドキュメントのルーティングに関する項を参照してください：*Command Reference (Catalyst 9400 Series Switches)*

OSPFv3 のルート再配布数制限の機能情報

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

表 *22 : OSPFv3* のルート再配布数制限の機能情報

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



第 13 章

EIGRP の設定

- EIGRP に関する情報 （179 ページ）
- EIGRP の設定方法 （185 ページ）
- EIGRP のモニタリングおよびメンテナンス （193 ページ）
- EIGRP の機能情報 （194 ページ）

EIGRP に関する情報

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

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

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

EIGRP IPv6

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

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

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

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

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

EIGRP の機能

EIGRP には次の機能があります。

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

EIGRP コンポーネント

EIGRP には次に示す 4 つの基本コンポーネントがあります。

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

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

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

EIGRP NSF

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

- EIGRP NSF 認識
- EIGRP NSF 対応

EIGRP NSF 認識

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

EIGRP NSF 対応

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

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

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

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

EIGRP スタブ ルーティング

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

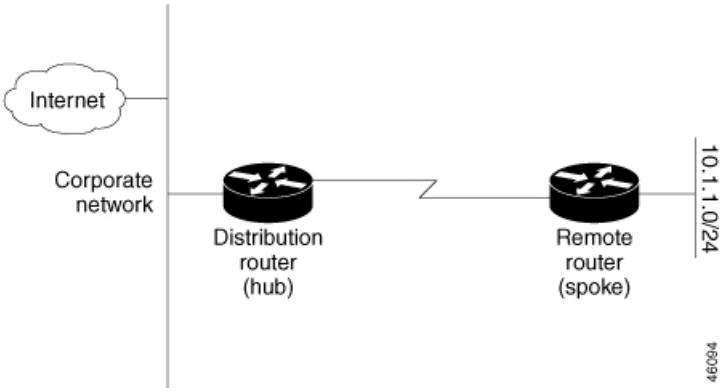
スタブルーティングは一般にハブアンドスポーク型のネットワークトポロジで使用されます。ハブアンドスポーク型ネットワークでは、1 つ以上のエンド（スタブ）ネットワークが 1 台のリモートデバイス（スポーク）に接続され、そのリモートデバイスは 1 つ以上のディストリビューションデバイス（ハブ）に接続されています。リモートデバイスは、1つ以上のディストリビューション デバイスに隣接しています。IP トラフィックがリモートデバイスに到達するための唯一のルートは、ディストリビューションデバイスを経由するものです。このタイプの設定は、一般的に、ディストリビューション デバイスが WAN に直接接続されている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

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

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

次の図は、単純なハブアンドスポーク型ネットワークを示しています。

図 *5 :* 単純なハブアンドスポーク型ネットワーク



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

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

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

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

EIGRPv6 スタブ ルーティング

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

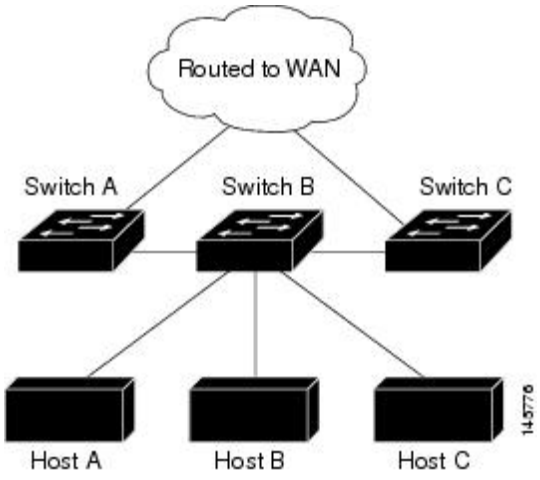
EIGRPv6 スタブ ルーティングを使用するネットワークでは、ユーザに対する IPv6 トラフィックの唯一の許容ルートは、EIGRPv6 スタブ ルーティングを設定しているスイッチ経由です。スイッチは、ユーザインターフェイスとして設定されているインターフェイスまたは他のデバイスに接続されているインターフェイスにルーテッド トラフィックを送信します。

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

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

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

図 *6 : EIGRP* スタブ ルータ設定



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

EIGRP の設定方法

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

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

EIGRP のデフォルト設定

表 *23 : EIGRP* のデフォルト設定

機能	デフォルト設定
自動サマリー	ディセーブル
デフォルト情報	再配信中は外部ルートが許可され、EIGRP プロセス間でデフォルト情報が渡されます。
デフォルト メトリック	デフォルト メトリックなしで再配信できるのは、接続されたルートおよびインターフェイスのスタティック ルートだけです。デフォルト メトリックは次のとおりです。 • 帯域幅：0 以上の kb/s • 遅延（10 マイクロ秒）：0 または 39.1 ナノ秒の倍数である任意の正の数値 • 信頼性：0 ～ 255 の任意の数値（255 の場合は信頼性が 100%） • 負荷：0 ～ 255 の数値で表される有効帯域幅（255 の場合は 100% の負荷） • MTU：バイトで表されたルートの MTU サイズ（0 または任意の正の整数）

機能	デフォルト設定
ディスタンス	内部距離：90 外部距離：170
EIGRP の隣接関係変更ログ	ディセーブル隣接関係の変更はロギングされません。
IP 認証キーチェーン	認証なし
IP 認証モード	認証なし
IP 帯域幅比率	50%
IP hello 間隔	低速非ブロードキャスト マルチアクセス（NBMA）ネットワークの場合：60 秒、それ以外のネットワークの場合：5 秒
IP ホールドタイム	低速 NBMA ネットワークの場合：180 秒、それ以外のネットワークの場合：15 秒
IP スプリットホライズン	イネーブル
IP サマリー アドレス	サマリー集約アドレスは未定義
メトリック重み	tos：0、k1 および k3：1、k2、k4、および k5：0
ネットワーク	指定なし
ノンストップ フォワーディング（NSF）認識	レイヤ 3 スイッチでは、ハードウェアやソフトウェアの変更中に、隣接する NSF 対応ルータからのパケットを転送し続けることができます。
NSF 対応	ディセーブル （注） デバイスは EIGRP NSF 対応ルーティングを IPv4 に対してサポートします。
オフセットリスト	ディセーブル
ルータ EIGRP	ディセーブル
メトリック設定	ルート マップにはメトリック設定なし
トラフィック共有	メトリックの比率に応じて配分
バリアンス	1（等コスト ロード バランシング）

基本的な EIGRP パラメータの設定

手順

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

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

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

EIGRP インターフェイスの設定

インターフェイスごとに、他の EIGRP パラメータを任意で設定できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： Device>enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： Device#**configure terminal**	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface***interface-id* 例： Device(config)#interface gigabitethernet 1/0/1	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 4	**ip bandwidth-percent eigrp** *percent* 例： Device(config-if)#ip bandwidth-percent eigrp 60	（任意）インターフェイスで EIGRP が使用できる帯域幅の割合を設定します。デフォルト値は 50% です。
ステップ 5	**ip summary-address eigrp** *autonomous-system-number address mask* 例： Device(config-if)#ip summary-address eigrp 109 192.161.0.0 255.255.0.0	（任意）指定されたインターフェイスのサマリー集約アドレスを設定します（auto-summary がイネーブルの場合は、通常設定する必要はありません）。
ステップ 6	**ip hello-interval eigrp** *autonomous-system-number seconds* 例：	（任意）EIGRP ルーティング プロセスの hello 時間間隔を変更します。指定できる範囲は 1 ～ 65535 秒です。低速 NBMA ネットワークの場合のデフォル

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

IPv6 の EIGRP の設定

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

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

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

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

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

EIGRP ルート認証の設定

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

手順

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

	コマンドまたはアクション	目的
ステップ 5	**ip authentication key-chain eigrp** *autonomous-system key-chain* 例： `Device(config-if)#ip authentication key-chain eigrp 105 chain1`	IP EIGRP パケットの認証をイネーブルにします。
ステップ 6	**exit** 例： `Device(config-if)#exit`	グローバル コンフィギュレーション モードに戻ります。
ステップ 7	**key chain** *name-of-chain* 例： `Device(config)#key chain chain1`	キーチェーンを識別し、キーチェーン コンフィギュレーション モードを開始します。ステップ 4 で設定した名前を指定します。
ステップ 8	**key** *number* 例： `Device(config-keychain)#key 1`	キーチェーン コンフィギュレーション モードで、キー番号を識別します。
ステップ 9	**key-string** *text* 例： `Device(config-keychain-key)#key-string key1`	キーチェーン コンフィギュレーション モードで、キー ストリングを識別します。
ステップ 10	**accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} 例： `Device(config-keychain-key)#accept-lifetime 13:30:00 Jan 25 2011 duration 7200`	（任意）キーを受信できる期間を指定します。 *start-time* および *end-time* 構文には、*hh:mm:ss Month date year* または *hh:mm:ss date Month year* のいずれかを使用できます。デフォルトは、デフォルトの *start-time* 以降、無制限です。指定できる最初の日付は 1993 年 1 月 1 日です。デフォルトの *end-time* および **duration** は **infinite** です。
ステップ 11	**send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} 例： `Device(config-keychain-key)#send-lifetime 14:00:00 Jan 25 2011 duration 3600`	（任意）キーを送信できる期間を指定します。 *start-time* および *end-time* 構文には、*hh:mm:ss Month date year* または *hh:mm:ss date Month year* のいずれかを使用できます。デフォルトは、デフォ

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

EIGRP のモニタリングおよびメンテナンス

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

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

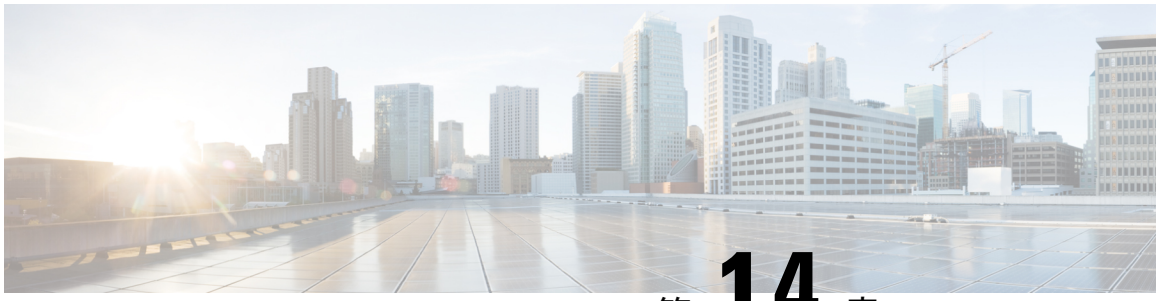
コマンド	目的
clear ip eigrp neighbors [*if-address* | *interface*]	ネイバーテーブルからネイバーを削除します。
show ip eigrp interface [*interface*] [*as number*]	EIGRP に設定されているインターフェイスに関する情報を表示します。
show ip eigrp neighbors [*type-number*]	EIGRP によって検出されたネイバーを表示します。
show ip eigrp topology [*autonomous-system-number*] | [[*ip-address*]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

EIGRP の機能情報

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

表 *25 : EIGRP* 機能の機能情報

リリース	機能情報
Cisco IOS XE Everest 16.6.1	この機能が導入されました。



第 14 章

EIGRP ループフリー代替 IP Fast Reroute の設定

Enhanced Interior Gateway Routing Protocol（EIGRP）ループフリー代替（LFA）IP Fast Reroute（IP FRR）機能により、EIGRP は、修復パスまたはバックアップルートを事前に計算し、これらのパスまたはルートをルーティング情報ベース（RIB）にインストールすることで、ルーティングの遷移時間を 50 ミリ秒未満に短縮できます。FRR は、障害が発生したリンクを通過するトラフィックを再ルーティングして障害を回避させることを可能にするメカニズムです。EIGRP ネットワークでは、事前に計算されたバックアップルートまたは修復パスは、フィージブルサクセサまたは LFA と呼ばれます。このモジュールでは、EIGRP ループフリー代替 Fast Reroute 機能を設定し、EIGRP によって識別されるフィージブルサクセサまたは LFA のロードシェアリングおよびタイブレーク設定を有効にする方法について説明します。

- EIGRP ループフリー代替 IP Fast Reroute に関する制約事項 （195 ページ）
- EIGRP ループフリー代替 IP Fast Reroute に関する情報 （196 ページ）
- EIGRP ループフリー代替 IP Fast Reroute の設定方法 （198 ページ）
- EIGRP ループフリー代替 IP Fast Reroute の設定例 （201 ページ）
- EIGRP ループフリー代替 IP Fast Reroute の機能情報 （203 ページ）

EIGRP ループフリー代替 IP Fast Reroute に関する制約事項

- IPv6 LFA IP FRRはサポートされていません。
- LFA IP FRR は、マルチプロトコル ラベル スイッチング（MPLS）としてのプライマリパスまたはバックアップパスではサポートされていません。
- LFA IP FRR は、等コストマルチパス（ECMP）としてのプライマリパスまたはバックアップパスではサポートされていません。
- LFA IP FRR は、network-advantage ライセンスレベルでのみ使用できます。
- プライマリパスとしての Generic Routing Encapsulation（GRE）トンネルはサポートされていません。
- CPU 使用率が高い場合、コンバージェンス時間が長くなる可能性があります。

- コンバージェンス時間はプライマリリンクステータスの検出に依存するため、スイッチ仮想インターフェイス（SVI）やポートチャネルなどの論理インターフェイスの場合に物理リンクがダウンすると、コンバージェンス時間は長くなると予想されます。

EIGRP ループフリー代替 IP Fast Reroute に関する情報

修復パスの概要

リンクまたはデバイスに障害が発生すると、分散ルーティングアルゴリズムによって新しいルートまたは修復パスが計算されます。この計算のための時間をルーティングの遷移と呼びます。遷移が完了し、すべてのデバイスがネットワーク上の共通のビューで収束されるまで、デバイスの送信元/宛先ペア間の接続は中断されます。修復パスでは、ルーティングの遷移時にトラフィックが転送されます。

リンクまたはデバイスに障害が発生すると、最初は隣接デバイスだけが障害を認識します。ネットワーク内の他のデバイスはすべて、この障害に関する情報がルーティングプロトコルによって伝播されるまで、この障害の性質と場所を認識しません。この情報の伝播には数百ミリ秒かかる場合があります。その間、ネットワーク障害の影響を受けるパケットをそれぞれの宛先に誘導する必要があります。障害が発生したリンクに隣接するデバイスは、障害が発生したリンクを使用していた可能性のあるパケットに対して、一連の修復パスを使用します。これらの修復パスは、ルータが障害を検出してから、ルーティングの遷移が完了するまで使用されます。ルーティングの遷移が完了するまでに、ネットワーク内のすべてのデバイスは転送データを変更し、障害が発生したリンクはルーティングの計算から除外されます。ルーティングプロトコルは、障害が検出されるとすぐに修復パスをアクティブ化できるように、障害を予測して修復パスを事前に計算します。EIGRP ネットワークでは、事前に計算された修復パスまたはバックアップルートは、フィージブルサクセサまたは LFA と呼ばれます。

LFA 計算

LFA は、ループバックしないで宛先にパケットを送る事前計算されたネクストホップルートです。ネットワーク障害が発生するとトラフィックは LFA にリダイレクトされ、LFA は障害を認識せずに転送を決定します。

内部ゲートウェイプロトコル（IGP）は、次の 2 つの方法で LFA を計算します。

- リンクごと（リンクベース）の計算：リンクベース LFA では、プライマリ（保護される）リンクを介して到達できるすべてのプレフィックスは、同じバックアップ情報を共有します。つまり、プライマリリンクを共有するプレフィックスの全体のセットは、修復または Fast Reroute（FRR）機能も共有します。リンクごとの方法は、ネクストホップアドレスだけが保護されます。宛先ノードは必ずしも保護する必要がありません。そのため、プライマリリンクからのすべてのトラフィックが複数のパスに分散されるのではなくネクストホップにリダイレクトされるので、リンクごとの方法は次善策であり、キャパシティプランに最適なアプローチではありません。すべてのトラフィックをネクストホップにリダイレクトすると、ネクストホップへのリンクで輻輳が発生する可能性があります。

- プレフィックスごと（プレフィックスベース）の計算：プレフィックスベース LFA は、プレフィックス（ネットワーク）ごとのバックアップ情報の計算と、宛先アドレスの保護を可能にします。プレフィックスごとの方法は、適用性や帯域幅利用率が優れているため、リンクごとの方法よりも推奨されます。プレフィックスごとの計算では、可能なすべての LFA が評価され、タイブレーカーを使用して利用可能な LFA の中から最適な LFA が選択されるため、プレフィックスごとの計算はリンクごとの計算よりも優れたロードシェアリングと保護範囲を提供します。

（注） プレフィックスベースの LFA を使用してプライマリパスで計算される修復またはバックアップ情報は、リンクベースの LFA を使用して計算されるものとは異なることがあります。

EIGRP は、常に、プレフィックスベースの LFA を計算します。EIGRP は、Diffusing Update Algorithm（DUAL）を使用してサクセサおよびフィージブルサクセサを計算します。EIGRP は、サクセサをプライマリパスとして使用し、フィージブルサクセサを修復パスまたは LFA として使用します。

LFA タイブレークルール

特定のプライマリパスに複数の候補 LFA がある場合、EIGRP は、タイブレークルールを使用して、プレフィックス単位のプライマリパスごとに 1 つの LFA を選択します。タイブレークルールは、特定の条件を満たすか特定の属性を持つ LFA を考慮します。EIGRP は、次の 4 つの属性を使用してタイブレークルールを実装します。

- interface-disjoint：保護されたパスと発信インターフェイスを共有する LFA を排除します。
- linecard-disjoint：保護されたパスとラインカードを共有する LFA を排除します。
- lower-repair-path-metric：保護されたプレフィックスに対するメトリックが高い LFA を排除します。このタイブレーカーが適用された後、同じ最小パスメトリックを持つ複数の LFA がルーティングテーブルに残る場合があります。
- srlg-disjoint：保護されたパス SRLG（共有リスクリンクグループ）のいずれかに属する LFA を排除します。SRLG は、ネットワーク内のリンクが共通のファイバ（または共通の物理属性）を共有する状況を意味します。1 つのリンクで障害が発生すると、グループ内の他のリンクでも障害が発生する可能性があります。そのため、グループ内のリンクはリスクを共有します。

EIGRP ループフリー代替 IP Fast Reroute の設定方法

プレフィックスごとの LFA IP FRR の設定

EIGRP ネットワークでプレフィックスごとに LFA IPFRR を設定するには、次のタスクを実行します。EIGRP トポロジの使用可能なすべてのプレフィックス、またはルートマップで指定されたプレフィックスに対して、LFA を有効にできます。

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 eigrp** *virtual-name* 例： `Device(config)# router eigrp 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 4	**address-family ipv4 autonomous-system** *autonomous-system-number* 例： `Device(config-router)# address-family ipv4 autonomous-system 1`	IPv4 VRF アドレス ファミリ コンフィギュレーション モードを開始して、EIGRP ルーティング インスタンスを設定します。
ステップ 5	**topology base** 例： `Device(config-router-af)# topology base`	基本 EIGRP トポロジを設定し、ルータ 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 6	**fast-reroute per-prefix {all | route-map** *route-map-name***}** 例： `Device(config-router-af-topology)# fast-reroute per-prefix all`	トポロジ内のすべてのプレフィックスに対して IPFRR を有効にします。 • ルートマップによって指定されたプレフィックスで IP FRR を有効にするには、**route-map** キーワードを入力します。

	コマンドまたはアクション	目的
ステップ 7	**end** 例： `Device(config-router-af-topology)# end`	ルータ アドレス ファミリ トポロジ コンフィギュレーション モードを終了して、特権 EXEC モードに戻ります。
ステップ 8	**show ip eigrp topology frr** 例： `Device# show ip eigrp topology frr`	EIGRP トポロジ テーブルで設定されている LFA のリストを表示します。

プレフィックス間のロードシェアリングの無効化

プライマリパスが複数の LFA を持つ等コストマルチパス（ECMP）パスである場合、ECMP パスのデフォルトの動作はロードシェアリングであるため、プレフィックス（ネットワーク）は LFA 間で均等に分散されます。ただし、タイブレーク設定を有効にすることで、LFA の選択を制御できます。プレフィックス間のロードシェアリングを無効にするには、次のタスクを実行します。

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	**router eigrp** *virtual-name* 例： `Device(config)# router eigrp 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 4	**address-family ipv4 autonomous-system** *autonomous-system-number* 例： `Device(config-router)# address-family ipv4 autonomous-system 1`	IPv4 VRF アドレス ファミリ コンフィギュレーション モードを開始して、EIGRP ルーティング インスタンスを設定します。
ステップ 5	**topology base** 例： `Device(config-router-af)# topology base`	基本 EIGRP トポロジを設定し、ルータ 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。

	コマンドまたはアクション	目的
ステップ 6	**fast-reroute load-sharing disable** 例： `Device(config-router-af-topology)# fast-reroute load-sharing disable`	プレフィックス間のロードシェアリングを無効にします。
ステップ 7	**end** 例： `Device(config-router-af-topology)# end`	ルータ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て、特権 EXEC モードに戻ります。
ステップ 8	**show ip eigrp topology frr** 例： `Device# show ip eigrp topology frr`	EIGRP トポロジテーブルで設定されているフィージブルサクセサまたは LFA のリストを表示します。

EIGRP LFA のタイブレークルールの有効化

特定のプライマリパスに複数の LFA がある場合に単一の LFA を選択するためのタイブレークルールを有効にするには、このタスクを実行します。EIGRP では、4 つの属性を使用してタイブレークルールを設定できます。**fast-reroute tie-break** コマンドの **interface-disjoint**、**linecard-disjoint**、**lowest-backup-path-metric**、および **srlg-disjoint** キーワードを使用すると、それぞれ、特定の属性に基づいてタイブレークルールを設定できます。各属性に優先順位値を割り当てることができます。タイブレークルールは、各属性に割り当てられた優先順位に基づいて適用されます。割り当てられる優先順位値が小さくなると、タイブレーク属性の優先順位が高くなります。

手順

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

	コマンドまたはアクション	目的
ステップ 4	**address-family ipv4 autonomous-system** *autonomous-system-number* 例： `Device(config-router)# address-family ipv4 autonomous-system 1`	IPv4 VRF アドレス ファミリ コンフィギュレーション モードを開始して、EIGRP ルーティングインスタンスを設定します。
ステップ 5	**topology base** 例： `Device(config-router-af)# topology base`	基本 EIGRP トポロジを設定し、ルータアドレス 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 6	**fast-reroute tie-break {interface-disjoint | linecard-disjoint | lowest-backup-path-metric | srlg-disjoint}** *priority-number* 例： `Device(config-router-af-topology)# fast-reroute tie-break lowest-backup-path-metric 2`	タイブレーク属性を設定し、その属性に優先順位を割り当てることにより、EIGRP が LFA を選択することを可能にします。 • 1 つのアドレスファミリで属性を複数回設定することはできません。
ステップ 7	**end** 例： `Device(config-router-af-topology)# end`	ルータ 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て、特権 EXEC モードに戻ります。
ステップ 8	**show ip eigrp topology frr** 例： `Device# show ip eigrp topology frr`	EIGRP トポロジテーブルで設定されているフィージブルサクセサまたは LFA のリストを表示します。

EIGRP ループフリー代替 IP Fast Reroute の設定例

例：プレフィックスごとの LFA IP FRR の設定

次に、map1 という名前のルートマップによって指定されたプレフィックスに関して EIGRP LFA IPFRR を設定する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# router eigrp name
Device(config-router)# address-family ipv4 autonomous-system 1
Device(config-router-af)# topology base
Device(config-router-af-topology)# fast-reroute per-prefix route-map map1
Device(config-router-af-topology)# end
```

例：プレフィックス間のロードシェアリングの無効化

次に、プレフィックス間のロードシェアリングを無効にする例を示します。

```
Device> enable
Device# configure terminal
Device(config)# router eigrp name
Device(config-router)# address-family ipv4 autonomous-system 1
Device(config-router-af)# topology base
Device(config-router-af-topology)# fast-reroute load-sharing disable
Device(config-router-af-topology)# end
```

例：タイブレークルールの有効化

次に、タイブレーク設定を有効にして、特定のプライマリパスに複数の候補LFAがある場合にEIGRPがLFAを選択できるようにする例を示します。次に、発信インターフェイスをプライマリパスと共有するLFAを排除するタイブレークルールを有効にする例を示します。

```
Device> enable
Device# configure terminal
Device(config)# router eigrp name
Device(config-router)# address-family ipv4 autonomous-system 1
Device(config-router-af)# topology base
Device(config-router-af-topology)# fast-reroute tie-break interface-disjoint 2
Device(config-router-af-topology)# end
```

次に、ラインカードをプライマリパスと共有するLFAを排除するタイブレークルールを有効にする例を示します。

```
Device> enable
Device# configure terminal
Device(config)# router eigrp name
Device(config-router)# address-family ipv4 autonomous-system 1
Device(config-router-af)# topology base
Device(config-router-af-topology)# fast-reroute tie-break linecard-disjoint 3
Device(config-router-af-topology)# end
```

次に、保護されたプレフィックスに対して最も低いメトリックを持つLFAを選択するタイブレークルールを有効にする例を示します。

```
Device> enable
Device# configure terminal
Device(config)# router eigrp name
Device(config-router)# address-family ipv4 autonomous-system 1
Device(config-router-af)# topology base
Device(config-router-af-topology)# fast-reroute tie-break lowest-backup-path-metric 4
Device(config-router-af-topology)# end
```

次に、SRLGをプライマリパスと共有するLFAを排除するタイブレークルールを有効にする例を示します。

```
Device> enable
Device# configure terminal
Device(config)# router eigrp name
Device(config-router)# address-family ipv4 autonomous-system 1
Device(config-router-af)# topology base
```

```
Device(config-router-af-topology)# fast-reroute tie-break srlg-disjoint 1
Device(config-router-af-topology)# end
```

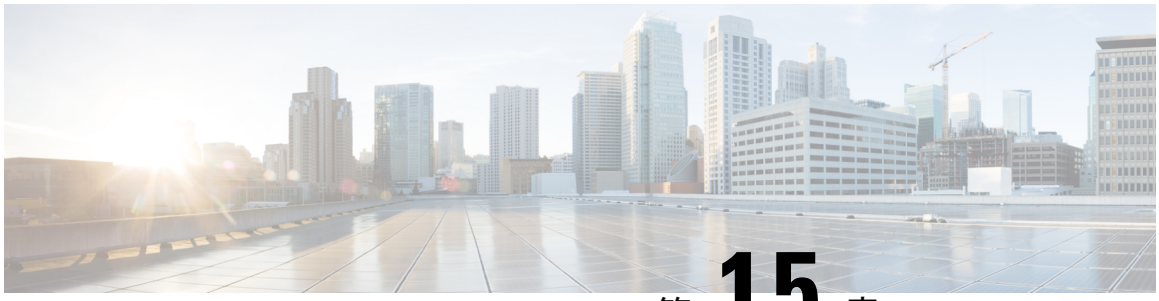
EIGRP ループフリー代替 IP Fast Reroute の機能情報

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

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

表 *26 : EIGRP* ループフリー代替 *IP Fast Reroute* の機能情報

機能名	リリース	機能情報
EIGRP ループフリー代替 IP Fast Reroute（IPFRR）	Cisco IOS XE Amsterdam 17.3.1	EIGRP ループフリー代替 IP Fast Reroute 機能により、EIGRP は、修復パスまたはバックアップルートを事前に計算し、これらのパスまたはルートを RIB にインストールすることで、ルーティングの遷移時間を 50 ミリ秒未満に短縮できます。EIGRP ネットワークでは、事前に計算されたバックアップルートは、フィージブルサクセサまたは LFA と呼ばれます。



第 15 章

BFD-EIGRP サポートの設定

• BFD-EIGRP サポート （205 ページ）

BFD-EIGRP サポート

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

BFD-EIGRP サポートの前提条件

- Enhanced Interior Gateway Routing Protocol（EIGRP）は、関連するすべての参加ルータで実行する必要があります。
- Bidirectional Forwarding Detection（BFD）セッションを BFD ネイバーに対して実行するインターフェイスで、**bfd** コマンドを使用して BFD セッションの基本パラメータを設定する必要があります。

BFD-EIGRP サポートに関する情報

BFD-EIGRP サポートの概要

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

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

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

BFD-EIGRP サポートの設定方法

BFD-EIGRP サポートの設定方法

BFD-EIGRP サポートの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router eigrp** *as-number* 例： `Device(config)#router eigrp 123`	EIGRP ルーティング プロセスを設定し、ルータ コンフィギュレーション モードを開始します。

ステップ 4	次のいずれかを実行します。
	• **bfd all-interfaces**
	• **bfd interface** *type number*
	例：
	`Device(config-router)#bfd all-interfaces`
	例：
	`Device(config-router)#bfd interface FastEthernet 6/0`
		EIGRP ルーティング プロセスに関連付けられたすべてのインターフェイスで、BFD をグローバルにイネーブルにします。
		または
		EIGRP ルーティング プロセスに関連付けられた 1 つ以上のインターフェイスに対して、インターフェイスごとに BFD をイネーブルにします。

ステップ 5	**end** 例： `Device(config-router)#end`	ルータ コンフィギュレーション モードを終了して、特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
ステップ 6	**show bfd neighbors** [**details**] 例： `Device#show bfd neighbors details`	（任意）BFD ネイバーがアクティブで、BFD が登録したルーティング プロトコルが表示されることを確認します。
ステップ 7	**show ip eigrp interfaces** [*type number*] [*as-number*] [**detail**] 例： `Device#show ip eigrp interfaces detail`	（任意）EIGRP に対する BFD サポートがイネーブルになっているインターフェイスを表示します。

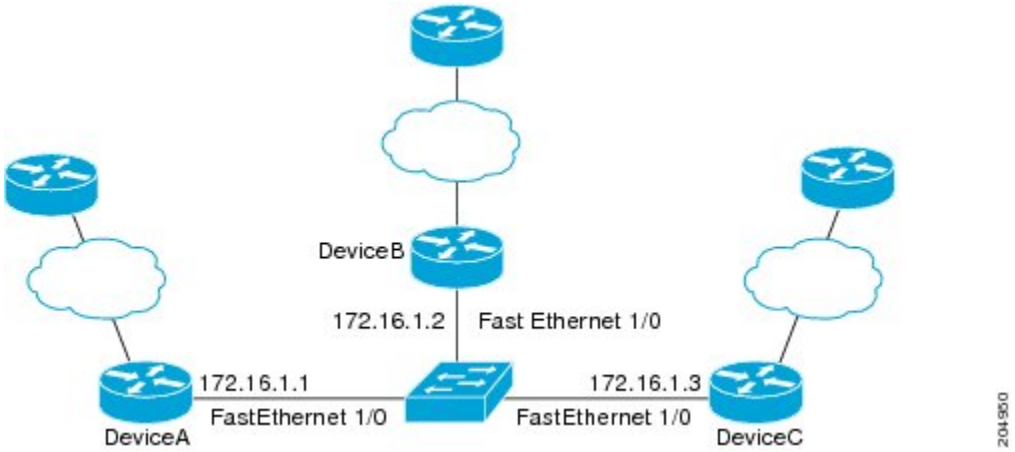
BFD-EIGRP サポートの設定例

例：エコー モードがデフォルトでイネーブルになった EIGRP ネットワークでの BFD の設定

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

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

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



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

デバイス A の設定

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

デバイス B の設定

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

デバイス C の設定

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

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

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

```
DeviceA# show bfd neighbors details

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

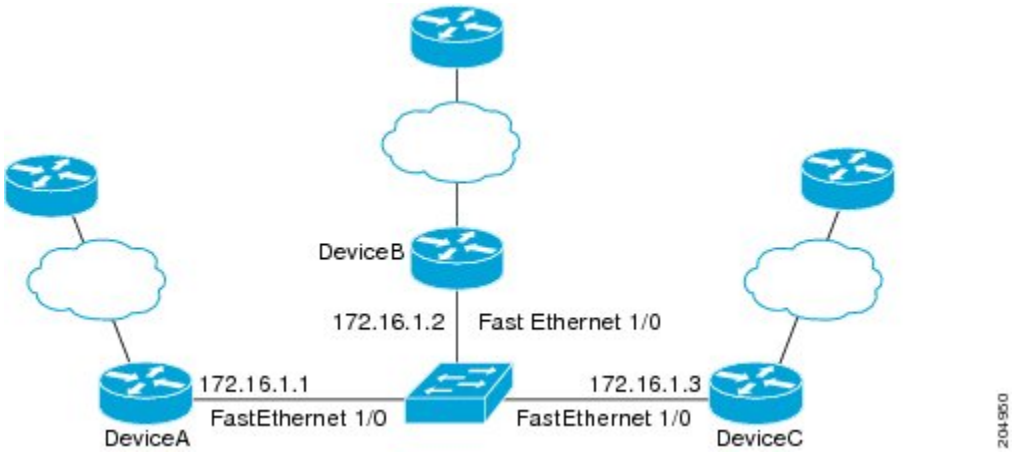
```
    6/1    Up        0    (3 )   Up        Fa1/0
Session state is UP and using echo function with 50 ms interval.
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 1000000, MinRxInt: 1000000, Multiplier: 3
Received MinRxInt: 1000000, Received Multiplier: 3
Holdown (hits): 3000(0), Hello (hits):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

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

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

```
DeviceB# show bfd neighbors details

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

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

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

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



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

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

```
DeviceA# show bfd neighbors
OurAddr       NeighAddr

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

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

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

```
DeviceC# show bfd neighbors
```

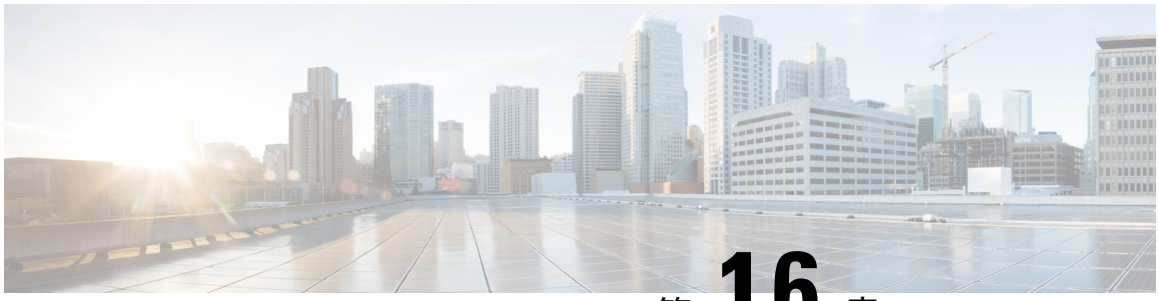
```
OurAddr        NeighAddr

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

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

BFD-EIGRP サポートの機能情報

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

表 *27 : BFD-EIGRP* サポートの機能情報

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



第 16 章

EIGRP IPv6 に対する BFD サポートの設定

- EIGRP IPv6 に対する BFD サポートの前提条件 （215 ページ）
- EIGRP IPv6 に対する BFD サポートに関する制約事項 （215 ページ）
- EIGRP IPv6 に対する BFD サポートに関する情報 （216 ページ）
- EIGRP IPv6 に対する BFD サポートの設定方法 （216 ページ）
- EIGRP IPv6 に対する BFD サポートの設定例 （220 ページ）
- EIGRP IPv6 に対する BFD サポートの機能情報 （221 ページ）
- その他の参考資料 （221 ページ）

EIGRP IPv6 に対する BFD サポートの前提条件

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

EIGRP IPv6 に対する BFD サポートに関する制約事項

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

EIGRP IPv6 に対する BFD サポートに関する情報

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

EIGRP IPv6 に対する BFD サポートの設定方法

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

すべてのインターフェイスでの BFD サポートの設定

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

手順

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

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

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

インターフェイスでの BFD サポートの設定

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

手順

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

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

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

EIGRP IPv6 に対する BFD サポートの設定例

ここでは、EIGRP に対する BFD サポートの設定例を示します。

例：すべてのインターフェイスでの BFD サポートの設定

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

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

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

Max Nbrs: 0, Current Nbrs: 0

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

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

```
Device# show bfd neighbors

IPv6 Sessions
```

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

例：インターフェイスでの BFD サポートの設定

次に、インターフェイスで BFD サポートを設定する例を示します。

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

EIGRP IPv6 に対する BFD サポートの機能情報

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

表 *28 : EIGRP IPv6* に対する *BFD* サポートの機能情報

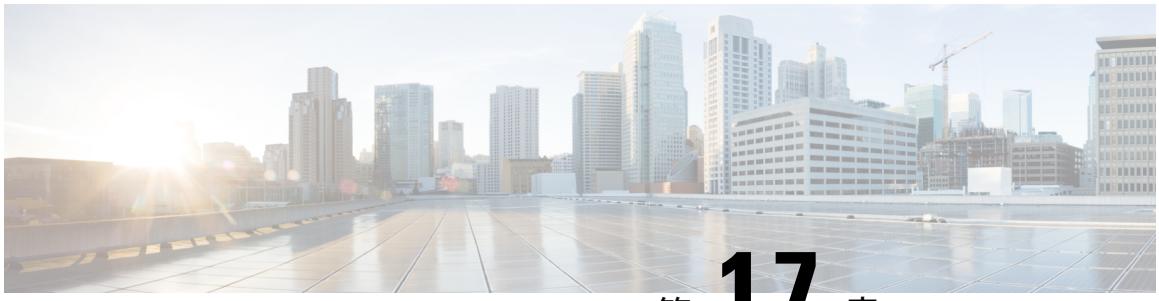
機能名	リリース	機能情報
EIGRP IPv6 に対する BFD サポート	Cisco IOS XE Gibraltar 16.11.x	この機能が導入されました。

その他の参考資料

関連資料

関連項目	マニュアル タイトル
BFD コマンド：コマンド構文、コマンドモード、コマンド履歴、デフォルト、使用に関する注意事項、および例。	次のドキュメントの IP ルーティングに関する項を参照してください：*Command Reference (Catalyst 9400 Series Switches)*

関連項目	マニュアル タイトル
EIGRP コマンド：コマンド構文の詳細、コマンドモード、コマンド履歴、デフォルト設定、使用に関する注意事項、および例	次のドキュメントの IP ルーティングに関する項を参照してください：*Command Reference (Catalyst 9400 Series Switches)*
EIGRP の設定	次のドキュメントのルーティングに関する項を参照してください：*Software Configuration Guide (Catalyst 9400 Series Switches)*



第 17 章

BFD- スタティックルートサポートの設定

- BFD- スタティックルートサポートの前提条件 （223 ページ）
- BFD- スタティックルートサポートに関する制約事項 （223 ページ）
- BFD- スタティックルートサポートに関する情報 （224 ページ）
- BFD- スタティックルートサポートの設定方法 （225 ページ）
- BFD- スタティックルートサポートの設定例 （226 ページ）
- BFD- スタティックルートサポートの機能情報 （227 ページ）

BFD- スタティックルートサポートの前提条件

- シスコ エクスプレス フォワーディングおよび IP ルーティングが、関連するすべてのデバイスでイネーブルになっていること。
- Bidirectional Forwarding Detection（BFD）セッションを BFD ネイバーに対して実行するインターフェイスで、BFD 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- スタティックルートサポートに関する制約事項

- 仮想テンプレートおよびダイヤラインターフェイスで BFD の設定はソフトウェアによって誤って許可される可能性がありますが、仮想テンプレートおよびダイヤラインターフェイスで BFD 機能はサポートされません。仮想テンプレートおよびダイヤラ インターフェイスで BFD を設定しないでください。
- BFD は直接接続されたネイバーだけに対して動作します。BFD のネイバーは 1 ホップ以内に限られます。マルチホップのコンフィギュレーションはサポートされません。

BFD- スタティックルートサポートに関する情報

BFD- スタティックルートサポートの概要

BFD- スタティックルートサポート機能を使用すると、設定済みの BFD セッションを使用してスタティックルートの到達可能性をモニタするために、スタティックルートをスタティック Bidirectional Forwarding Detection（BFD）設定に関連付けることができます。BFD セッションのステータスに応じて、スタティックルートがルーティング情報ベース（RIB）に追加または RIB から削除されます。

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

BFD セッションが正常に確立されるように、ピア上のインターフェイスで BFD を設定し、ピア上の BFD クライアントに BFD ネイバーのアドレスを登録する必要があります。インターフェイスがダイナミック ルーティング プロトコルで使用される場合、後者の要件は通常、BFD の各ネイバーでルーティング プロトコル インスタンスを設定することによって満たされます。インターフェイスがスタティック ルーティングに排他的に使用される場合、この要件はピア上でスタティック ルートを設定することによって満たす必要があります。

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

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

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

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

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

BFD- スタティックルートサポートの設定方法

BFD-EIGRP サポートの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router eigrp** *as-number* 例： `Device(config)#router eigrp 123`	EIGRP ルーティング プロセスを設定し、ルータ コンフィギュレーション モードを開始します。
ステップ 4	次のいずれかを実行します。 • **bfd all-interfaces** • **bfd interface** *type number* 例： `Device(config-router)#bfd all-interfaces` 例：	EIGRP ルーティング プロセスに関連付けられたすべてのインターフェイスで、BFD をグローバルにイネーブルにします。 または EIGRP ルーティング プロセスに関連付けられた 1 つ以上のインターフェイスに

	コマンドまたはアクション	目的
	`Device(config-router)#bfd interface FastEthernet 6/0`	対して、インターフェイスごとに BFD をイネーブルにします。
ステップ 5	**end** 例： `Device(config-router)#end`	ルータ コンフィギュレーション モードを終了して、特権 EXEC モードに戻ります。
ステップ 6	**show bfd neighbors** [**details**] 例： `Device#show bfd neighbors details`	（任意）BFD ネイバーがアクティブで、BFD が登録したルーティング プロトコルが表示されることを確認します。
ステップ 7	**show ip eigrp interfaces** [*type number*] [*as-number*] [**detail**] 例： `Device#show ip eigrp interfaces detail`	（任意）EIGRP に対する BFD サポートがイネーブルになっているインターフェイスを表示します。

BFD- スタティックルートサポートの設定例

例：BFD- スタティックルートサポートの設定

次の例では、ネットワークはデバイス A とデバイス B で構成されています。デバイス A のシリアル インターフェイス 2/0 は、デバイス B のシリアル インターフェイス 2/0 と同じネットワークに接続されています。BFD セッションを起動するには、デバイス B を設定する必要があります。

デバイス A

```
configure terminal
interface Serial 2/0
ip address 10.201.201.1 255.255.255.0
bfd interval 500 min_rx 500 multiplier 5
ip route static bfd Serial 2/0 10.201.201.2
ip route 10.0.0.0 255.0.0.0 Serial 2/0 10.201.201.2
```

デバイス B

```
configure terminal
interface Serial 2/0
ip address 10.201.201.2 255.255.255.0
bfd interval 500 min_rx 500 multiplier 5
ip route static bfd Serial 2/0 10.201.201.1
ip route 10.1.1.1 255.255.255.255 Serial 2/0 10.201.201.1
```

デバイス B のスタティック ルートが単独で存在していて、10.201.201.1 と 10.201.201.2 の間で BFDセッションをイネーブルにすることに注意してください。設定する必要のある有益なスタティック ルートがない場合、パケットの転送に影響しないプレフィックス、たとえば、ローカルで設定されたループバック インターフェイスを選択します。

次の例では、BFD グループ testgroup のイーサネット インターフェイス 0/0 を介して 209.165.200.225 に到達するアクティブなスタティック BFD 設定があります。設定されたスタティック BFD によってトラッキングされるスタティック ルートが設定されるとすぐに、単一のホップ BFD セッションがイーサネット インターフェイス 0/0 を介して 209.165.200.225 に開始されます。BFD セッションが正常に確立されると、プレフィックス 10.0.0.0/8 が RIB に追加されます。

```
configure terminal
ip route static bfd Ethernet 0/0 209.165.200.225 group testgroup
ip route 10.0.0.0 255.255.255.224 Ethernet 0/0 209.165.200.225
```

次の例では、イーサネット インターフェイス 0/0.1001 を介した 209.165.200.226 への BFD セッションがグループ testgroup を使用するようにマークされます。つまり、この設定はパッシブなスタティック BFD です。2 つ目のスタティック BFD 設定によってトラッキングされるスタティック ルートがあるものの、209.165.200.226 に対する BFD セッションはイーサネット インターフェイス 0/0.1001 を介しては開始されません。プレフィックス 10.1.1.1/8 と 10.2.2.2/8 の存在は、アクティブなスタティック BFD セッション（イーサネット インターフェイス 0/0 209.165.200.225）によって制御されます。

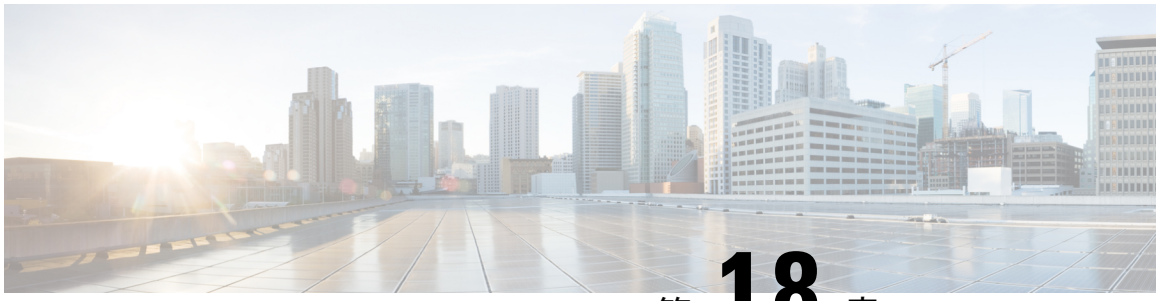
```
configure terminal
ip route static bfd Ethernet 0/0 209.165.200.225 group testgroup
ip route 10.0.0.0 255.255.255.224 Ethernet 0/0 209.165.200.225
ip route static bfd Ethernet 0/0.1001 209.165.200.226 group testgroup passive
ip route 10.1.1.1 255.255.255.224 Ethernet 0/0.1001 209.165.200.226
ip route 10.2.2.2 255.255.255.224 Ethernet 0/0.1001 209.165.200.226
```

BFD- スタティックルートサポートの機能情報

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

表 *29 : BFD-* スタティックルートサポートの機能情報

機能名	リリース	機能情報
BFD：スタティック ルート サポート	Cisco IOS XE Everest 16.6.2	BFD- スタティックルートサポート機能を使用すると、設定済みの BFD セッションを使用してスタティックルートの到達可能性をモニタするために、スタティックルートをスタティック Bidirectional Forwarding Detection（BFD）設定に関連付けることができます。BFD セッションのステータスに応じて、スタティックルートがルーティング情報ベース（RIB）に追加または RIB から削除されます。 この機能は、Cisco IOS XE Everest 16.6.2 で、Cisco Catalyst 9400 シリーズ スイッチに実装されました。



第 18 章

BFD-VRF サポートの設定

- BFD-VRF サポートの前提条件 （229 ページ）
- BFD-VRF サポートに関する情報 （229 ページ）
- BFD：VRF サポートの機能情報 （230 ページ）

BFD-VRF サポートの前提条件

すべての Bidirectional Forwarding Detection（BFD）クライアントは、Virtual Route Forwarding（VRF）に対応している必要があります。

BFD-VRF サポートに関する情報

BFD-VRF サポートの概要

BFD-VRF サポート機能により、プロバイダーエッジ（PE）デバイスおよびカスタマーエッジ（CE）デバイス上の Virtual Route Forwarding（VRF）に対する Bidirectional Forwarding Detection（BFD）サポートが有効になり、デバイス間のルーティングプロトコル障害を迅速に検出されます。

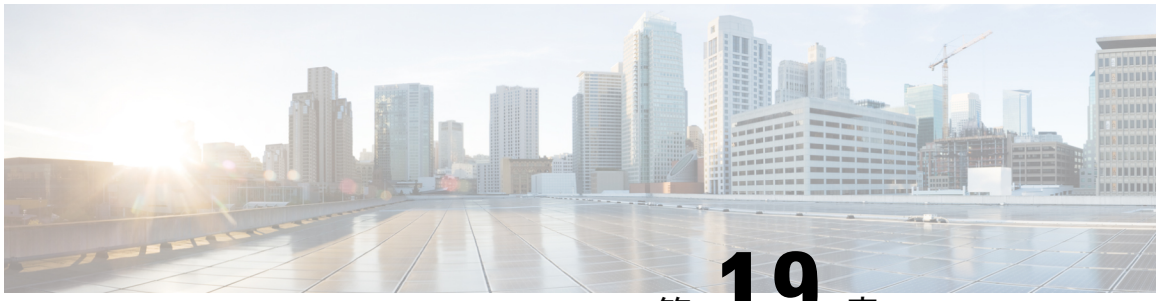
BFD クライアントは、セッションモニタリングを要求する前に、BFD が設定されているデバイスとのバーチャル プライベート ネットワーク（VPN）セッションを確立します。ただし、BFD ネイバーが同じ VPN セッションに接続されているか別の VPN セッションに接続されているかを判断するためのルートルックアップはありません。BFD は、クライアントに依存して VPN セッションに関する情報を取得し、関連するネイバーデバイスをモニタします。VPN セッションに関するすべての情報を使用して、シスコ エクスプレス フォワーディング（CEF）を介して BFD 制御パケットが適切な VPN に転送されます。

BFD：VRF サポートの機能情報

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

表 *30 : BFD-VRF* サポートの機能情報

機能名	リリース	機能情報
BFD-VRF サポート	Cisco IOS XE Everest 16.6.2	BFD-VRF サポート機能により、PE デバイスおよび CE デバイス上の VRF に対する BFD サポートが有効になり、デバイス間のルーティングプロトコル障害が迅速に検出されます。 この機能は、Cisco IOS XE Everest 16.6.2 で、Cisco Catalyst 9400 シリーズ スイッチに実装されました。



第 19 章

BFD IPv6 カプセル化サポートの設定

• BFD IPv6 カプセル化サポート （231 ページ）

BFD IPv6 カプセル化サポート

Bidirectional Forwarding Detection（BFD）for IPv6 カプセル化は、セッション情報構造内に記述されます。これらのセッション情報構造は、サポートされているプロトコルに対して BFDv6 によって定義されます。BFDv6 は、セッション情報構造の情報を使用して、そのセッション上の BFDv6 パケットに対する正しいカプセル化を決定します。

BFD IPv6 カプセル化サポートの前提条件

- Bidirectional Forwarding Detection over IPv6（BFDv6）を使用している場合、参加するすべてのルータ上で IPv6 シスコ エクスプレス フォワーディングおよび IPv6 ユニキャストルーティングが有効になっている必要があります。
- BFD IPv6 ソフトウェアセッションを設定する際は、次の CLI コマンドを設定する必要があります。

  no ipv6 nd nud igp

BFD IPv6 カプセル化サポートに関する制約事項

- グローバル IPv6 アドレスがインターフェイス上で設定されている場合、BFDv6 はグローバル IPv6 ネイバー アドレスだけをサポートします。
- 非同期モードのみがサポートされます。非同期モードでは、どちらの BFDv6 ピアも BFDv6 セッションを開始できます。

BFD IPv6 カプセル化サポートに関する情報

BFDv6 プロトコルの概要

ここでは、BFDv6 プロトコル、IPv4 用の BFD との違い、および IPv4 用の BFD との協調動作について説明します。BFD はあらゆるメディア タイプ、カプセル化、トポロジ、およびルーティング プロトコルの高速転送パス障害検出回数を提供するように設計された検出プロトコルです。高速転送パス障害検出に加えて、BFDはネットワーク管理者に整合性のある障害検出方法を提供します。BFDv6 は、IPv6 アドレスに対応することで IPv6 サポートを提供します。また、BFDv6 セッションを作成する機能も提供します。

BFDv6 登録

BFD クライアントは、レジストリ アプリケーション プログラム インターフェイス（API）を使用して BFD に登録します。レジストリ引数には、プロトコル タイプ、監視するルートのアドレスとインターフェイス記述ブロック（IDB）などがあります。これらの API と引数は、BFD によってすべて IPv4 であると仮定されます。

BFDv6 には、これらの引数を削除したレジストリがあります。プロトコルおよびカプセル化は、セッション情報構造内に記述されます。これらのセッション情報構造は、サポートされているプロトコルに対して BFDv6 によって定義されます。BFDv6 は、セッション情報構造の情報を使用して、そのセッション上の BFDv6 パケットに対する正しいカプセル化を決定します。

BFDv6 のグローバルおよびリンクローカル アドレス

BFDv6 では、ネイバーの作成に、グローバルとリンクローカルの両方のアドレスがサポートされています。BFDv6 セッションでは、ネイバーのアドレス タイプと一致するように送信元アドレスが選択されます（たとえば、グローバル IPv6 アドレスのネイバーはグローバル IPv6 送信元アドレスと、リンクローカル IPv6 アドレスのネイバーはリンクローカル IPv6 送信元アドレスとペアになる必要があります）。次の表に、BFDv6 でサポートされるアドレスのペアを示します。

表 *31 :* ネイバー作成のための *BFDv6* アドレスのペア

Source Address	Destination Address	Status
グローバル	グローバル	サポート対象
グローバル	リンク ローカル	サポート対象外
リンク ローカル	グローバル	サポート対象外
リンク ローカル	リンク ローカル	サポート対象

すべての IPv6 対応インターフェイスにはリンクローカル アドレスがあり、BFDv6 によって送信元アドレスが選択されるため、常にリンクローカル アドレス ネイバーがリンクローカル インターフェイス アドレスとペアになります。グローバル宛先アドレスとリンクローカル送信元アドレスの組み合わせは、シスコ エクスプレス フォワーディングではサポートされていません。そのため、グローバル アドレス ネイバーとのセッションを BFDv6 で確立するには、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。BFDv6 では、ネイバー アドレスがグローバルなのに、グローバル アドレスがインターフェイス上に設定されていないセッションは、すべて拒否されます。

（注） BFDv6 での一意のローカル アドレス（ULA）の動作は、グローバル アドレスと同じです。

同じインターフェイス上での IPv4 用と IPv6 用の BFD

BFD では、インターフェイスごとに複数の IPv4 および IPv6 セッションがサポートされます。これらのセッションのプロトコルに制約はありません。

BFD IPv6 カプセル化サポートの設定方法

インターフェイスの基本 BFD セッション パラメータの設定

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、次の作業を繰り返します。

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	**interface** *type number* 例： `Device(config)#interface GigabitEthernet 0/0/0`	インターフェイスのタイプと番号を指定し、デバイスをインターフェイス コンフィギュレーション モードにします。

	コマンドまたはアクション	目的
ステップ 4	**bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier* 例： `Device(config-if)#bfd interval 50 min_rx 50 multiplier 5`	インターフェイスで BFD をイネーブルにします。

BFD IPv6 カプセル化サポートの設定例

例：インターフェイスでの BFD セッション パラメータの設定

```
Device#show ipv6 ospf neighbor detail

 Neighbor 172.16.4.4
    In the area 0 via interface POS4/0
    Neighbor: interface-id 14, link-local address FE80::205:5FFF:FED3:5406
    Neighbor priority is 1, State is FULL, 6 state changes
    Options is 0x63AD1B0D
    Dead timer due in 00:00:33
    Neighbor is up for 00:48:56
    Index 1/1/1, retransmission queue length 0, number of retransmission 1
    First 0x0(0)/0x0(0)/0x0(0) Next 0x0(0)/0x0(0)/0x0(0)
    Last retransmission scan length is 1, maximum is 1
    Last retransmission scan time is 0 msec, maximum is 0 msec
 Neighbor 172.16.3.3
    In the area 1 via interface FastEthernet0/0
    Neighbor: interface-id 3, link-local address FE80::205:5FFF:FED3:5808
    Neighbor priority is 1, State is FULL, 6 state changes
    DR is 172.16.6.6 BDR is 172.16.3.3
    Options is 0x63F813E9
    Dead timer due in 00:00:33
    Neighbor is up for 00:09:00
    Index 1/1/2, retransmission queue length 0, number of retransmission 2
    First 0x0(0)/0x0(0)/0x0(0) Next 0x0(0)/0x0(0)/0x0(0)
    Last retransmission scan length is 1, maximum is 2
    Last retransmission scan time is 0 msec, maximum is 0 msec
 Neighbor 172.16.5.5
    In the area 2 via interface ATM3/0
    Neighbor: interface-id 13, link-local address FE80::205:5FFF:FED3:6006
    Neighbor priority is 1, State is FULL, 6 state changes
    Options is 0x63F7D249
    Dead timer due in 00:00:38
    Neighbor is up for 00:10:01
    Index 1/1/3, retransmission queue length 0, number of retransmission 0
    First 0x0(0)/0x0(0)/0x0(0) Next 0x0(0)/0x0(0)/0x0(0)
    Last retransmission scan length is 0, maximum is 0
    Last retransmission scan time is 0 msec, maximum is 0 msec
```

BFD IPv6 カプセル化サポートに関するその他の参考資料

標準および RFC

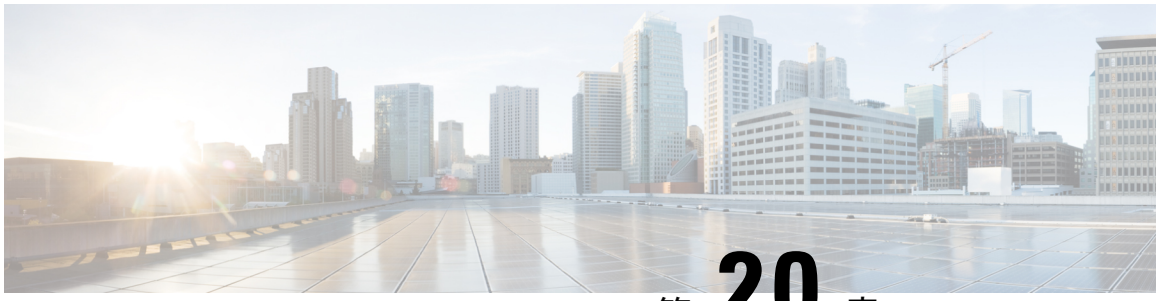
標準/RFC	タイトル
IPv6 に関する RFC	IPv6 RFCs

BFD IPv6 カプセル化サポートの機能情報

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

表 *32 : BFD IPv6* カプセル化サポートの機能情報

機能名	リリース	機能情報
BFD IPv6 カプセル化サポート	Cisco IOS XE Everest 16.6.2	BFDv6 カプセル化は、セッション情報構造内に記述されます。これらのセッション情報構造は、サポートされているプロトコルに対して BFDv6 によって定義されます。BFDv6 は、セッション情報構造の情報を使用して、そのセッション上の BFDv6 パケットに対する正しいカプセル化を決定します。 この機能は、Cisco IOS XE Everest 16.6.2 で、Cisco Catalyst 9400 シリーズ スイッチに実装されました。



第 20 章

HSRP BFD ピアリングの設定

- HSRP BFD ピアリングに関する制約事項 （237 ページ）
- HSRP BFD ピアリングに関する情報 （237 ページ）
- HSRP BFD ピアリングの設定方法 （238 ページ）
- HSRP BFD ピアリングの設定例 （243 ページ）
- HSRP BFD ピアリングの機能情報 （244 ページ）

HSRP BFD ピアリングに関する制約事項

Bidirectional Forwarding Detection（BFD）に対する Hot Standby Router Protocol（HSRP）サポートは、すべてのプラットフォームおよびインターフェイスで利用できるわけではありません。

HSRP BFD ピアリングに関する情報

ここでは、HSRP BFD ピアリングの概要を示します。

HSRP の BFD ピアリング

HSRP の BFD ピアリング機能は、ホット スタンバイ ルータ プロトコル（HSRP）グループのメンバのヘルス モニタリング システムに双方向フォワーディング検出（BFD）を導入します。HSRP は、HSRP グループ メンバーのヘルス モニタリング システムの一部として BFD をサポートしています。BFD がないと、HSRP はマルチプロセス システムの 1 つのプロセスとして動作するため、hello タイマーやホールド タイマー（ミリ秒単位）を使用して大量のグループに対応できるように適切なタイミングでスケジュールされることが保証されません。BFD は疑似プリエンプティブ プロセスとして動作するため、必要なときに実行されることが保証されます。複数の HSRP グループに早期フェールオーバー通知を実行できるのは、2 台のデバイス間の 1 つの BFD セッションだけです。

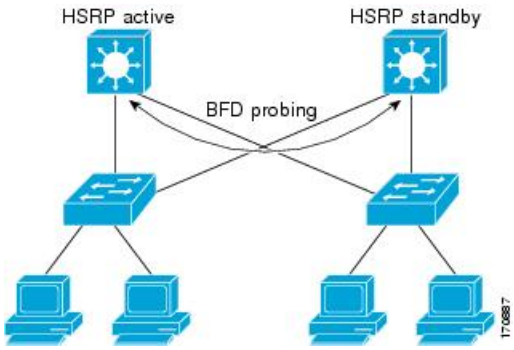
この機能は、デフォルトでイネーブルにされています。HSRP スタンバイ デバイスは、HSRP アクティブ デバイスの実際の IP アドレスを HSRP hello メッセージから検出します。また、BFD クライアントとして登録し、アクティブ デバイスが使用不能になった場合に通知するよ

うに要求します。BFD はスタンバイ デバイスとアクティブ デバイス間の接続が失敗したことを確認すると、アクティブ デバイスとしてすぐに引き継ぐスタンバイ デバイス上の HSRP に通知します。

BFDは、インターフェイス、データ リンク、および転送プレーンを含む、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。BFD はインターフェイス レベルおよびルーティング プロトコル レベルでイネーブルにする検出プロトコルです。シスコでは BFD 非同期モードをサポートしています。これは、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 がインターフェイスでイネーブルになっているとともに、HSRP 用にデバイス レベルでイネーブルになっている場合、BFD セッションが作成されて、BFD タイマーがネゴシエートされ、ネゴシエートされた間隔で BFD ピアが互いに BFD 制御パケットの送信を開始します。

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 Border Gateway Protocol（BGP）、Enhanced Interior Gateway Routing Protocol（EIGRP）、Hot Standby Router Protocol（HSRP）、Intermediate System to Intermediate System（IS-IS）、Open Shortest Path First（OSPF）などのルーティング プロトコルとは関係なく、BFD ピアの障害検出時間を短縮します。ローカル デバイスのルーティング プロトコルに高速障害検出通知を送信して、ルーティング テーブル再計算プロセスを開始すると、BFD はネットワーク コンバージェンス時間全体を大幅に短縮できます。下の図は、HSRP と BFD を実行する 2 台のデバイスがある単純なネットワークを示しています。

図 *7: HSRP の BFD ピアリング*



HSRP BFD ピアリングの設定方法

ここでは、HSRP BFD ピアリングの設定について説明します。

インターフェイスでの BFD セッション パラメータの設定

ここでは、Bidirectional Forwarding Detection（BFD）セッションのベースライン パラメータをインターフェイスで設定して、インターフェイスで BFD を設定する作業を行います。BFD ネ

イバーに対して BFD セッションを実行するインターフェイスごとに、この手順を繰り返します。

手順

	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	**bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier* 例： `Device(config-if)# bfd interval 50 min_rx 50 multiplier 5`	インターフェイスで BFD をイネーブルにします。
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 モードを終了します。

HSRP BFD ピアリングの設定

ここでは、Hot Standby Router Protocol（HSRP）Bidirectional Forwarding Detection（BFD）ピアリングをイネーブルにする作業を行います。この作業のステップは、HSRP ピアに BFD セッションを実行する各インターフェイスで行ってください。

HSRP はデフォルトで BFD ピアリングをサポートしています。HSRP BFD ピアリングがディセーブルになっている場合、デバイス レベルで再度イネーブルにして、すべてのインターフェイスの BFD サポートをまとめてイネーブル化したり、インターフェイス レベルでインターフェイスごとに再度イネーブルにしたりすることができます。

HSRP BFD ピアリングの設定

始める前に

この作業を進める前に

- HSRP 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。
- シスコ エクスプレス フォワーディングをイネーブルにする必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip cef** [**distributed**] 例： `Device(config)# ip cef`	シスコ エクスプレス フォワーディングまたは分散型シスコ エクスプレス フォワーディングをイネーブルにします。
ステップ 4	**interface** *type number* 例： `Device(config)# interface FastEthernet 6/0`	インターフェイス コンフィギュレーション モードを開始します。
ステップ 5	**ip address** *ip-address mask* 例： `Device(config-if)# ip address 10.0.0.11 255.255.255.0`	インターフェイスにIPアドレスを設定します。
ステップ 6	**standby** [*group-number*] **ip** [*ip-address* [**secondary**]] 例： `Device(config-if)# standby 1 ip 10.0.0.11`	HSRP をアクティブにします。
ステップ 7	**standby bfd** 例： `Device(config-if)# standby bfd`	（任意）インターフェイスでBFDに対する HSRP をイネーブルにします。

	コマンドまたはアクション	目的
ステップ 8	**exit** 例： `Device(config-if)# exit`	インターフェイス コンフィギュレーション モードを終了します。
ステップ 9	**standby bfd all-interfaces** 例： `Device(config)# standby bfd all-interfaces`	（任意）すべてのインターフェイスで BFD に対する HSRP をイネーブルにします。
ステップ 10	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了します。
ステップ 11	**show standby** [**neighbors**] 例： `Device# show standby neighbors`	（任意）BFD に対する HSRP サポートについての情報を表示します。

HSRP BFD ピアリングの検証

Hot Standby Router Protocol（HSRP）Bidirectional Forwarding Detection（BFD）ピアリングを確認するには、次のオプション コマンドを使用します。

手順

ステップ 1 **show standby**

show standby コマンドを実行すると、HSRP 情報が表示されます。

例：

```
Device# show standby

FastEthernet2/0 - Group 1
  State is Active
    2 state changes, last state change 00:08:06
  Virtual IP address is 10.0.0.11
  Active virtual MAC address is 0000.0c07.ac01
    Local virtual MAC address is 0000.0c07.ac01 (v1 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 2.772 secs
  Preemption enabled
  Active router is local
  Standby router is 10.0.0.2, priority 90 (expires in 8.268 sec)
    BFD enabled !
```

```
  Priority 110 (configured 110)
    Group name is "hsrp-Fa2/0-1" (default)
```

ステップ 2 **show standby brief**

show standby brief コマンドを実行すると、HSRP スタンバイデバイス情報が簡潔に表示されます。

例：

```
Device# show standby brief

Interface   Grp  Pri P State   Active  Standby                     Virtual IP

Et0/0       4    120 P Active  local   172.24.1.2                  172.24.1.254
Et1/0       6    120 P Active  local   FE80::A8BB:CCFF:FE00:3401   FE80::5:73FF:FEA0:6
```

ステップ 3 **show standby neighbors** [*type number*]

show standby neighbors コマンドを実行すると、インターフェイスの HSRP ピアデバイスに関する情報が表示されます。

例：

```
Device1# show standby neighbors

HSRP neighbors on FastEthernet2/0
    10.1.0.22
    No active groups
    Standby groups: 1
    BFD enabled !

Device2# show standby neighbors

HSRP neighbors on FastEthernet2/0
    10.0.0.2
    Active groups: 1
    No standby groups
    BFD enabled !
```

ステップ 4 **show bfd neighbors**

show bfd neighbors コマンドを実行すると、現在の双方向フォワーディング検出（BFD）の隣接関係が 1 行ずつ一覧表示されます。

例：

```
Device# show bfd neighbors

IPv6 Sessions

NeighAddr                              LD/RD         RH/RS     State     Int

FE80::A8BB:CCFF:FE00:3401               4/3          Up        Up        Et1/0
FE80::A8BB:CCFF:FE00:3401               4/3          Up        Up        Et1/0
```

ステップ 5 **show bfd neighbors details**

details キーワードを使用すると、各ネイバーの BFD プロトコルのパラメータとタイマーが表示されます。

例：

```
Device# show bfd neighbors details

OurAddr       NeighAddr     LD/RD  RH/RS   Holdown(mult)  State     Int
10.0.0.2      10.0.0.1       5/0    Down      0    (0 )   Down      Fa2/0
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 1000000, MinRxInt: 1000000, Multiplier: 3
Received MinRxInt: 0, Received Multiplier: 0
Holdown (hits): 0(0), Hello (hits): 1000(55)
Rx Count: 0, Rx Interval (ms) min/max/avg: 0/0/0 last: 3314120 ms ago
Tx Count: 55, Tx Interval (ms) min/max/avg: 760/1000/872 last: 412 ms ago
Registered protocols: HSRP !
Last packet: Version: 1            - Diagnostic: 0
             State bit: AdminDown  - Demand bit: 0
             Poll bit: 0           - Final bit: 0
             Multiplier: 0         - Length: 0
             My Discr.: 0          - Your Discr.: 0
             Min tx interval: 0    - Min rx interval: 0
             Min Echo interval: 0
```

HSRP BFD ピアリングの設定例

ここでは、HSRP BFD ピアリングの設定例を示します。

例：HSRP BFD ピアリング

Hot Standby Router Protocol（HSRP）は、HSRP グループ メンバのヘルス モニタリング システムの一部として Bidirectional Forwarding Detection（BFD）をサポートします。BFD がないと、HSRP はマルチプロセス システムの 1 つのプロセスとして動作するため、ミリ秒の hello タイマーやホールドタイマーを使用して大量のグループに対応できるように適切なタイミングでスケジュールされることが保証されません。BFD は疑似プリエンプティブ プロセスとして動作するため、必要なときに実行されることが保証されます。複数の HSRP グループに早期フェールオーバー通知を実行できるのは、2 台のデバイス間の 1 つの BFD セッションだけです。

次の例では、**standby bfd** コマンドと **standby bfd all-interfaces** コマンドは表示されていません。**bfd interval** コマンドを使用して、BFD がデバイスまたはインターフェイスで設定されているときは、HSRP の BFD サポートはデフォルトでイネーブルになっています。**standby bfd** コ **standby bfd all-interfaces** コマンドは、BFD がデバイスまたはインターフェイスで手動で無効にされている場合にのみ必要です。

デバイス A

```
DeviceA(config)# ip cef
DeviceA(config)# interface FastEthernet2/0
DeviceA(config-if)#  no shutdown
DeviceA(config-if)# ip address 10.0.0.2 255.0.0.0
```

```
DeviceA(config-if)# ip router-cache cef
DeviceA(config-if)# bfd interval 200 min_rx 200 multiplier 3
DeviceA(config-if)# standby 1 ip 10.0.0.11
DeviceA(config-if)# standby 1 preempt
DeviceA(config-if)# standby 1 priority 110
DeviceA(config-if)# standby 2 ip 10.0.0.12
DeviceA(config-if)# standby 2 preempt
DeviceA(config-if)# standby 2 priority 110
```

デバイス B

```
DeviceB(config)# interface FastEthernet2/0
DeviceB(config-if)# ip address 10.1.0.22 255.255.0.0
DeviceB(config-if)# no shutdown
DeviceB(config-if)# bfd interval 200 min_rx 200 multiplier 3
DeviceB(config-if)# standby 1 ip 10.0.0.11
DeviceB(config-if)# standby 1 preempt
DeviceB(config-if)# standby 1 priority 90
DeviceB(config-if)# standby 2 ip 10.0.0.12
DeviceB(config-if)# standby 2 preempt
DeviceB(config-if)# standby 2 priority 80
```

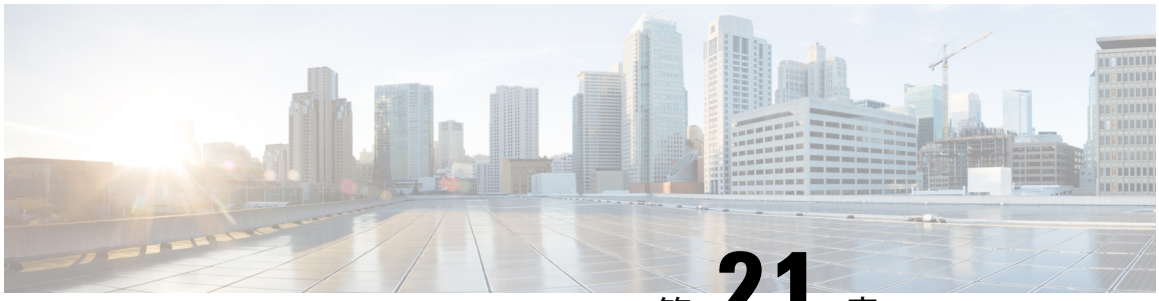
HSRP BFD ピアリングの機能情報

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

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

表 *33 : HSRP BFD* ピアリングの機能情報

機能名	リリース	機能情報
HSRP の BFD ピアリング	Cisco IOS XE Gibraltar 16.11.x	この機能が導入されました。



第 21 章

BGP ベスト外部の設定

- BGP 最良外部に関する情報 （245 ページ）
- BGP 最良外部 （246 ページ）
- BGP 最良外部機能の仕組み （248 ページ）
- BGP 最良外部を有効にするためのコンフィギュレーション モード （249 ページ）
- クラスタ間の RR での BGP ベスト外部パス （249 ページ）
- クラスタ間の RR でのベスト外部パスに関する CLI の違い （250 ページ）
- クラスタ間の RR での BGP ベスト外部パスの計算に使用されるルール （250 ページ）
- BGP 最良外部の設定方法 （251 ページ）
- BGP 最良外部の設定例 （258 ページ）
- その他の参考資料 （259 ページ）
- BGP 最良外部の機能情報 （260 ページ）

BGP 最良外部に関する情報

BGP 最良外部の概要

サービスプロバイダーはルーティングポリシーを使用し、そのルーティングポリシーにより、境界ルータは、iBGP セッションを通じて受信するパス（別の境界ルータのパス）を、プレフィックスのベストパスとして選択します。これは、ルータが eBGP 学習パスを保持する場合も同じです。この手法は一般にアクティブバックアップトポロジと呼ばれており、自律システムのプレフィックスに対し 1 つの終了または出力ポイントを定義すること、およびプライマリリンクまたは eBGP ピアリングが使用不可になった場合のバックアップとして他のポイントを使用することを目的としています。

ポリシーには利点もありますが、ポリシーにより、境界ルータは、eBGP セッションを通じて学習したパスを、自律システムから隠します。これは、そういったプレフィックスのパスをアドバタイズしないためです。この状況に対処するために、一部のルータは、ベスト外部パスと呼ばれる 1 つの外部学習パスをアドバタイズします。最良外部の動作により、次のように、BGP 選択プロセスではすべての宛先に対して 2 つのパスが選択されます。

- その宛先への既知ルートの完全セットからベスト パスが選択されます。

- その外部ピアから受信したルートのセットからベスト外部パスが選択されます。

BGP は外部ピアにベスト パスをアドバタイズします。BGP では、iBGP パスをベスト パスとして選択した場合に内部ピアからベストパスを取り消すのではなく、ベスト外部パスを内部ピアにアドバタイズします。

BGP 最良外部機能は、インターネット アクセスと MPLS VPN シナリオのプレフィックス独立コンバージェンス（PIC）エッジの必須コンポーネントであり、代替パスをアクティブバックアップ トポロジのネットワークで利用可能にします。

ベスト外部ルートとは

BGP 最良外部機能では、「ベスト外部ルート」をバックアップ パスとして使用します。これは、draft-marques-idr-best-external に基づく、外部ネイバーから受信したルートのうち最も優先されるルートです。外部ネイバーからの最優先ルートとして以下が有効です。

- 内部ボーダー ゲートウェイ プロトコル（iBGP）セッションを相互間で使用する、異なるクラスタ内の 2 つのルータ。
- 外部ボーダー ゲートウェイ プロトコル（eBGP）セッションを相互間で使用する、コンフェデレーションの異なる自律システム内の 2 つのルータ。

ベスト外部ルートは、ルーティング情報ベース（RIB）にインストールされているベストルートとは異なる場合があります。ベスト ルートが内部ルートの場合もあります。ベスト ルートに加えて、ベスト外部ルートをアドバタイズおよび保存できるようにすることで、プライマリ パスに障害が発生した場合でも、使用可能な追加のパスが用意されているため、ネットワークの接続をより迅速に復元できます。

BGP 最良外部

BGP最良外部機能を使用すると、ネットワークにバックアップ外部ルートを用意でき、プライマリ外部ルートの接続が失われるのを回避できます。BGP最良外部機能は、外部ネイバーから受信したルートのうち最も優先するルートを、バックアップ ルートとしてアドバタイズします。この機能は、アクティブバックアップ トポロジで便利です。アクティブバックアップ トポロジでは、サービス プロバイダーはルーティング ポリシーを使用し、そのルーティング ポリシーにより、境界ルータは、内部ボーダー ゲートウェイ プロトコル（iBGP）セッションを通じて受信するパス（別の境界ルータのパス）を、プレフィックスのベストパスとして選択します。これは、ルータが外部ボーダー ゲートウェイ プロトコル（eBGP）学習パスを保持する場合も同じです。このアクティブバックアップ トポロジでは、自律システムのプレフィックスに対し1つの終了または出力ポイントが定義され、プライマリ リンクまたはeBGP ピアリングが使用不可になった場合のバックアップとして他のポイントが使用されます。ポリシーにより、境界ルータは、eBGP セッションを通じて学習したパスを、自律システムから隠します。これは、そういったプレフィックスのパスをアドバタイズしないためです。この状況に対処するために、一部のデバイスは、ベスト外部パスと呼ばれる1つの外部学習パスをアドバタイズします。

BGP 最良外部の前提条件

- リンク障害を迅速に検出するために、双方向フォワーディング検出（BFD）プロトコルをイネーブルにする必要があります。
- BGP およびマルチプロトコル ラベル スイッチング（MPLS）ネットワークが稼働していて、複数のパス（マルチホーム）によりプロバイダーサイトと接続されているカスタマーサイトで実行されている必要があります。
- バックアップ パスには、ベスト パスのネクスト ホップと異なる固有のネクスト ホップがある必要があります。
- BGP では、動作するパス間のロスレス スイッチオーバーをサポートする必要があります。

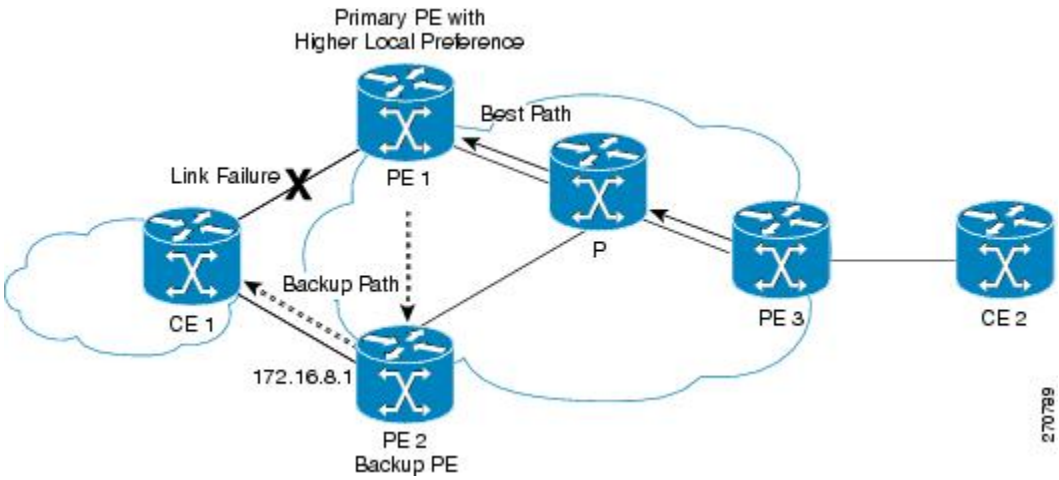
BGP 最良外部の制約事項

- BGP マルチパスがインストールされており、BGP テーブル内にマルチパスが存在する場合、BGP 最良外部機能では、バックアップ パスはインストールされません。マルチパスのいずれかが、自動的に他のパスのバックアップとして機能します。
- 次の機能では、BGP 最良外部機能はサポートされていません。
  - MPLS VPN Carrier Supporting Carrier
  - MPLS VPN 相互自律システム、オプション B
  - Virtual Routing and Forwarding（VRF）ラベル単位での MPLS VPN
- BGP 最良外部機能は、マルチキャストまたは L2VPN VRF アドレス ファミリでは設定できません。
- Cisco IOS XE Release 3.4S 以降を実行している場合を除き、BGP 最良外部機能をルート リフレクタで設定することはできません。
- BGP 最良外部機能は NSF/SSO をサポートしていません。ただし、両方のルート プロセッサで BGP 最良外部機能が設定されている場合は、ISSU がサポートされます。
- BGP 最良外部機能は、VPNv4、VPNv6、IPv4 VRF、IPv6 VRF アドレス ファミリでのみ設定できます。
- **bgp advertise-best-external** コマンドを使用して BGP 最良外部機能を設定する場合は、**bgp additional-paths install** コマンドで BGP PIC 機能を有効にする必要はありません。BGP PIC 機能は、BGP 最良外部機能によって自動的に有効化されます。
- BGP 最良外部機能を設定すると、「MPLS VPN--BGP ローカル コンバージェンス」の機能がオーバーライドされます。ただし、設定から **protection local-prefixes** コマンドを削除する必要はありません。

BGP 最良外部機能の仕組み

BGP 最良外部機能は、Internet Engineering Task Force（IETF）の draft-marques-idr-best-external.txt に基づいています。BGP 最良外部機能は、ベスト外部ルートをバックアップ ルートとして内部ピアにアドバタイズします。バックアップ ルートは RIB および Cisco Express Forwarding に保存されます。プライマリ パスに障害が発生した場合でも、BGP PIC 機能により、ベスト外部パスを代わりに使用できるため、接続をより迅速に復元できます。

図 *8 : MPLS VPN*： *MPLS VPN* エッジの最良外部



上の図は、BGP 最良外部機能を使用した MPLS VPN を示しています。このネットワークは、以下のコンポーネントで構成されています。

- プロバイダー エッジ（PE）ルータとカスタマー エッジ（CE）ルータの間に eBGP セッションが存在します。
- PE1 はプライマリ ルータで、ローカル プリファレンス設定がより高くなっています。
- CE2 からのトラフィックでは、PE1 を使用してルータ CE1 に到達します。
- PE1 には、CE1 に到達するためのパスが 2 つあります。
- CE1 は PE1 および PE2 とデュアルホーム接続されています。
- PE1 はプライマリ パスで、PE2 はバックアップ パスです。

上の図では、MPLS クラウドのトラフィックは PE1 を通過して CE1 に到達します。したがって、PE2 は、PE1 をベスト パスとして、PE2 をバックアップ パスとして使用します。

PE1 および PE2 は BGP 最良外部機能を使用して設定されています。BGP は、ベスト パス（PE1-CE1 リンク）とバックアップ パス（PE2）を計算し、両方のパスを RIB および Cisco Express Forwarding にインストールします。ベスト パスに加えて、ベスト外部パス（PE2）もピア ルータにアドバタイズされます。

Cisco Express Forwarding は PE1-CE1 リンクでリンク障害を検出すると、ただちにバックアップ パス PE2 に切り替えます。トラフィックは、バックアップ パスを使用して、Cisco Express Forwarding でのローカル高速コンバージェンスによって迅速に再ルーティングされます。これにより、トラフィックの損失は最小限に抑えられ、迅速なコンバージェンスが行われます。

BGP 最良外部を有効にするためのコンフィギュレーション モード

BGP最良外部機能はさまざまなモードで有効にすることができ、各モードはそれぞれ独自の方法で Virtual Routing and Forwarding（VRF）を保護します。

- VPNv4 アドレス ファミリ コンフィギュレーション モードで **bgp advertise-best-external** コマンドを発行すると、すべての IPv4 VRF に適用されます。このモードでコマンドを発行する場合は、特定の VRF に対して発行する必要はありません。
- IPv4 アドレス ファミリ コンフィギュレーション モードで **bgp advertise-best-external** コマンドを発行すると、その VRF にのみ適用されます。

クラスタ間の RR での BGP ベスト外部パス

BGP ベスト外部機能は、クラスタ間の RR での BGP ベスト外部に拡張されました。この機能は、非クライアント iBGP ピアに対する最良外部機能を提供して、RR クラスタ間におけるパスの多様性を実現します。この機能は、「クラスタ間ベスト外部パス」とも呼ばれます。

RR でのベスト外部パスとは、RR のクラスタ内のベスト パスを意味します。このパスは、ベスト内部パスと呼ばれる場合もあります。

ある RR（RR1）が非クライアント iBGP パス（つまり、別の RR（たとえば、RR2）から学習したパス）を全体でのベスト パスとして選択する場合、クラスタ間の RR での BGP 最良外部機能を使用すると、RR1 はそのベスト内部パスを非クライアント iBGP ピアにアドバタイズできるようになります。これにより、RR2 は追加のパスを学習して、ダイバース パスを提供できます。

RR での最良外部機能は、非クライアント iBGP ピアのみを対象とします。RR は、全体としてのベストパス（クライアントパスである場合も非クライアントeBGPパスである場合もある）をアドバタイズする必要があるため、ベスト外部パスをクライアントにアドバタイズすることはできません。

RR によって計算されるベスト外部パスは、クラスタのベスト内部パスです。このパスは、この RR での全体としてのベスト パスが非クライアント iBGP パスである場合にのみ非クライアント iBGP ピアにアドバタイズされます。

複数の RR が存在し、それぞれ独自のクラスタに含まれている場合、各 RR では、ネイバー RR ごとに **neighbor advertise best-external** コマンドを設定する必要があります。

RR がフォワーディング プレーンにある場合は、**bgp additional paths install** コマンドが必要です。

クラスタ間の RR でのベスト外部パスに関する CLI の違い

PE および RR では BGP ベスト外部機能を使用できます。**bgp advertise-best-external** コマンドの機能は、それぞれベスト外部パスを計算、インストール、およびアドバタイズする次の 3 つのコマンドに分けられています。

- **bgp additional-path select best-external**
- **bgp additional-path install**
- **neighbor advertise diverse-path best-external**

bgp additional-path select best-external コマンドが設定されていない場合は、ベスト外部パスが計算されてインストールされますが、アドバタイズは行われません。

neighbor advertise diverse-path best-external コマンドは、指定したネイバーにベスト外部パスをアドバタイズできるようにします。

クラスタ間の RR での BGP ベスト外部パスの計算に使用されるルール

非クライアント（別のクラスタの RR）に対する RR でのベスト内部パスの実装は、次のルールに基づいて計算されます。

1. 通常のベストパス選択ルールに従って、RR での全体としてのプライマリ ベストパスを計算します。
2. バックアップ パス設定が有効になっている場合は、2 番目のベストパス（ルール 1 で選択されたプライマリ ベストパスとは異なるパスで、このベストパスとは異なるネクストホップを持つパス）を計算し、バックアップ パスとしてマークします。バックアップ パス選択は、**bgp additional-paths install** または **bgp additional-paths select [best-external] [backup]** コマンドを使用して有効にします。
3. RR での全体としてのベスト パスが非クライアント iBGP パスであり、eBGP パスでない場合は、ルール 1 およびルール 2 による結果を除外した後、他のクラスタから得た他のパスをすべて無視して、残りのパスからベスト外部/内部パスを計算し、残りの eBGP パスおよび iBGP パスをすべて含めて通常のベストパス ルールを実行します。新たに得られたベストパスを選択し、ベスト内部パスとしてマークします。
4. このベスト内部パスをアドバタイズします。これは、**neighbor advertise best-external** が非クライアント RR に対して設定されている場合、非クライアント RR に対する eBGP パス（RR/ASBR の CE ピアから受信）または iBGP パス（RR クライアントから受信）になります。

5. 全体としてのベストパスが RR クライアントまたは eBGP ピア（RR/ASBR の場合）から受信されたパスである場合は、iBGP パスまたは eBGP パスが通常のベストパス アルゴリズムに従ってベストパスとして選択されます。全体としてのベストパスは内部クライアントパスであるため、通常のアドバタイズメント ルールによって自動的にこのパスが非クライアント iBGP ピア/RR にアドバタイズされます。この動作は、RR クライアントのパスが全体としてのベストパスとして選択される場合、既存の動作と同じになります（RR で最良外部が有効になっていない場合）。
6. RR クライアントに対する RR でベスト外部パスを設定することはできません。**neighbor advertise best-external** コマンドは、非クライアントに対する、または他のクラスタ内の RR とピアリングする RR/ASBR のみで設定できます。
7. RR でマルチパスが有効になっている場合に、全体としてのベストパスが非クライアントからのパスであり、クラスタ内クライアント パスも一部がマルチパスとしてマークされているときに限り、RR で最良外部を有効化すると（RR 非クライアントに対する **neighbor advertise best-external**）、アルゴリズムでは、クラスタ内クライアントのマルチパス（クラスタ内の RR クライアントおよび eBGP ピアから取得されたパス）のうち、より古いマルチパスを選択し、ベスト内部パスとしてマークし、ベスト外部パスとして非クライアントに通知します。これにより、非クライアントに対してこのクラスタからパスの多様性が提供されます。クラスタ内のマルチパスが見つからない場合は、ルール 3 ～ 5 に従ってベスト外部パスが選択されます。

BGP 最良外部の設定方法

BGP 最良外部機能の設定

BGP 最良外部機能を設定するには、次の作業を実行します。この作業では、IPv4 または VPNv4 アドレス ファミリで BGP 最良外部機能を設定する方法を示します。VPNv4 アドレス ファミリ コンフィギュレーション モードでは、すべての IPv4 Virtural Routing Forwarding（VRF）に BGP 最良外部機能が適用されます。特定の VRF に対して設定する必要はありません。IPv4 VRF アドレス ファミリ コンフィギュレーション モードで **bgp advertise-best-external** コマンドを発行した場合は、その VRF にのみ BGP 最良外部機能が適用されます。

始める前に

- BGP 最良外部機能を設定する前に、MPLS VPN を設定し、正常に動作していることを確認します。詳細については、「Configuring MPLS Layer 3 VPNs」の項を参照してください。
- マルチプロトコル VRF を設定して、ルート ターゲット ポリシー（インポートおよびエクスポート）を IPv4 と IPv6 との間で共有したり、IPv4 VPN と IPv6 VPN に別々のルート ターゲット ポリシーを設定したりすることができるようにします。マルチプロトコル VRF の設定については、「MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs」の項を参照してください。

- カスタマー エッジ（CE）ルータが少なくとも 2 つのパスによってネットワークに接続されていることを確認します。

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	**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	次のいずれかを実行します。
	• **address-family ipv4** [**unicast** | **vrf** *vrf-name*]
	• または
	• **address-family vpnv4** [**unicast**]
	• または
	例：
	`Device(config-router)# address-family ipv4 unicast`
	例：
	`Router(config-router)# address-family vpnv4`
		IPv4 または 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。
		• **unicast** キーワードは、IPv4 または VPNv4 ユニキャスト アドレス ファミリを指定します。
		• **vrf** キーワードおよび *vrf-name* 引数では、後続の IPv4 アドレス ファミリ コンフィギュレーション モード コマンドに関連付ける VRF インスタンスの名前を指定します。

ステップ 5	**bgp advertise-best-external** 例： `Device(config-router-af)# bgp advertise-best-external`	外部バックアップ パスを計算および使用し、RIB および Cisco Express Forwarding にインストールします。
ステップ 6	**neighbor** *ip-address* **remote-as** *autonomous-system-number* 例：	指定された自律システムのネイバーの IP アドレスを、ローカル ルータの IPv4 マルチプロトコル BGP ネイバー テーブルに追加します。

	コマンドまたはアクション	目的
	`Device(config-router-af)# neighbor 192.168.1.1 remote-as 45000`	• デフォルトでは、ルータ コンフィギュレーション モードで **neighbor remote-as** コマンドを使用して定義したネイバーは、IPv4 ユニキャスト アドレス プレフィックスだけを交換します。その他のアドレス プレフィックス タイプを交換するには、そのプレフィックス タイプについて、アドレス ファミリ コンフィギュレーション モードで **neighbor activate** コマンドを使用してネイバーをアクティブ化する必要もあります。
ステップ 7	**neighbor** *ip-address* **activate** 例： `Device(config-router-af)# neighbor 192.168.1.1 activate`	ネイバーが IPv4 ユニキャスト アドレス ファミリのプレフィックスをローカル ルータと交換できるようにします。
ステップ 8	**neighbor** *ip-address* **fall-over** [**bfd** | **route-map** *map-name*] 例： `Device(config-router-af)# neighbor 192.168.1.1 fall-over bfd`	高速セッションの非アクティブ化を使用するように BGP ピアリングを設定し、フェールオーバーでの BFD プロトコル サポートを有効にします。 • BGP は、セッションが無効になると、このピアで学習したすべてのルートを削除します。
ステップ 9	**end** 例： `Device(config-router-af)# end`	（任意）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。

BGP 最良外部機能の確認

BGP 最良外部機能が正しく設定されていることを確認するには、次の作業を実行します。

手順

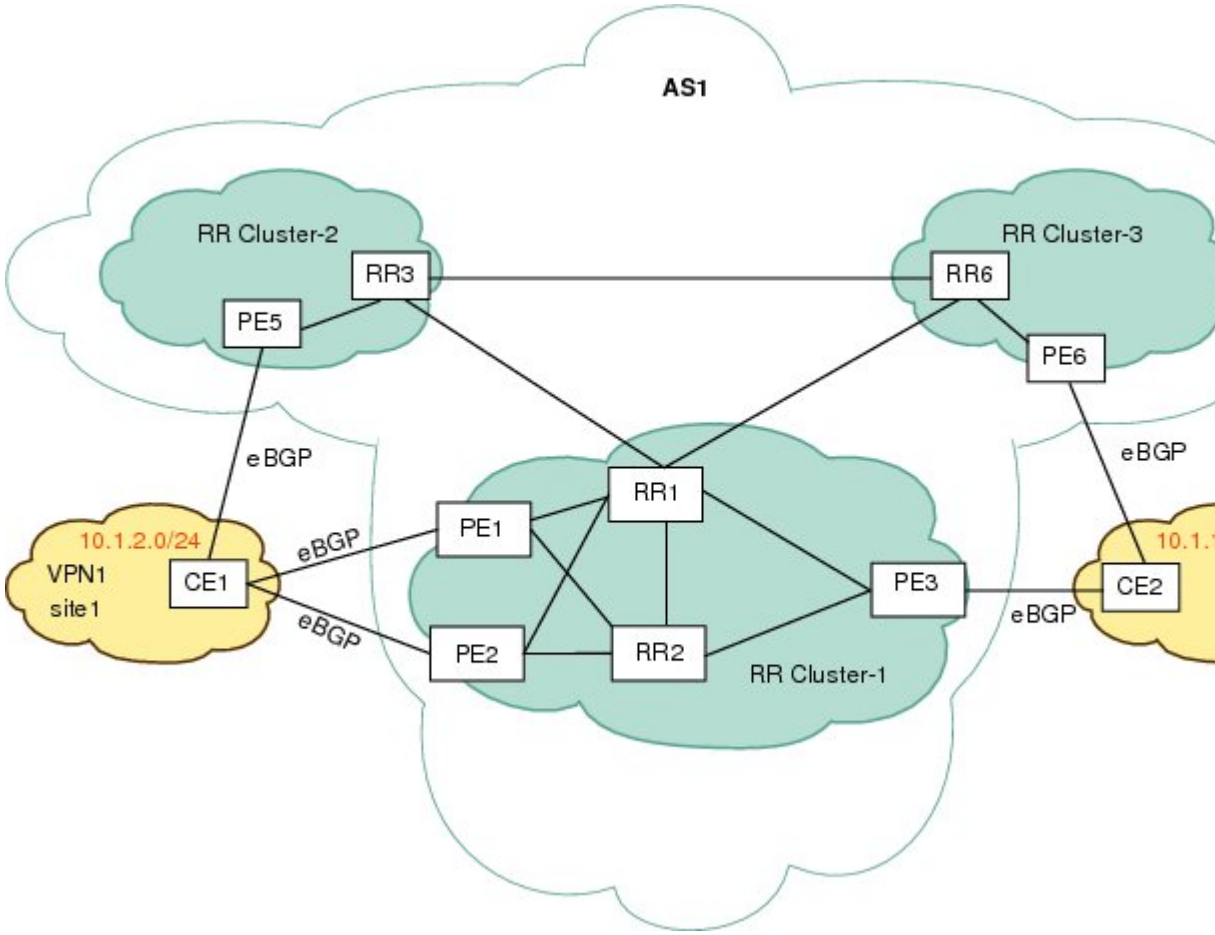
	コマンドまたはアクション	目的
ステップ 1	**enable** 例：	このコマンドを使用して、特権 EXEC モードをイネーブルにします。プロンプ

	コマンドまたはアクション	目的
	`Device> enable`	トが表示されたらパスワードを入力します。次に例を示します。
ステップ 2	**show vrf detail** 例： `Device> show vrf detail`	このコマンドを使用して、BGP 最良外部機能が有効になっていることを確認します。次の **show vrf detail** コマンド出力は、 BGP 最良外部機能が有効になっていることを示しています。
ステップ 3	**show ip bgp ipv4 mdt all** | **rd vrf**} | **multicast** | **tunnel unicast** or **show ip bgp vpn4 all rd** *route-distinguisher* | **vrf** *vrf-name* **rib-failure** *ip-prefix/length* **longer-prefixes**]] *network-address mask* **longer-prefixes**]] **cidr-only community** *community-list* **dampened-paths filter-list**] [**flap-statistics inconsistent-as neighbors paths line**]] **peer-group quote-regexp regexp**] [**summary labels** 例： `Device# show ip bgp vpnv4 all`	このコマンドを使用して、ベスト外部ルートがアドバタイズされていることを確認します。コマンド出力で、コードbはバックアップ パスを示し、コード x はベスト外部パスを示します。
ステップ 4	**show bgp vpnv4 unicast vrf** *vrf-name ip-address* 例： `Device# show bgp vpnv4 unicast vrf vpn1 10.10.10.10`	このコマンドを使用して、ベスト外部ルートがアドバタイズされていることを確認します。
ステップ 5	**show ip route vrf** *vrf-name* **repair-paths** *ip-address* 例： `Device# show ip route vrf vpn1 repair-paths`	このコマンドを使用して、修復ルートを表示します。
ステップ 6	**show ip cef vrf** *vrf-name ip-address* **detail** 例： `Device# show ip cef vrf test 10.71.8.164 detail`	このコマンドを使用して、ベスト外部ルートを表示します。

クラスタ間の RR でのベスト外部パスの設定

クラスタ間の RR でのベスト外部パスを設定するには、次の作業を実行します。この特定作業の手順では、IPv4 アドレス ファミリで、下の図の RR1 を設定します。アドレス ファミリを設定する手順では、サポートされているその他のアドレス ファミリを一覧表示します。

図 *9:* クラスタ間の *RR* での *BGP* ベスト外部パスを設定するシナリオ



手順

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

	コマンドまたはアクション	目的
	`Device# configure terminal`	
ステップ 3	**router bgp** *autonomous-system-number* 例： `Device(config)# router bgp 1`	指定したルーティングプロセスのルータコンフィギュレーションモードを開始します。
ステップ 4	**neighbor** *ip-address* **remote-as** *autonomous-system-number* 例： `Device(config-router)# neighbor 10.5.1.1 remote-as 1`	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。 • この手順は RR3 用です。
ステップ 5	**neighbor** *ip-address* **remote-as** *autonomous-system-number* 例： `Device(config-router)# neighbor 10.5.1.2 remote-as 1`	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。 • この手順は RR6 用です。
ステップ 6	**address-family ipv4 unicast** 例： `Device(config-router)# address-family ipv4 unicast`	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。 • サポートされているアドレスファミリは、IPv4 ユニキャスト、VPNv4 ユニキャスト、IPv6 ユニキャスト、VPNv6ユニキャスト、IPv4+ラベル、IPv6+ラベルです。
ステップ 7	**neighbor***ip-address* **activate** 例： `Device(config-router-af)# neighbor 10.5.1.1 activate`	BGP ネイバーとの情報交換を有効にします。 • この手順は RR3 用です。
ステップ 8	**neighbor***ip-address* **activate** 例： `Device(config-router-af)# neighbor 10.5.1.2 activate`	BGP ネイバーとの情報交換を有効にします。 • この手順は RR6 用です。
ステップ 9	**bgp additional-paths select best-external** 例： `Device(config-router-af)# bgp additional-paths select best-external`	ベスト外部パス（RR クラスタ外）を計算するようにシステムを設定します。
ステップ 10	**bgp additional-paths install** 例：	BGP で特定のアドレス ファミリのバックアップ パスを計算し、RIB および

	コマンドまたはアクション	目的
	`Device(config-router-af)# bgp additional-paths install`	CEF にインストールできるようにします。 • この手順は、RRが転送に対して有効になっている場合（RR がフォワーディング プレーンにある場合）に必要です。それ以外の場合、この手順は不要です。
ステップ 11	**neighbor** *ip-address* **advertise best-external** 例： `Device(config-router-af)# neighbor 10.5.1.1 advertise best-external`	（任意）アドバタイズでベスト外部パスを受信するようにネイバーを設定します。 • この手順は RR3 用です。
ステップ 12	**neighbor** *ip-address***advertise best-external** 例： `Device(config-router-af)# neighbor 10.5.1.2 advertise best-external`	（任意）アドバタイズでベスト外部パスを受信するようにネイバーを設定します。 • この手順は RR6 用です。
ステップ 13	**end** 例： Device(config-router-af)# end	（任意）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。

上記のシナリオでは、次のパスが、3 つの異なるクラスタ内にある 3 つの RR でベスト パス、バックアップ パス、ベスト内部パスとして選択されます。

RR1：

RR3：

RR6：

プレフィックス 10/8 に到達するため	ネクスト ホップ：
	PE5（ベストパス、ローカル プリファレンス = 200）
	PE3（バックアップ パス、ローカル プリファレンス = 150）
	PE3（ベスト内部パス、ローカル プリファレンス = 150）

プレフィックス 10/8 に到達するため	ネクスト ホップ：
	PE5（ベストパス、ローカル プリファレンス = 200）
	PE6（バックアップ パス、ローカル プリファレンス = 50）
	PE3（RR1 からベスト外部パスとして受信、ローカル プリファレンス = 150）

プレフィックス 10/8 に到達するため	ネクスト ホップ：
	PE5（ベストパス、ローカル プリファレンス = 200）
	PE6（バックアップ パス、ローカル プリファレンス = 50）
	PE3（RR1 からベスト外部パスとして受信、ローカル プリファレンス = 150）

BGP 最良外部の設定例

例：BGP 最良外部機能の設定

次の例は、VPNv4 モードで BGP 最良外部機能を設定する方法を示しています。

```
vrf definition test1
 rd 400:1
 route-target export 100:1
 route-target export 200:1
 route-target export 300:1
 route-target export 400:1
 route-target import 100:1
 route-target import 200:1
 route-target import 300:1
 route-target import 400:1
 address-family ipv4
 exit-address-family
 exit
!
interface Ethernet1/0
 vrf forwarding test1
 ip address 10.0.0.1 255.0.0.0
 exit
!
router bgp 64500
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.5.5.5 remote-as 64500
 neighbor 10.5.5.5 update-source Loopback0
 neighbor 10.6.6.6 remote-as 64500
```

```
 neighbor 10.6.6.6 update-source Loopback0
 no auto-summary
 !
 address-family vpnv4

bgp advertise-best-external
  neighbor 10.5.5.5 activate
  neighbor 10.5.5.5 send-community extended
  neighbor 10.6.6.6 activate
  neighbor 10.6.6.6 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf test1
  no synchronization
bgp recursion host
  neighbor 192.168.13.2 remote-as 64511
  neighbor 192.168.13.2 fall-over bfd
  neighbor 192.168.13.2 activate
  neighbor 192.168.13.2 as-override
 exit-address-family
```

例：クラスタ間の RR でのベスト外部パスの設定

次の例では、「クラスタ間の RR でのベスト外部パスの設定」の項に示されている図の RR1 を設定しています。RR1 は、クラスタ間の RR ネイバーへのベスト外部パスを計算、インストール、およびアドバタイズするように設定されています。

RR1

```
router bgp 1
 neighbor 10.5.1.1 remote-as 1
 neighbor 10.5.1.2 remote-as 1
 address-family ipv4 unicast
 neighbor 10.5.1.1 activate
 neighbor 10.5.1.2 activate
 bgp additional-paths select best-external
 bgp additional-paths install
 neighbor 10.5.1.1 advertise best-external
 neighbor 10.5.1.2 advertise best-external
 end
```

その他の参考資料

関連資料

関連項目	マニュアル タイトル
基本的な MPLS VPN	『*MPLS: Layer 3 VPNs Configuration Guide*』の「Configuring MPLS Layer 3 VPNs」モジュール
マルチプロトコル VRF	『*MPLS: Layer 3 VPNs Configuration Guide*』の「MPLS VPN VRF CLI for IPv4 and IPv6 VPNs」モジュール

関連項目	マニュアル タイトル
リンクまたはノード障害の後に新しいパスを作成するフェールオーバー機能	『*MPLS VPN--BGP Local Convergence*』

標準

標準	タイトル
draft-marques-idr-best-external	『*BGP Best External, Advertisement of the best external route to iBGP*』

RFC

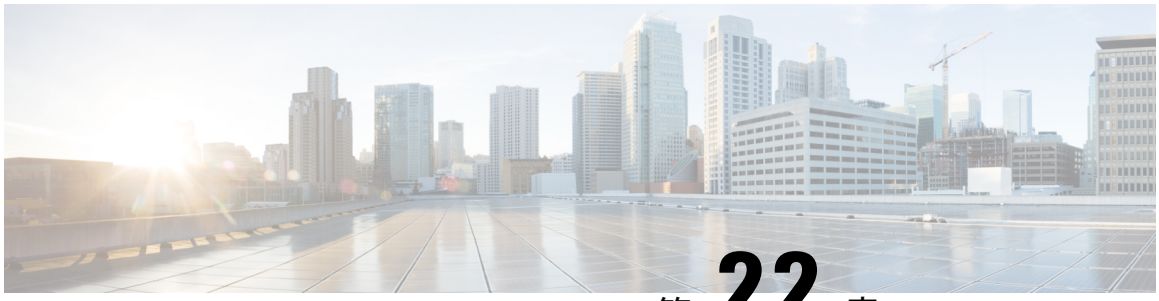
RFC	タイトル
RFC 1771	『*A Border Gateway Protocol 4 (BGP-4)*』
RFC 2547	『*BGP/MPLS VPNs*』

BGP 最良外部の機能情報

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

表 *34 : BGP* 最良外部の機能情報

リリース	機能情報
Cisco IOS XE Gibraltar 16.10.x	この機能が導入されました。



第 22 章

BGP-VPN 識別子属性の設定

- BGP-VPN 識別子属性に関する情報 （261 ページ）
- BGP-VPN 識別子属性 （263 ページ）
- BGP-VPN 識別子属性の設定方法 （263 ページ）
- BGP-VPN 識別子属性の設定例 （269 ページ）
- BGP-VPN 識別子属性の機能情報 （270 ページ）

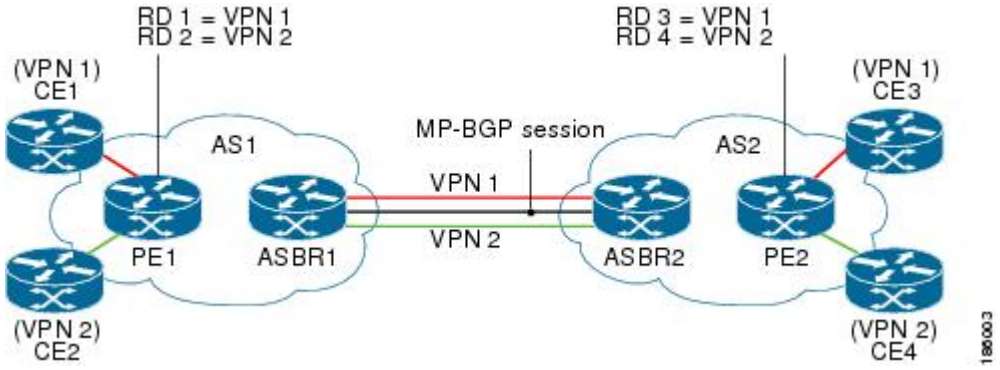
BGP-VPN 識別子属性に関する情報

VPN 識別子属性の役割と利点

route-target（RT）拡張コミュニティ属性は、ルートの VPN メンバーシップを識別します。RT 属性は、エクスポート側（出力）プロバイダー エッジ ルータ（PE）でルートに配置され、iBGP クラウド全体およびすべての自律システムに転送されます。このようなルートをインポートする必要があるリモート PE の Virtual Routing and Forwarding（VRF）インスタンスでは、対応する RT がその VRF のインポート RT として設定されている必要があります。

下の図には、異なる VPN に属するカスタマー エッジ（CE）ルータを含む 2 つの自律システムが示されています。各 PE は、どのルート識別子（RD）がどの VPN に対応するかを追跡して、各 VPN に属するトラフィックを制御します。

図 *10* : 自律システム間で *ASBR* が *RT* を変換するシナリオ



上の図に示されているような Inter-AS オプション B のシナリオでは、これらのルートは、MP-eBGP セッションを介して自律システム境界ルータ 1（ASBR1）から AS 境界を越えて ASBR2 に伝送され、ルートの各 RT は拡張コミュニティ属性として ASBR2 によって受信されます。

ASBR2 では、CE3 および CE4 に対する PE2 上の各 VPN メンバーシップの CE 接続で RT をインポートできるように、AS1 によって生成された RT を AS2 で認識できる RT に変換するための複雑な RT マッピング スキームを維持する必要があります。

ネットワーク管理者によっては、AS1 の送信元 RT を AS2 内のデバイスからは認識できないようにすることを必要とする場合があります。それには、各 VPN に属するルートを特定の属性によって区別する必要があります。これにより、ASBR2 にルートを送信する前に ASBR1 の発信側で RT を削除できるようになり、ASBR2 でその属性を AS2 の認識可能な RT にマッピングできるようになります。VPN 識別子（VD）拡張コミュニティ属性はこの目的に役立ちます。

BGP—VPN 識別子属性機能の利点は、送信元 RT を宛先自律システムのデバイスからプライベートに保てることです。

VPN 識別子属性の仕組み

ネットワーク管理者は、VPN 識別子拡張コミュニティ属性への RT の変換を実行するように出力 ASBR を設定し、RT への VPN 識別子の変換を実行するように入力 ASBR を設定します。より具体的には、この変換は次のように実現されます。

出力 ASBR 側

- 発信ルート マップで、ルートの RT 値に基づいてどの VPN ルートがマッピング対象となるかを判別する **match excommunity** 句を指定します。
- **set extcommunity vpn-distinguisher** コマンドで、RT を置き換える VPN 識別子を設定します。
- RT を削除するように、同じ RT セットを参照する **set extcomm-list delete** コマンドを設定します。その後、隣接する入力 ASBR にルートが送信されます。

入力 ARBR 側

- 着信ルート マップで、ルートの VPN 識別子に基づいてどの VPN ルートがマッピング対象となるかを判別する **match excommunity vpn-distinguisher** コマンドを指定します。
- **set extcommunity rt** コマンドで、VPN 識別子を置き換える RT を指定します。
- この句に一致するルートでは、VPN 識別子は設定した RT に置き換えられます。

VPN 識別子に関連するその他の動作

出力 ASBR で、**set extcommunity vpn-distinguisher** コマンドが設定されていないルート マップ句に VPN ルートが一致した場合、VPN ルートにタグ付けされている RT は保持されます。

VPN識別子はAS境界を越えて移動しますが、iBGPクラウド内では伝送されません。つまり、入力ASBRはeBGPピアからVPN識別子を受信できますが、VPN識別子は対応するRTにマッピングされた後に着信側で破棄されます。

入力ASBRで、VPN識別子を伝送するVPNルートが、着信ルートマップの**set extcommunity rt**コマンドが設定されていないルートマップ句と一致した場合、その属性は、破棄されることも、iBGPクラウド内で伝播されることもありません。ルートのVPN識別子は保持されるため、ネットワーク管理者は、VPNルートで伝送する必要があるRTにVPN識別子を変換するための適切な着信ポリシーを設定できます。ルートがeBGPピアに送信される場合、VPN識別子はそのまま伝送されます。ネットワーク管理者は、eBGPピアに送信されるルートからVPN識別子を削除するようにルートマップ エントリを設定できます。

発信ルートマップで**set extcommunity vpn-distinguisher**コマンドを設定すると、または着信ルートマップで**match excommunity**コマンドを設定すると、送受信されるルートを更新するために、それぞれ発信または着信ルートリフレッシュ リクエストが生成されます。

BGP-VPN 識別子属性

BGP—VPN識別子属性機能により、ネットワーク管理者は、宛先自律システム内の自律システム境界ルータ（ASBR）から送信元ルートターゲット（RT）をプライベートに保つことができます。出力ASBRのRTがVPN識別子にマッピングされ、VPN識別子がeBGPを介して伝送されて、入力ASBRのRTにマッピングされます。

BGP-VPN 識別子属性の設定方法

RT を VPN 識別子属性に置き換える

ルートターゲット（RT）をVPN識別子拡張コミュニティ属性に置き換えるには、出力ASBRでこの作業を実行します。必ず、入力ASBRでVPN識別子をルートターゲットに置き換えてください。この作業については、「VPN識別子属性をRTに置き換える」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：	グローバル 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 extcommunity-list** *expanded-list* {**permit** | **deny**} **rt** *value* 例： `Device(config)# ip extcommunity-list 4 permit rt 101:100`	IP 拡張コミュニティ リストを設定して、指定した RT を持つルートが拡張コミュニティ リストに含まれるように、バーチャル プライベート ネットワーク（VPN）ルート フィルタリングを設定します。 • この例では、RT 101:100 を持つルートを拡張コミュニティ リスト 4 に対して許可しています。
ステップ 4	**exit** 例： `Device(config-extcomm-list)# exit`	コンフィギュレーション モードを終了し、次に高い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	**route-map** *map-tag* {**permit** | **deny**} [*sequence-number*] 例： `Device(config)# route-map vpn-id-map1 permit 10`	後続の **match** コマンドで一致と認められたルートを許可または拒否するルート マップを設定します。 • この例では、後続の **match** コマンドで一致と認められたルートを許可します。
ステップ 6	**match extcommunity** *extended-community-list-name* 例： `Device(config-route-map)# match extcommunity 4`	指定したコミュニティ リストを照合します。 • この例では、拡張コミュニティ リスト 4（手順 3 で設定）に一致するルートが後続の **set** コマンドの対象となります。
ステップ 7	**set extcomm-list** *extcommunity-name* **delete** 例： `Device(config-route-map)# set extcomm-list 4 delete`	指定した拡張コミュニティ リスト内のルートから RT を削除します。 • この例では、拡張コミュニティ リスト 4 内のルートから RT が削除されます。
ステップ 8	**set extcommunity vpn-distinguisher** *id* 例： `Device(config-route-map)# set`	ルート マップで許可されているルートに対して、指定した VPN 識別子を設定します。

	コマンドまたはアクション	目的
	`extcommunity vpn-distinguisher 111:100`	• この例では、拡張コミュニティ 4 に一致するルートに VPN 識別子 111:100 を設定します。
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、グローバル コンフィギュレーション モードを開始します。
ステップ 10	**route-map** *map-name* {**permit** | **deny**} [*sequence-number*] 例： `Device(config)# route-map vpn-id-map1 permit 20`	（任意）ルートを許可するルートマップ エントリを設定します。 • この例では、RT から VPN 識別子へのマッピングの対象とならない他のルートを許可するルートマップエントリを設定します。この手順を実行しない場合、他のすべてのルートは暗黙の deny の対象となります。
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** *as-number* 例： `Device(config)# router bgp 2000`	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* **remote-as** *autonomous-system-number* 例： `Device(config-router)# neighbor 192.168.101.1 remote-as 2000`	自律システムに属するネイバーを指定します。
ステップ 14	**address-family vpnv4** 例： `Device(config-router)# address-family vpnv4`	アドレス ファミリ コンフィギュレーション モードを開始して、アドレス ファミリ固有の設定を受け入れるように BGP ピアを設定します。
ステップ 15	**neighbor** *ip-address* **activate** 例：	指定したネイバーをアクティブにします。

	コマンドまたはアクション	目的
	`Device(config-router-af)# neighbor 192.168.101.1 activate`	
ステップ 16	**neighbor** *ip-address* **route-map** *map-name* **out** 例： `Device(config-router-af)# neighbor 192.168.101.1 route-map vpn-id-map1 out`	指定した発信ルートマップを、指定したネイバーに適用します。
ステップ 17	**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 モードを終了して、特権 EXEC モードを開始します。

VPN 識別子属性を RT に置き換える

VPN 識別子拡張コミュニティ属性をルート ターゲット（RT）属性に置き換えるには、入力 ASBR でこの作業を実行します。この作業では、RT を VPN 識別子に置き換えるように出力 ASBR を設定済みであることを前提としています。この作業については、「RT を VPN 識別子属性に置き換える」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	**ip extcommunity-list** *expanded-list* {**permit** | **deny**} **vpn-distinguisher** *id* 例： `Device(config)# ip extcommunity-list 51 permit vpn-distinguisher 111:100`	IP 拡張コミュニティ リストを設定して、指定した VPN 識別子を持つルートが拡張コミュニティ リストに含まれるように、バーチャル プライベート ネットワーク（VPN）ルート フィルタリングを設定します。

	コマンドまたはアクション	目的
		• この例では、VPN 識別子 111:110 を持つルートを拡張コミュニティリスト 51 に対して許可しています。
ステップ 4	**exit** 例： `Device(config-extcomm-list)# exit`	コンフィギュレーションモードを終了し、次に高い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	**route-map** *map-tag* {**permit** | **deny**} [*sequence-number*] 例： `Device(config)# route-map vpn-id-rewrite-map1 permit 10`	後続の **match** コマンドで一致と認められたルートを許可または拒否するルート マップを設定します。 • この例では、後続の **match** コマンドで一致と認められたルートを許可します。
ステップ 6	**match extcommunity** *extended-community-list-name* 例： `Device(config-route-map)# match extcommunity 51`	指定したコミュニティリストを照合します。 • この例では、拡張コミュニティリスト 51（手順 3 で設定）に一致するルートが後続の **set** コマンドの対象となります。
ステップ 7	**set extcomm-list** *extcommunity-name* **delete** 例： `Device(config-route-map)# set extcomm-list 51 delete`	指定した拡張コミュニティリスト内のルートから VPN 識別子を削除します。 • この例では、拡張コミュニティリスト 51 内のルートから VPN 識別子が削除されます。
ステップ 8	**set extcommunity rt** *value* **additive** 例： `Device(config-route-map)# set extcommunity rt 101:1 additive`	ルートマップで許可されているルートに、指定した RT を設定します。 • この例では、拡張コミュニティ 51 に一致するルートに RT 101:1 を設定します。**additive** キーワードを指定すると、RT を置き換えずに RT が RT リストに追加されます。

	コマンドまたはアクション	目的
ステップ 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、グローバル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 vpn-id-rewrite-map1 permit 20`	（任意）ルートを許可するルートマップ エントリを設定します。 • この例では、VPN 識別子から RT へのマッピングの対象とならない他のルートを許可するルートマップエントリを設定します。この手順を実行しない場合、他のすべてのルートは暗黙の deny の対象となります。
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** *as-number* 例： `Device(config)# router bgp 3000`	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* **remote-as** *autonomous-system-number* 例： `Device(config-router)# neighbor 192.168.0.81 remote-as 3000`	自律システムに属するネイバーを指定します。
ステップ 14	**address-family vpnv4** 例： `Device(config-router-af)# address-family vpnv4`	アドレス ファミリ コンフィギュレーション モードを開始して、アドレス ファミリ固有の設定を受け入れるように BGP ピアを設定します。
ステップ 15	**neighbor** *ip-address* **activate** 例： `Device(config-router-af)# neighbor 192.168.0.81 activate`	指定したネイバーをアクティブにします。

	コマンドまたはアクション	目的
ステップ 16	**neighbor** *ip-address* **route-map** *map-name* **in** 例： `Device(config-router-af)# neighbor 192.168.0.81 route-map vpn-id-rewrite-map1 in`	指定した発信ルートマップを、指定したネイバーに適用します。
ステップ 17	**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 モードを終了して、特権 EXEC モードを開始します。

例

BGP-VPN 識別子属性の設定例

例：RT から VPN 識別子への変換と VPN 識別子 から RT への変換

次の例は、ルート ターゲット（RT）を VPN 識別子に置き換えるための出力 ASBR の設定、および VPN 識別子をルート ターゲットに置き換えるための入力 ASBR の設定を示しています。

出力 ASBR では、VPN ルートをフィルタ処理して RT 101:100 のルートのみを許可するように、IP 拡張コミュニティ リスト 1 を設定します。vpn-id-map1 という名前のルートマップで、IP 拡張コミュニティ リスト 1 によって許可されているルートに一致するすべてのルートが 2 つの **set** コマンドの対象となるように指定します。1 つ目の **set** コマンドは、ルートから RT を削除します。2 つ目の **set** コマンドは、VPN 識別子属性を 111:100 に設定します。

route-map vpn-id-map1 permit 20 コマンドは、RT から VPN 識別子へのマッピングに含まれない他のルートが、破棄されないようルート マップを通過できるようにします。このコマンドを使用しないと、暗黙の deny によってこれらのルートは破棄されます。

最後に、自律システム 2000 で、VPNv4 アドレス ファミリについて、ルート マップ vpn-id-map1 を 192.168.101.1 のネイバーに送出されるルートに適用します。

出力 ASBR

```
ip extcommunity-list 1 permit rt 101:100
!
```

```
route-map vpn-id-map1 permit 10
 match extcommunity 1
 set extcomm-list 1 delete
 set extcommunity vpn-distinguisher 111:100
!
route-map vpn-id-map1 permit 20
!
router bgp 2000
 neighbor 192.168.101.1 remote-as 2000
 address-family vpnv4
  neighbor 192.168.101.1 activate
  neighbor 192.168.101.1 route-map vpn-id-map1 out
  exit-address-family
!
```

入力 ASBR では、IP 拡張コミュニティ リスト 51 で、VPN 識別子が 111:100 であるルートを許可します。vpn-id-rewrite-map1 という名前のルート マップで、IP 拡張コミュニティ リスト 51 によって許可されているルートに一致するすべてのルートが 2 つの **set** コマンドの対象となるように指定します。1 つ目の **set** コマンドは、ルートから VPN 識別子を削除します。2 つ目の **set** コマンドは RT を 101:1 に設定し、RT を置き換えずにその RT を RT リストに追加します。

route-map vpn-id-rewrite-map1 permit 20 コマンドは、VPN 識別子から RT へのマッピングに含まれない他のルートが、破棄されないようルート マップを通過できるようにします。このコマンドを使用しないと、暗黙の deny によってこれらのルートは破棄されます。

最後に、自律システム 3000 で、VPNv4 アドレス ファミリについて、vpn-id-rewrite-map1 という名前のルート マップを 192.168.0.81 のネイバーを宛先とする着信ルートに適用します。

入力 ASBR

```
ip extcommunity-list 51 permit vpn-distinguisher 111:100
!
route-map vpn-id-rewrite-map1 permit 10
 match extcommunity 51
 set extcomm-list 51 delete
 set extcommunity rt 101:1 additive
!
route-map vpn-id-rewrite-map1 permit 20
!
router bgp 3000
 neighbor 192.168.0.81 remote-as 3000
 address-family vpnv4
  neighbor 192.168.0.81 activate
  neighbor 192.168.0.81 route-map vpn-id-rewrite-map1 in
  exit-address-family
!
```

BGP-VPN 識別子属性の機能情報

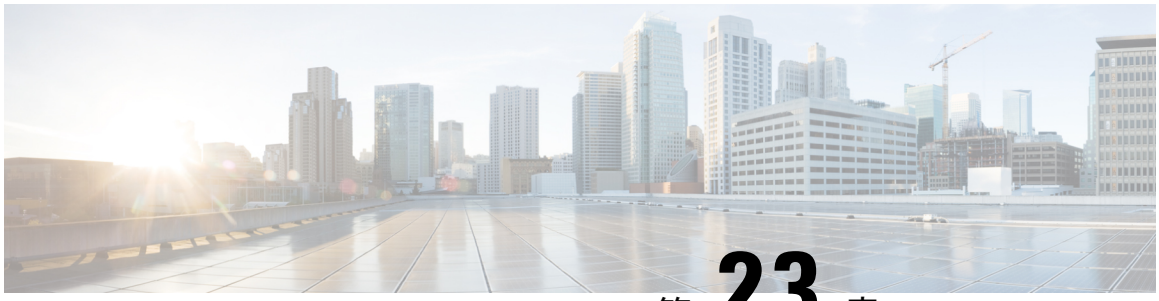
次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだ

けを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

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

表 *35 : BGP-VPN* 識別子属性の機能情報

リリース	機能情報
Cisco IOS XE Gibraltar 16.10.1	この機能が導入されました。



第 23 章

BGP-RT および VPN 識別子属性の書き換えワイルドカードの設定

- BGP-RT および VPN 識別子属性の書き換えワイルドカード （273 ページ）
- BGP-RT および VPN 識別子属性の書き換えワイルドカードに関する制約事項 （274 ページ）
- BGP-RT および VPN 識別子属性の書き換えワイルドカードに関する情報 （274 ページ）
- 範囲を使用してRT を RT にマッピングする方法 （275 ページ）
- BGP-RT および VPN 識別子属性の書き換えワイルドカードの設定例 （280 ページ）
- BGP-RT および VPN 識別子属性の書き換えワイルドカードの設定例 （280 ページ）
- BGP-RT および VPN 識別子属性の書き換えワイルドカードに関する追加情報 （282 ページ）
- BGP—RT および VPN 識別子属性の書き換えワイルドカードに関する機能情報 （282 ページ）

BGP-RT および VPN 識別子属性の書き換えワイルドカード

BGP—RT および VPN 識別子属性の書き換えワイルドカード機能は、マッピングの際にルートターゲット（RT）コミュニティ属性または VPN 識別子コミュニティ属性の範囲を設定できるようにします。出力 ASBR における 1 つ以上の RT を入力 ASBR における別の RT にマッピングすることが必要となる場合があります。VPN 識別子属性機能により、管理者は、eBGP を介して伝送される VPN 識別子に RT をマッピングし、次に入力 ASBR で RT にマッピングすることができます。このマッピングは、拡張コミュニティ属性の RT 範囲または VPN 識別子範囲を指定するルートマップを設定することによって実現されます。個々の RT ではなく範囲を指定することにより、時間が節約され、設定が簡素化されます。また、VPN 識別子範囲では、route-map 句ごとに複数の VPN 識別子属性を使用できるため、この機能が導入される前に適用されていた制約がなくなります。

BGP-RT および VPN 識別子属性の書き換えワイルドカードに関する制約事項

- 範囲（**set extcommunity rt** コマンドまたは **set extcommunity vpn-distinguisher** コマンドで指定）には、最大 450 個の拡張コミュニティを含めることができます。
- VPN 識別子範囲は、iBGP ピアにはリレーされません。

BGP-RT および VPN 識別子属性の書き換えワイルドカードに関する情報

RT および VPN 識別子属性のマッピング範囲の利点

出力 ASBR における 1 つ以上のルート ターゲット（RT）を入力 ASBR における別の RT に書き換える（マッピングする）ことが必要となる場合があります。1 つの使用例は、出力 ASBR の RT を入力 ASBR からプライベートに保つことです。

この書き換えは、着信ルート マップを使用し、route-map 句で着信 RT とプレフィックスを照合して、一致する RT をネイバー AS で認識できる別の RT にマッピングすることによって実現されます。このような書き換えの設定は、着信ルート マップで、数百もの RT を個別に指定（**set extcommunity rt** *value1 value2 value3 ...* のように設定）しなければならない場合もあるため、複雑になることがあります。プレフィックスに対応する RT が連続している場合は、RT の範囲を指定することで設定を簡素化できます。つまり、RT マッピング範囲の利点は、時間の節約と設定の簡素化です。

同様に、VPN 識別子属性への RT のマッピング（およびその逆）も、RT または VPN 識別子の範囲を指定することで簡素化できます。BGP—VPN 識別子属性機能により、ネットワーク管理者は、宛先 AS 内の ASBR から送信元 RT をプライベートに保つことができます。出力 ASBR の RT が VPN 識別子にマッピングされ、VPN 識別子が eBGP を介して伝送されて、入力 ASBR の RT にマッピングされます。

RT および VPN 識別子属性のマッピング範囲機能は、マッピングの際にルート ターゲット（RT）または VPN 識別子の範囲を指定できるようにします。

もう 1 つの利点は、VPN 識別子の設定で得られます。この機能が導入される前は、route-map 句ごとに使用できる **set extcommunity vpn-distinguisher** 値は 1 つだけでした。マッピング範囲の導入により、VPN 識別子の範囲をルートに設定できます。

範囲を使用してRT を RT にマッピングする方法

RT を RT 範囲に置き換える

ルート ターゲット（RT）を RT 範囲に置き換えるには、出力 ASBR でこの作業を実行します。必ず、入力 ASBR で RT の範囲を RT に置き換えてください。この作業については、「RT 範囲を RT に置き換える」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	**ip extcommunity-list** *expanded-list* {**permit** | **deny**} 例： `Device(config)# ip extcommunity-list 22 permit`	IP 拡張コミュニティリストを設定して、バーチャル プライベート ネットワーク（VPN）ルートフィルタリングを設定します。
ステップ 4	**exit** 例： `Device(config-extcomm-list)# exit`	コンフィギュレーション モードを終了し、次に高い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	**route-map** *map-tag* {**permit** | **deny**} [*sequence-number*] 例： `Device(config)# route-map rt-mapping permit 10`	後続の **match** コマンドで一致と認められたルートを許可または拒否するルート マップを設定します。 • この例では、後続の **match** コマンドで一致と認められたルートを許可します。
ステップ 6	**match extcommunity** *extended-community-list-name* 例：	指定したコミュニティ リストを照合します。

	コマンドまたはアクション	目的
	`Device(config-route-map)# match extcommunity 22`	• この例では、拡張コミュニティリスト 22（手順 3 で設定）に一致するルートが後続の **set** コマンドの対象となります。
ステップ 7	**set extcomm-list** *extcommunity-name* **delete** 例： `Device(config-route-map)# set extcomm-list 22 delete`	指定した拡張コミュニティリスト内のルートから RT を削除します。 • この例では、拡張コミュニティリスト 22 内のルートから RT が削除されます。
ステップ 8	**set extcommunity rt range** *start-value end-value* 例： `Device(config-route-map)# set extcommunity rt range 500:1 500:9`	ルートマップで許可されているルートに対して、拡張コミュニティ属性の指定した RT 範囲（境界値を含む）を設定します。 • この例では、拡張コミュニティ 22 に一致するルートに対して、500:1、500:2、500:3、500:4、500:5、500:6、500:7、500:8、500:9 の RT 拡張コミュニティ属性値を設定しています。
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、グローバル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 rt-mapping permit 20`	（任意）ルートを許可するルートマップ エントリを設定します。 • この例では、RT から RT 範囲へのマッピングの対象とならない他のルートを許可するルートマップエントリを設定します。この手順を実行しない場合、他のすべてのルートは暗黙の deny の対象となります。
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。

	コマンドまたはアクション	目的
ステップ 12	**router bgp** *as-number* 例： `Device(config)# router bgp 3000`	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* **remote-as** *autonomous-system-number* 例： `Device(config-router)# neighbor 192.168.103.1 remote-as 3000`	自律システムに属するネイバーを指定します。
ステップ 14	**address-family vpnv4** 例： `Device(config-router)# address-family vpnv4`	アドレス ファミリ コンフィギュレーション モードを開始して、アドレス ファミリ固有の設定を受け入れるように BGP ピアを設定します。
ステップ 15	**neighbor** *ip-address* **activate** 例： `Device(config-router-af)# neighbor 192.168.103.1 activate`	指定したネイバーをアクティブにします。
ステップ 16	**neighbor** *ip-address* **route-map** *map-tag* **out** 例： `Device(config-router-af)# neighbor 192.168.103.1 route-map rt-mapping out`	指定した発信ルートマップを、指定したネイバーに適用します。
ステップ 17	**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 モードを終了して、特権 EXEC モードを開始します。

RT 範囲を RT に置き換える

属性の RT 範囲を RT 属性に置き換えるには、入力 ASBR でこの作業を実行します。この作業では、RT を RT 範囲に置き換えるように出力 ASBR を設定済みであることを前提としています。この作業については、「RT を RT 範囲に置き換える」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	**ip extcommunity-list** *expanded-list* {**permit** | **deny**} **rt** *reg-exp* 例： `Device(config)# ip extcommunity-list 128 permit rt 500:[1-9]`	IP 拡張コミュニティ リストを設定して、指定した RT 範囲の RT を持つルートが拡張コミュニティリストに含まれるように、バーチャル プライベート ネットワーク（VPN）ルートフィルタリングを設定します。 • この例では、500:1 ～ 500:9 の範囲内の RT を持つルートを拡張コミュニティ リスト 128 に対して許可しています。
ステップ 4	**exit** 例： `Device(config-extcomm-list)# exit`	コンフィギュレーションモードを終了し、次に高い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	**route-map** *map-tag* {**permit** | **deny**} [*sequence-number*] 例： `Device(config)# route-map rtmap2 permit 10`	後続の **match** コマンドで一致と認められたルートを許可または拒否するルート マップを設定します。 • この例では、後続の **match** コマンドで一致と認められたルートを許可します。
ステップ 6	**match extcommunity** *extended-community-list-name* 例： `Device(config-route-map)# match extcommunity 128`	指定したコミュニティリストを照合します。 • この例では、拡張コミュニティ リスト 128（手順 3 で設定）に一致するルートが後続の **set** コマンドの対象となります。

	コマンドまたはアクション	目的
ステップ 7	**set extcomm-list** *extcommunity-name* **delete** 例： `Device(config-route-map)# set extcomm-list 128 delete`	指定した拡張コミュニティ リスト内のルートから範囲内の RT を削除します。 • この例では、拡張コミュニティ リスト 128 内のルートから範囲内の RT が削除されます。
ステップ 8	**set extcommunity rt** *value* **additive** 例： `Device(config-route-map)# set extcommunity rt 400:1 additive`	ルートマップで許可されているルートに、指定した RT を設定します。 • この例では、拡張コミュニティ 128 に一致するルートに RT 400:1 を設定します。**additive** キーワードを指定すると、RT を置き換えずに RT が RT リストに追加されます。
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、グローバル 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 rtmap2 permit 20`	（任意）ルートを許可するルートマップ エントリを設定します。 • この例では、RT 範囲から RT へのマッピングの対象とならない他のルートを許可するルートマップ エントリを設定します。この手順を実行しない場合、他のすべてのルートは暗黙の deny の対象となります。
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** *as-number* 例： `Device(config)# router bgp 4000`	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* **remote-as** *autonomous-system-number* 例：	自律システムに属するネイバーを指定します。

	コマンドまたはアクション	目的
	`Device(config-router)# neighbor 192.168.0.50 remote-as 4000`	
ステップ 14	**address-family vpnv4** 例： `Device(config-router-af)# address-family vpnv4`	アドレス ファミリ コンフィギュレーション モードを開始して、アドレス ファミリ固有の設定を受け入れるよう BGP ピアを設定します。
ステップ 15	**neighbor** *ip-address* **activate** 例： `Device(config-router-af)# neighbor 192.168.0.50 activate`	指定したネイバーをアクティブにします。
ステップ 16	**neighbor** *ip-address* **route-map** *map-tag* **in** 例： `Device(config-router-af)# neighbor 192.168.0.50 route-map rtmap2 in`	指定した着信ルートマップを指定したネイバーに適用します。
ステップ 17	**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 モードを終了して、特権 EXEC モードを開始します。

BGP-RT および VPN 識別子属性の書き換えワイルドカードの設定例

BGP-RT および VPN 識別子属性の書き換えワイルドカードの設定例

例：RT を VPN 識別子範囲に置き換える

次の例では、出力 ASBR で、RT 201:100 を持つルートが拡張コミュニティ リスト 22 に含まれます。rt-mapping という名前のルート マップで拡張コミュニティ リスト 22 を照合し、コミュニティ リスト内のルートから RT を削除します。コミュニティ リス

トに一致するルートに対して、600:1 ～ 600:8 の範囲内の VPN 識別子を設定します。このルート マップはネイバー 192.168.103.1 に適用されます。

出力 ASBR

```
ip extcommunity-list 22 permit rt 201:100
!
route-map rt-mapping permit 10
 match extcommunity 22
 set extcomm-list 22 delete
 set extcommunity vpn-distinguisher range 600:1 600:8
!
route-map rt-mapping permit 20
!
router bgp 3000
 neighbor 192.168.103.1 remote-as 3000
 address-family vpnv4
  neighbor 192.168.103.1 activate
  neighbor 192.168.103.1 route-map rt-mapping out
  exit-address-family
!
```

入力 ASBR では、600:1 ～ 600:8 の範囲内の VPN 識別子が拡張コミュニティ リスト 101 に属します。rtmap2 という名前のルート マップで、これらの VPN 識別子を RT 範囲 700:1 ～ 700:10 にマッピングします。このルート マップはネイバー 192.168.0.50 に適用されます。additive オプションを指定すると、新しい範囲が既存の値に置き換えなしに追加されます。

入力 ASBR

```
ip extcommunity-list 101 permit VD:600:[1-8]
!
route-map rtmap2 permit 10
 match extcommunity 101
 set extcomm-list 101 delete
 set extcommunity rt 700:1 700:10 additive
!
route-map rtmap2 permit 20
!
router bgp 4000
 neighbor 192.168.0.50 remote-as 4000
 address-family vpnv4
  neighbor 192.168.0.50 activate
  neighbor 192.168.0.50 route-map rtmap2 in
  exit-address-family
!
```

BGP-RT および VPN 識別子属性の書き換えワイルドカードに関する追加情報

関連資料

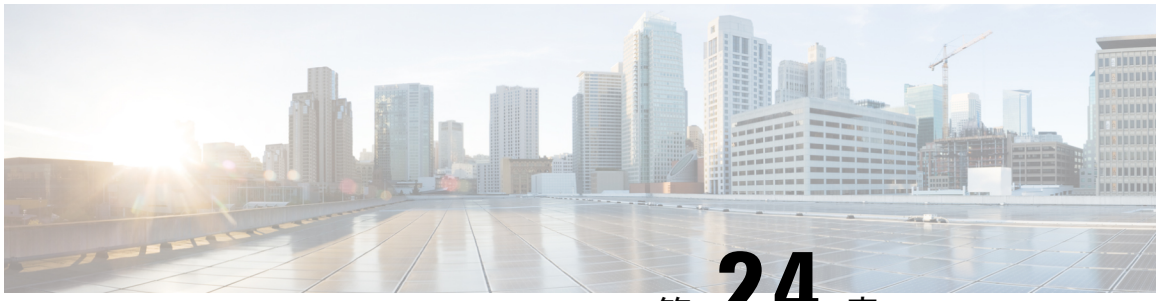
関連項目	マニュアル タイトル
BGP コマンド	『*Cisco IOS IP Routing: BGP Command Reference*』
BGP—VPN 識別子属性	『*IP : BGP Configuration Guide, XE 3S*』の「BGP—VPN Distinguisher Attribute」モジュール

BGP—RT および VPN 識別子属性の書き換えワイルドカードに関する機能情報

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

表 *36 : BGP—RT* および *VPN* 識別子属性の書き換えワイルドカードに関する機能情報

リリース	機能情報
Cisco IOS XE Gibraltar 16.10.1	この機能が導入されました。



第 24 章

4 バイト ASN に対する BGP サポートの設定

- 4 バイト ASN に対する BGP サポートに関する情報 （283 ページ）
- 4 バイト ASN に対する BGP サポートの設定方法 （287 ページ）
- 4 バイト ASN に対する BGP サポートの設定例 （294 ページ）
- 4 バイト ASN に対する BGP サポートに関する追加情報 （299 ページ）
- 4 バイト ASN に対する BGP サポートの機能履歴と機能情報 （299 ページ）

4 バイト ASN に対する BGP サポートに関する情報

BGP 自律システム番号の形式

RFC 4271『*A Border Gateway Protocol 4 (BGP-4)*』に記述されているように、2009 年 1 月まで、企業に割り当てられていた BGP 自律システム（AS）番号は 1 ～ 65535 の範囲の 2 オクテットの数値でした。現在は、AS 番号の需要増加に伴い、Internet Assigned Numbers Authority（IANA）によって割り当てられる AS 番号は 65536 ～ 4294967295 の範囲の 4 オクテットの番号になりました。RFC 5396『*Textual Representation of Autonomous System (AS) Numbers*』には、AS 番号を表す 3 つの方式が記述されています。シスコでは、次の 2 つの方式を実装しています。

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

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

asdot だけを使用する自律システム番号形式

4 オクテット（4 バイト）の AS 番号は asdot 表記法だけで入力および表示されます。たとえば、1.10 または 45000.64000 です。4 バイト AS 番号のマッチングに正規表現を使用する場合、

asdot 形式には正規表現で特殊文字となるピリオドが含まれていることに注意します。正規表現でのマッチングに失敗しないよう、（1\.14 のように）ピリオドの前にバックスラッシュを入力する必要があります。次の表は、asdot 形式だけが使用できる Cisco IOS イメージで、2 バイトおよび 4 バイト AS 番号の設定、正規表現とのマッチング、および **show** コマンド出力での表示に使用される形式をまとめたものです。

表 *37 : asdot* だけを使用する *4* バイト *AS* 番号形式

書式	設定形式	showコマンド出力および正規表現のマッチング形式
asdot	2 バイト：1 ～ 655354 バイト：1.0 ～ 65535.65535	2 バイト：1 ～ 655354 バイト：1.0 ～ 65535.65535

asplain をデフォルトとする AS 番号形式

シスコ実装の 4 バイト AS 番号では asplain がデフォルトの AS 番号表示形式として使用されていますが、4 バイト AS 番号は asplain および asdot 形式のどちらにも設定できます。また、正規表現で 4 バイト AS 番号とマッチングするためのデフォルト形式は asplain であるため、4 バイト AS 番号とマッチングする正規表現はすべて、asplain 形式で記述する必要があります。デフォルトの **show** コマンド出力を変更して、4 バイトの自律システム番号を asdot 形式で表示する場合は、ルータ コンフィギュレーション モードで **bgp asnotation dot** コマンドを使用します。デフォルトで asdot 形式が有効にされている場合、正規表現の 4 バイト AS 番号のマッチングには、すべて asdot 形式を使用する必要があり、使用しない場合正規表現によるマッチングは失敗します。次の表に示すように、4 バイト AS 番号は asplain と asdot のどちらにも設定できますが、**show** コマンド出力と正規表現を使用した 4 バイト AS 番号のマッチング制御には 1 つの形式だけが使用されます。デフォルトは asplain 形式です。**show** コマンド出力の表示と正規表現のマッチング制御で asdot 形式の 4 バイト AS 番号を使用する場合、**bgp asnotation dot** コマンドを設定する必要があります。**bgp asnotation dot** コマンドを有効にした後、**clear ip bgp *** コマンドを入力してすべての BGP セッションに対してハード リセットを開始する必要があります。

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

表 *38 : asplain* をデフォルトとする *4* バイト *AS* 番号形式

書式	設定形式	show コマンド出力および正規表現のマッチング形式
asplain	2 バイト：1 ～ 655354 バイト：65536 ～ 4294967295	2 バイト：1 ～ 655354 バイト：65536 ～ 4294967295

書式	設定形式	show コマンド出力および正規表現のマッチング形式
asdot	2 バイト：1 ～ 655354 バイト：1.0 ～ 65535.65535	2 バイト：1 ～ 655354 バイト：65536 ～ 4294967295

表 *39 : asdot* を使用する *4* バイト *AS* 番号形式

書式	設定形式	show コマンド出力および正規表現のマッチング形式
asplain	2 バイト：1 ～ 655354 バイト：65536 ～ 4294967295	2 バイト：1 ～ 655354 バイト：1.0 ～ 65535.65535
asdot	2 バイト：1 ～ 655354 バイト：1.0 ～ 65535.65535	2 バイト：1 ～ 655354 バイト：1.0 ～ 65535.65535

予約済みおよびプライベートの AS 番号

シスコが採用している BGP は、RFC 4893 をサポートしています。RFC 4893 は、2 バイト AS 番号から 4 バイト AS 番号への段階的移行を BGP がサポートできるように開発されました。新しい予約済み（プライベート）AS 番号（23456）は RFC 4893 により作成された番号で、Cisco IOS CLI ではこの番号を AS 番号として設定できません。

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

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

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

シスコが採用している 4 バイト自律システム番号

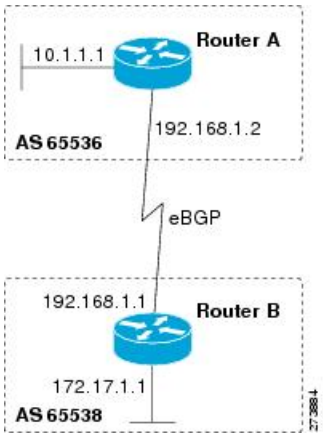
シスコが採用している 4 バイト自律システム（AS）番号は、AS 番号の正規表現のマッチングおよび出力表示形式のデフォルトとして asplain（たとえば、65538）を使用していますが、RFC 5396 に記載されているとおり、4 バイト AS 番号を asplain 形式および asdot 形式の両方で設定できます。4 バイト AS 番号の正規表現マッチングと出力表示のデフォルトを asdot 形式に変更するには、**bgp asnotation dot** コマンドの後に **clear ip bgp *** コマンドを実行し、現在の BGP セッションをすべてハード リセットします。4 バイト AS 番号形式の詳細については、「BGP 自律システム番号の形式」の項を参照してください。

シスコが採用している 4 バイト AS 番号は、設定形式、正規表現とのマッチング、および出力表示として、asdot（たとえば、1.2）だけを使用しています。asplain はサポートしていません。4 バイト番号を使用する 2 つの自律システム内の BGP ピアの例については、下の図を参照してください。asdot 表記法を使用して設定された、異なる 4 バイトの自律システムにある 3 つのネイバー ピアの間での設定例については、「例：BGP ルーティング プロセスと 4 バイト自律システム番号を使用したピアの設定」を参照してください。

シスコは、BGP が 2 バイト AS 番号から 4 バイト AS 番号へ段階的に移行できるように開発された RFC 4893 もサポートしています。スムーズな移行を確実に行うには、4 バイト AS 番号を使用して識別される AS 内の BGP スピーカーをすべて、4 バイト AS 番号をサポートするようにアップグレードすることを推奨します。

（注） 新しいプライベート AS 番号（23456）は RFC 4893 により作成された番号で、Cisco IOS CLI ではこの番号を AS 番号として設定できません。

図 ***11 : 4*** バイト番号を使用する ***2*** つの自律システム内の ***BGP*** ピア



4 バイト ASN に対する BGP サポートの設定方法

BGP ルーティング プロセスと 4 バイト自律システム番号を使用したピアの設定

4 バイト自律システム（AS）番号を使用する AS にボーダー ゲートウェイ プロトコル（BGP）ピアが配置されているときに、BGP ルーティング プロセスおよび BGP ピアを設定するには、この作業を実行します。ここで設定するアドレス ファミリは、デフォルトの IPv4 ユニキャスト アドレス ファミリで、設定は上の図（「シスコが採用している 4 バイト自律システム番号」の項）のルータ A で行われています。この作業にある 4 バイト AS 番号は、デフォルトの asplain（10 進数値）形式にフォーマットされています。たとえば、上の図にあるルータ B の AS 番号は 65538 です。BGP ピアとなりうるネイバー ルータすべてについて、必ず、この作業を実行してください。

始める前に

（注） デフォルトでは、ルータ コンフィギュレーション モードで **neighbor remote-as** コマンドを使用して定義したネイバーは、IPv4 ユニキャスト アドレス プレフィックスだけを交換します。IPv6 プレフィックスなど、その他のアドレス プレフィックス タイプを交換するには、そのプレフィックス タイプについて、アドレス ファミリ コンフィギュレーション モードで **neighbor activate** コマンドを使用してネイバーをアクティブ化する必要もあります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router bgp** *autonomous-system-number* 例： `Device(config)# router bgp 65538`	指定したルーティング プロセスのルータ コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
		• この例では、4 バイト AS 番号 65538 は asplain 表記法で定義されています。
ステップ 4	**neighbor** *ip-address* **remote-as** *autonomous-system-number* 例： `Device(config-router)# neighbor 192.168.1.2 remote-as 65536`	指定された AS のネイバーの IP アドレスを、ローカル デバイスの IPv4 マルチプロトコル BGP ネイバー テーブルに追加します。 • この例では、4 バイト AS 番号 65536 は asplain 表記法で定義されています。
ステップ 5	必要に応じて、手順 4 を繰り返し、その他の BGP ネイバーを定義します。	--
ステップ 6	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] 例： `Device(config-router)# address-family ipv4 unicast`	IPv4 アドレス ファミリを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。 • **unicast** キーワードは、IPv4 ユニキャスト アドレス ファミリを指定します。デフォルトでは、**address-family ipv4** コマンドに **unicast** キーワードが指定されていない場合、デバイスは IPv4 ユニキャスト アドレス ファミリのコンフィギュレーション モードになります。 • **multicast** キーワードは、IPv4 マルチキャスト アドレス プレフィックスを指定します。 • **vrf** キーワードおよび *vrf-name* 引数では、後続の IPv4 アドレス ファミリ コンフィギュレーション モード コマンドに関連付ける Virtual Routing and Forwarding（VRF; 仮想ルーティングおよび転送）インスタンスの名前を指定します。
ステップ 7	**neighbor** *ip-address* **activate** 例：	ネイバーが IPv4 ユニキャスト アドレス ファミリのプレフィックスをローカル デバイスと交換できるようにします。

	コマンドまたはアクション	目的
	`Device(config-router-af)# neighbor 192.168.1.2 activate`	
ステップ 8	必要に応じて、手順 7 を繰り返し、その他の BGP ネイバーをアクティブ化します。	--
ステップ 9	**network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*] 例： `Device(config-router-af)# network 172.17.1.0 mask 255.255.255.0`	（任意）この AS にローカルとしてネットワークを指定し、BGP ルーティング テーブルに追加します。 • 外部プロトコルの場合、**network** コマンドはアドバタイズされるネットワークを制御します。内部プロトコルは **network** コマンドを使用して、アップデートの送信先を決定します。
ステップ 10	**end** 例： `Device(config-router-af)# end`	アドレス ファミリ コンフィギュレーション モードを終了して、特権 EXEC モードに戻ります。
ステップ 11	**show ip bgp** [*network*] [*network-mask*] 例： `Device# show ip bgp 10.1.1.0`	（任意）BGP ルーティング テーブル内のエントリを表示します。 （注） この例では、このタスクに適用可能な構文だけが使用されています。詳細については、『*Cisco IOS IP Routing: BGP Command Reference*』を参照してください。
ステップ 12	**show ip bgp summary** 例： `Device# show ip bgp summary`	（任意）BGP 接続すべての状況を表示します。

次の例は、上の図のルータ B で実行された **show ip bgp** コマンドの出力ですが、ここにはルータ A で 192.168.1.2 にある BGP ネイバーから学習されたネットワーク 10.1.1.0 に対する BGP ルーティング テーブル エントリと、デフォルトの asplain 形式で表した 4 バイト AS 番号 65536 が表示されています。

```
RouterB# show ip bgp 10.1.1.0
```

```
BGP routing table entry for 10.1.1.0/24, version 2
Paths: (1 available, best #1)
Advertised to update-groups:
2
65536
192.168.1.2 from 192.168.1.2 (10.1.1.99)
Origin IGP, metric 0, localpref 100, valid, external, best
```

次の例は、**show ip bgp summary** コマンドの出力ですが、ここには、上の図のルータ B でこの作業を設定した後で、ルータ A にある BGP ネイバー 192.168.1.2 の 4 バイト AS 番号が 65536 であることが表示されています。

```
RouterB# show ip bgp summary

BGP router identifier 172.17.1.99, local AS number 65538
BGP table version is 3, main routing table version 3
2 network entries using 234 bytes of memory
2 path entries using 104 bytes of memory
3/2 BGP path/bestpath attribute entries using 444 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 806 total bytes of memory
BGP activity 2/0 prefixes, 2/0 paths, scan interval 60 secs
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  Stated
192.168.1.2     4       65536       6       6        3    0    0 00:01:33       1
```

4 バイト自律システム番号で使用される出力および正規表現とのマッチング形式のデフォルトを変更

4 バイト自律システム（AS）番号のデフォルト出力形式を asplain 形式から asdot 表記法形式に変更するには、この作業を実行します。4 バイト AS 番号の出力形式の変化を表示するには、**show ip bgp summary** コマンドを使用します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**show ip bgp summary** 例： `Device# show ip bgp summary`	すべてのボーダーゲートウェイ プロトコル（BGP）接続のステータスを表示します。
ステップ 3	**configure terminal** 例：	グローバル 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 4	**router bgp** *autonomous-system-number* 例： `Device(config)# router bgp 65538`	指定したルーティング プロセスのルータ コンフィギュレーション モードを開始します。 • この例では、4 バイト AS 番号 65538 は asplain 表記法で定義されています。
ステップ 5	**bgp asnotation dot** 例： `Device(config-router)# bgp asnotation dot`	BGP 4 バイト AS 番号のデフォルト出力形式を asplain（10 進数値）からドット表記法に変更します。 （注） 4 バイト AS 番号は、asplain 形式、または asdot 形式を使用して設定できます。このコマンドの影響を受けるのは、**show** コマンドの出力、または正規表現のマッチングだけです。
ステップ 6	**end** 例： `Device(config-router)# end`	アドレス ファミリ コンフィギュレーション モードを終了して、特権 EXEC モードに戻ります。
ステップ 7	**clear ip bgp *** 例： `Device# clear ip bgp *`	現在の BGP セッションをすべてクリアし、リセットします。 • この例では、4 バイト AS 番号形式の変更がすべての BGP セッションに反映されていることを確認するために、ハードリセットが実行されています。 （注） この例では、このタスクに適用可能な構文だけが使用されています。詳細については、『*Cisco IOS IP Routing: BGP Command Reference*』を参照してください。
ステップ 8	**show ip bgp summary** 例：	BGP 接続すべての状況を表示します。

	コマンドまたはアクション	目的
	`Device# show ip bgp summary`	
ステップ 9	**show ip bgp regexp** *regexp* 例： `Device# show ip bgp regexp ^1\.0$`	AS パスの正規表現と一致するルートを表示します。 • この例では、4 バイトの AS パスをマッチングする正規表現は、asdot 形式で設定されています。
ステップ 10	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 11	**router bgp** *autonomous-system-number* 例： `Device(config)# router bgp 65538`	指定したルーティング プロセスのルータ コンフィギュレーション モードを開始します。 • この例では、4 バイト AS 番号 65538 は asplain 表記法で定義されています。
ステップ 12	**no bgp asnotation dot** 例： `Device(config-router)# no bgp asnotation dot`	BGP 4 バイト AS 番号のデフォルト出力形式を asplain（10 進数値）にリセットします。 （注） 4 バイト AS 番号は、asplain 形式、または asdot 形式を使用して設定できます。このコマンドの影響を受けるのは、**show** コマンドの出力、または正規表現のマッチングだけです。
ステップ 13	**end** 例： `Device(config-router)# end`	ルータ コンフィギュレーション モードを終了して、特権 EXEC モードに戻ります。
ステップ 14	**clear ip bgp *** 例： `Device# clear ip bgp *`	現在の BGP セッションをすべてクリアし、リセットします。 • この例では、4 バイト AS 番号形式の変更がすべての BGP セッションに反映されていることを確認する

	コマンドまたはアクション	目的
		ために、ハードリセットが実行されています。 （注） この例では、このタスクに適用可能な構文だけが使用されています。詳細については、『*Cisco IOS IP Routing: BGP Command Reference*』を参照してください。

例

次の **show ip bgp summary** コマンドの出力は、4 バイト AS 番号のデフォルト asplain 形式を示しています。ここで、asplain 形式で表された 4 バイト AS 番号 65536 および 65550 に注意してください。

```
Router# show ip bgp summary

BGP router identifier 172.17.1.99, local AS number 65538
BGP table version is 1, main routing table version 1
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  Statd
192.168.1.2     4        65536       7       7        1    0    0 00:03:04      0
192.168.3.2     4        65550       4       4        1    0    0 00:00:15      0
```

bgp asnotation dot コマンドの設定後（これに、現在の BGP セッションをすべてハードリセットする **clear ip bgp *** コマンドが続きます）、出力は、次の **show ip bgp summary** コマンドの出力に示すように、asdot 表記法の形式に変換されます。asdot 形式で表された 4 バイト AS 番号 1.0 および 1.14 に注意してください。これらは AS 番号 65536 と 65550 を asdot 変換したものです。

```
Router# show ip bgp summary

BGP router identifier 172.17.1.99, local AS number 1.2
BGP table version is 1, main routing table version 1
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  Statd
192.168.1.2     4         1.0       9       9        1    0    0 00:04:13      0
192.168.3.2     4        1.14       6       6        1    0    0 00:01:24      0
```

bgp asnotation dot コマンドの設定後（これに、現在の BGP セッションをすべてハードリセットする **clear ip bgp *** コマンドが続きます）、4 バイトの AS パスで使用される正規表現とのマッチング形式は asdot 表記法の形式に変更されます。4 バイト AS 番号は、asplain 形式または asdot 形式のいずれかを使用して、正規表現で設定できますが、現在のデフォルト形式を使用して設定された 4 バイト AS 番号だけがマッチングされます。下の先頭の例では、**show ip bgp regexp** コマンドは、asplain 形式で表された 4 バイト AS 番号を使って設定されています。現在のデフォルト形式は asdot 形式なのでマッチングは失敗し、何も出力されません。asdot 形式を使用した 2 番目の例では、

マッチングは成功し、4 バイトの AS パスに関する情報が asdot 表記法を使って表示されます。

（注） このasdot表記法で使用されているピリオドは、シスコの正規表現では特殊文字です。特殊な意味を取り除くには、ピリオドの前にバックスラッシュをつけます。

```
Router# show ip bgp regexp ^65536$

Router# show ip bgp regexp ^1\.0$

BGP table version is 2, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.1.0/24      192.168.1.2              0             0 1.0 i
```

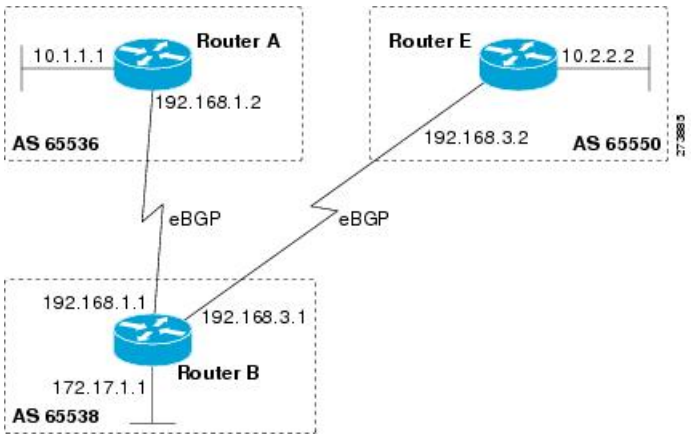
4 バイト ASN に対する BGP サポートの設定例

例：BGP ルーティング プロセスと 4 バイト自律システム番号を使用したピアの設定

asplain 形式

次に示すのは、下の図におけるボーダー ゲートウェイ プロトコル（BGP）プロセスを使ったルータ A、B、E のコンフィギュレーションの例で、このプロセスは、asplain 表記法を使用して設定された別々の 4 バイト自律システムのルータ A、B、E にある 3 つのネイバー ピアの間に設定されています。IPv4 ユニキャスト ルートはすべてのピアと交換されます。

図 *12 : asplain* 形式の *4* バイト自律システム番号を使用する *BGP* ピア



ルータ A

```
router bgp 65536
bgp router-id 10.1.1.99
no bgp default ipv4-unicast
bgp fast-external-fallover
bgp log-neighbor-changes
timers bgp 70 120
neighbor 192.168.1.1 remote-as 65538
!
address-family ipv4
neighbor 192.168.1.1 activate
no auto-summary
no synchronization
network 10.1.1.0 mask 255.255.255.0
exit-address-family
```

ルータ B

```
router bgp 65538
bgp router-id 172.17.1.99
no bgp default ipv4-unicast
bgp fast-external-fallover
bgp log-neighbor-changes
timers bgp 70 120
neighbor 192.168.1.2 remote-as 65536
neighbor 192.168.3.2 remote-as 65550
neighbor 192.168.3.2 description finance
!
address-family ipv4
neighbor 192.168.1.2 activate
neighbor 192.168.3.2 activate
no auto-summary
no synchronization
network 172.17.1.0 mask 255.255.255.0
exit-address-family
```

ルータ E

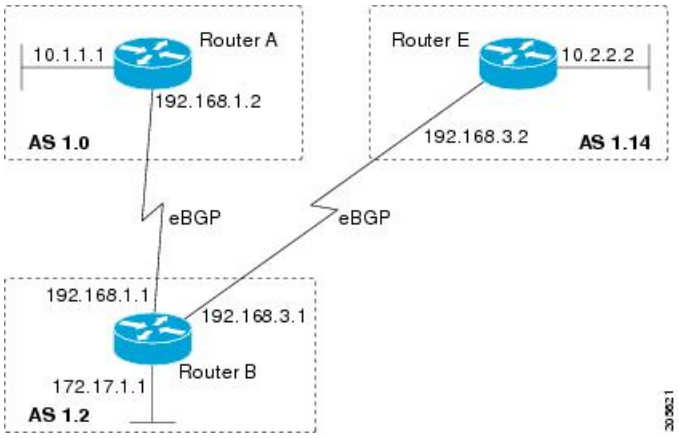
```
router bgp 65550
bgp router-id 10.2.2.99
no bgp default ipv4-unicast
bgp fast-external-fallover
bgp log-neighbor-changes
timers bgp 70 120
neighbor 192.168.3.1 remote-as 65538
!
address-family ipv4
neighbor 192.168.3.1 activate
no auto-summary
no synchronization
network 10.2.2.0 mask 255.255.255.0
exit-address-family
```

asdot 形式

次に示すのは、下の図における BGP プロセスを使ったルータ A、B、E のコンフィギュレーションを作成する方法の例で、このプロセスは、デフォルトの asdot 形式を使用して設定され

た別々の 4 バイト自律システムのルータ A、B、E にある 3 つのネイバー ピアの間に設定されています。IPv4 ユニキャスト ルートはすべてのピアと交換されます。

図 *13 : asdot* 形式の *4* バイト自律システム番号を使用する *BGP* ピア



ルータ A

```
router bgp 1.0
bgp router-id 10.1.1.99
no bgp default ipv4-unicast
bgp fast-external-fallover
bgp log-neighbor-changes
timers bgp 70 120
neighbor 192.168.1.1 remote-as 1.2
!
address-family ipv4
neighbor 192.168.1.1 activate
no auto-summary
no synchronization
network 10.1.1.0 mask 255.255.255.0
exit-address-family
```

ルータ B

```
router bgp 1.2
bgp router-id 172.17.1.99
no bgp default ipv4-unicast
bgp fast-external-fallover
bgp log-neighbor-changes
timers bgp 70 120
neighbor 192.168.1.2 remote-as 1.0
neighbor 192.168.3.2 remote-as 1.14
neighbor 192.168.3.2 description finance
!
address-family ipv4
neighbor 192.168.1.2 activate
neighbor 192.168.3.2 activate
no auto-summary
no synchronization
network 172.17.1.0 mask 255.255.255.0
exit-address-family
```

ルータ E

```
router bgp 1.14
bgp router-id 10.2.2.99
no bgp default ipv4-unicast
bgp fast-external-fallover
bgp log-neighbor-changes
timers bgp 70 120
neighbor 192.168.3.1 remote-as 1.2
!
address-family ipv4
neighbor 192.168.3.1 activate
no auto-summary
no synchronization
network 10.2.2.0 mask 255.255.255.0
exit-address-family
```

例：4バイトのBGP自律システム番号を使用したVRFおよび拡張コミュニティの設定

次に、4 バイト自律システム番号 65537 を使用するルートターゲットを使って VRF を作成する方法、およびルートターゲットに、ルートマップにより許可されたルートの拡張コミュニティ値 65537:100 を設定する例を示します。

```
ip vrf vpn_red
rd 64500:100
route-target both 65537:100
exit
route-map red_map permit 10
set extcommunity rt 65537:100
end
```

コンフィギュレーションの完了後、**show route-map** コマンドを使用して、拡張コミュニティが、4 バイト自律システム番号 65537 を含むルート ターゲットに設定されていることを確認します。

```
RouterB# show route-map red_map
route-map red_map, permit, sequence 10
Match clauses:
Set clauses:
extended community RT:65537:100
Policy routing matches: 0 packets, 0 bytes
```

4 バイト自律システム番号の RD サポート

次の例は、4 バイト AS 番号 65536 を含むルート識別子、および 4 バイト自律システム番号 65537 を含むルート ターゲットを使用して、VRF を作成する方法を示しています。

```
ip vrf vpn_red
rd 65536:100
route-target both 65537:100
exit
```

コンフィギュレーションの完了後、**show vrf** コマンドを使用して、4 バイト AS 番号ルート識別子が 65536:100 に設定されていることを確認します。

```
RouterB# show vrf vpn_red
Current configuration : 36 bytes
vrf definition x
rd 65536:100
!
```

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

次に、4 バイト自律システム番号 1.1 を使用するルートターゲットを使って VRF を作成する方法、およびルートターゲットに、ルートマップにより許可されたルートの拡張コミュニティ値 1.1:100 を設定する例を示します。

（注） 次の例が正常に動作するのは、**bgp asnotation dot** コマンドを使用して asdot をデフォルトの表示形式として設定した場合です。

```
ip vrf vpn_red
rd 64500:100
route-target both 1.1:100
exit
route-map red_map permit 10
set extcommunity rt 1.1:100
end
```

コンフィギュレーションの完了後、**show route-map** コマンドを使用して、拡張コミュニティが、4 バイト自律システム番号 1.1 を含むルート ターゲットに設定されていることを確認します。

```
RouterB# show route-map red_map
route-map red_map, permit, sequence 10
Match clauses:
Set clauses:
extended community RT:1.1:100
Policy routing matches: 0 packets, 0 bytes
```

4 バイト自律システム番号の RD サポートの asdot デフォルト形式

次の例が正常に動作するのは、**bgp asnotation dot** コマンドを使用して asdot をデフォルトの表示形式として設定した場合です。

```
ip vrf vpn_red
rd 1.0:100
route-target both 1.1:100
exit
```

4 バイト ASN に対する BGP サポートに関する追加情報

関連資料

関連項目	マニュアル タイトル
BGP コマンド	『*Cisco IOS IP Routing: BGP Command Reference*』

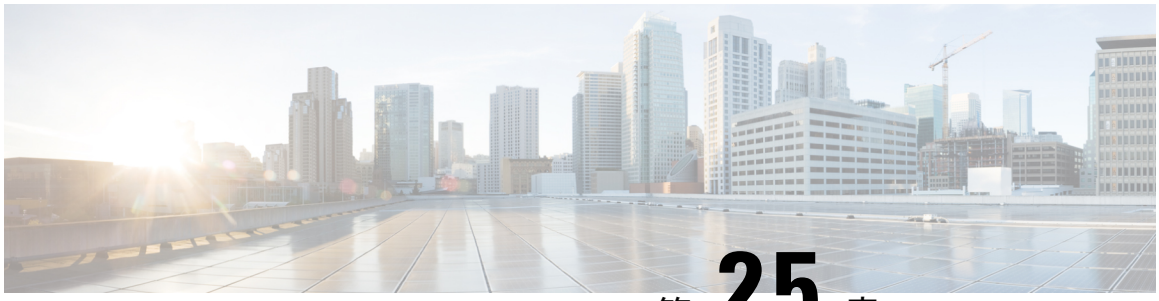
標準および RFC

標準/RFC	タイトル
RFC 4893	『*BGP Support for Four-octet AS Number Space*』
RFC 5396	『*Textual Representation of Autonomous System (AS) Numbers*』
RFC 5398	『*Autonomous System (AS) Number Reservation for Documentation Use*』
RFC 5668	『*4-Octet AS Specific BGP Extended Community*』

4 バイト ASN に対する BGP サポートの機能履歴と機能情報

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

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



第 25 章

BGP ネクストホップ非変更の設定

外部 BGP（eBGP）セッションでは、デフォルトで、ルータがルートの送信時に BGP ルートのネクスト ホップ属性を（自身のアドレスに）変更します。BGP ネクスト ホップ非変更機能では、ネクスト ホップ属性を変更せずに BGP によって eBGP マルチホップ ピアにアップデートを送信できます。

- BGP ネクストホップ非変更に関する制約事項 （301 ページ）
- BGP ネクスト ホップ非変更 （301 ページ）
- BGP ネクスト ホップ非変更の設定方法 （302 ページ）
- 例：eBGP ピアの BGP ネクスト ホップ非変更 （305 ページ）
- BGP ネクスト ホップ非変更機能の情報 （305 ページ）

BGP ネクストホップ非変更に関する制約事項

BGP ネクスト ホップ非変更機能は、マルチホップ eBGP ピア間だけで設定できます。直接接続されたネイバーにこの機能を設定しようとすると、次のエラー メッセージが表示されます。

```
%BGP: Can propagate the nexthop only to multi-hop EBGP neighbor
```

BGP ネクスト ホップ非変更

外部 BGP（eBGP）セッションでは、デフォルトで、ルータがルートの送信時に BGP ルートのネクスト ホップ属性を（自身のアドレスに）変更します。BGP ネクスト ホップ非変更機能が設定されている場合、BGP はネクスト ホップ属性を変更せずに eBGP マルチホップ ピアにルートを送信します。ネクスト ホップ属性は変更されません。

（注） ルータがルートを送信するとき、BGP ルートのネクストホップ属性を変更するルータのデフォルト動作の例外があります。ネクスト ホップが eBGP ピアのピアリング アドレスと同じサブネットにある場合、ネクスト ホップは変更されません。これは、サード パーティのネクスト ホップと呼ばれます。

BGP ネクスト ホップ非変更機能により、ネットワークの設計および移行を柔軟に実効できます。これは、マルチホップとして設定された eBGP ピア間だけで使用できます。2 つの自律システム間のさまざまなシナリオで使用できます。たとえば、同じ IGP を共有する複数の自律システムが接続される場合、または少なくともルータに互いのネクストホップに到達するための別の方法がある（このため、ネクストホップを変更しないままにできる）場合などが挙げられます。

この機能の一般的な用途は、RR 間で VPNv4 のマルチホップ MP-eBGP を持つマルチプロトコル ラベル スイッチング（MPLS）Inter-AS を設定することです。

この機能のもう 1 つの一般的な用途は、RFC4364、Section 10 で定義されている VPNv4 Inter-AS オプション C の設定です。この設定では、VPNv4 ルートは、自律システム間で（異なる自律システムの RR 間で）渡されます。RR は複数ホップ離れており、**neighbor next-hop unchanged** が設定されています。異なる自律システムの PE によって、その PE 間に LSP が確立されます（一般的な IGP 経由によって、または ASBR 間のラベル付きルート（1 ホップ離れた異なる自律システムからのルート）経由で PE に接続されたネクストホップのアドバタイズによって）。PE は、LSP 経由で別の AS 内の PE のネクスト ホップに到達でき、したがって VRF RIB に VPNv4 ルートをインストールできます。

BGP ネクスト ホップ非変更の設定方法

次の手順には、BGPネクストホップ非変更を設定する手順が含まれています。

eBGP ピアの BGP ネクスト ホップ非変更の設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router bgp** *as-number* 例： `Device(config)# router bgp 65535`	ルータ コンフィギュレーション モードを開始して、BGP ルーティング プロセスを作成します。

	コマンドまたはアクション	目的
ステップ 4	**address-family** {**ipv4** | **ipv6** | **l2vpn** | **nsap** | **rtfilter** | **vpnv4** | **vpnv6**} 例： `Device(config-router-af)# address-family vpnv4`	アドレス ファミリ コンフィギュレーション モードを開始して、アドレス ファミリ固有の設定を受け入れるように BGP ピアを設定します。
ステップ 5	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* 例： `Device(config-router-af)# neighbor 10.0.0.100 remote-as 65600`	エントリを BGP ネイバー テーブルに追加します。
ステップ 6	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate** 例： `Device(config-router-af)# neighbor 10.0.0.100 activate`	ピアとの情報交換をイネーブルにします。
ステップ 7	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **ebgp-multihop** *ttl* 例： `Device(config-router-af)# neighbor 10.0.0.100 ebgp-multihop 255`	ローカル ルータを設定して、直接接続されていないネットワークに存在する外部ピアとの接続を受け入れて開始するようにします。
ステップ 8	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **next-hop-unchanged** 例： `Device(config-router-af)# neighbor 10.0.0.100 next-hop-unchanged`	ネクストホップ属性を変更せずに指定された eBGP ピアに BGP アップデートを送信するようにルータを設定します。
ステップ 9	**end** 例： `Device(config-router-af)# end`	アドレス ファミリ コンフィギュレーション モードを終了して、特権 EXEC モードを開始します。
ステップ 10	**show ip bgp** 例： `Device# show ip bgp`	（任意）BGP ルーティング テーブルのエントリを表示します。 • 出力には、選択されたアドレスについて **neighbor next-hop-unchanged** コマンドが設定されているかどうかが示されます。

ルートマップを使用した BGP ネクスト ホップ非変更の設定

eBGP ネイバーに対する発信ルートマップの設定

ルートマップを定義し、ネイバーに対する発信ポリシーを適用するには、**set ip next-hop unchanged** コマンドを使用します。

次の設定では、プレフィックス 1.1.1.1 のネクストホップは eBGP ネイバー 15.1.1.2 への送信時に変更されません。

```
enable
config terminal
router bgp 2
 bgp log-neighbor-changes
 neighbor 15.1.1.2 remote-as 3
 neighbor 15.1.1.2 ebgp-multihop 10
 !
 address-family ipv4
  neighbor 15.1.1.2 activate
  neighbor 15.1.1.2 route-map A out
  exit address-family
!
route-map A permit 10
 match ip address 1
 set ip next-hop unchanged
!
access-list 1 permit 1.1.1.1
end
```

eBGP ネイバーへの送信時における iBGP および eBGP パス プレフィックスのネクストホップ非変更の設定

eBGP ネイバーへの送信時に iBGP および eBGP パス プレフィックスのネクストホップを変更しないよう設定するには、**next-hop-unchanged allpaths** コマンドを使用します。

次の設定では、iBGP パス プレフィックスでも eBGP パス プレフィックスでも、ネクストホップは eBGP ネイバー 15.1.1.2 への送信時に変更されません。

```
enable
config terminal
router bgp 2
 bgp log-neighbor-changes
 neighbor 15.1.1.2 remote-as 3
 neighbor 15.1.1.2 ebgp-multihop 10
!
address-family ipv4
 neighbor 15.1.1.2 activate
 neighbor 15.1.1.2 next-hop-unchanged allpaths
 exit address-family
!
end
```

例：eBGP ピアの BGP ネクスト ホップ非変更

次に、リモート AS にマルチホップ eBGP ピア 10.0.0.100 を設定する例を示します。ローカル ルータがそのピアにアップデートを送信する場合、ネクスト ホップ属性を変更せずにアップデートを送信します。

```
router bgp 65535
 address-family ipv4
 neighbor 10.0.0.100 remote-as 65600
 neighbor 10.0.0.100 activate
 neighbor 10.0.0.100 ebgp-multihop 255
 neighbor 10.0.0.100 next-hop-unchanged
 end
```

（注） IPv4、IPv6、VPNv4、VPNv6、L2VPN など、すべてのアドレス ファミリが **next-hop unchanged** コマンドをサポートしています。ただし、アドレス ファミリ L2VPN BGP VPLS シグナリングについては、正常に機能させるためには **next-hop self** コマンドを使用する必要があります。

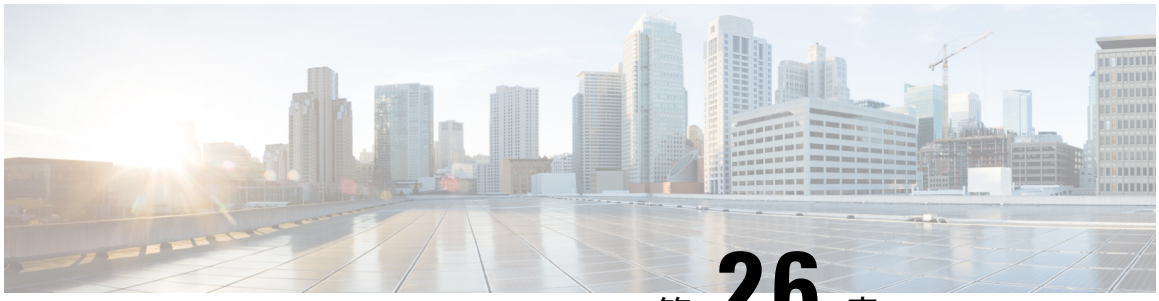
BGP ネクスト ホップ非変更機能の情報

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

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

表 *40 : BGP* ネクスト ホップ非変更機能の情報

機能名	リリース	機能情報
BGP ネクスト ホップ非変更	Cisco IOS XE Gibraltar 16.11.1	BGP ネクスト ホップ非変更機能では、ネクスト ホップ属性を変更せずに BGP によって eBGP マルチホップ ピアにアップデートを送信できます。



第 26 章

IS-IS ルーティングの設定

- IS-IS ルーティングに関する情報 （307 ページ）
- IS-IS の設定方法 （311 ページ）
- IS-IS 認証の設定方法 （323 ページ）
- IS-IS のモニタリングおよびメンテナンス （327 ページ）
- IS-IS の機能情報 （328 ページ）

IS-IS ルーティングに関する情報

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

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

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

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

IS-IS マルチエリア ルーティングでは、シスコの各装置に対して最大 29 個の レベル 1 エリアを定義できますが、レベル 2 ルーティングを実行するプロセスは 1 つだけ設定できます。レベル2ルーティングが任意のプロセス上に設定されている場合、追加のプロセスは、すべて自動的にレベル 1 に設定されます。同時に、このプロセスがレベル 1 ルーティングを実行するように設定することもできます。デバイスインスタンスにレベル2ルーティングが必要でない場合は、 グローバル コンフィギュレーション モードで **is-type** コマンドを使用してレベル 2 の機能を削除します。別のデバイスインスタンスをレベル2デバイスとして設定する場合にも **is-type** コマンドを使用します。

IS-IS 認証

無許可のデバイスがリンクステートデータベースに誤ったルーティング情報を挿入することを防ぐために、インターフェイスごとにプレーンテキストのパスワードを設定するとともにIS-IS エリアごとにエリアパスワードを設定するか、IS-IS 認証を設定することができます。

プレーンテキストのパスワードは、無許可のユーザに対するセキュリティを提供しません。プレーンテキストのパスワードを設定すると、無許可のネットワーキングデバイスがルータと隣接関係を形成することを防ぐことができます。このパスワードはプレーンテキストで交換されるため、アクセスして IS-IS パケットを表示できるエージェントによって参照されます。

新しい IS-IS 認証方式には、プレーンテキストパスワード設定コマンドに比べて次のような利点があります。

- ソフトウェア設定が表示されるときにパスワードが暗号化されます。
- パスワードの管理や変更がより容易になります。
- ネットワークの運用を中断させることなく、新しいパスワードに変更できます。
- 中断なしで認証を移行できます。

認証モード（IS-IS 認証またはプレーンテキストパスワード）は、特定の範囲（IS-IS インスタンスもしくはインターフェイス）またはレベルのいずれかで設定できますが、両方を設定することはできません。ただし、異なる範囲およびレベルに対して、異なるモードを設定することができます。混合モードが設定されている場合は、異なるモードには異なるキーを使用して、プロトコルデータユニット（PDU）で暗号化されたパスワードが危険にさらされないようにする必要があります。

クリアテキスト認証

IS-IS クリアテキスト認証は **area-password** コマンドまたは **domain-password** コマンドによって提供される機能と同じ機能を提供します。

HMAC-MD5 認証

IS-IS は、クリアテキスト認証より安全性の高いメッセージダイジェストアルゴリズム 5（MD5）認証をサポートしています。

ハッシュメッセージ認証コード（HMAC）は暗号学的ハッシュ関数を使用するメッセージ認証符号（MAC）のためのメカニズムです。HMAC-MD5 認証では、各 IS-IS PDU に HMAC-MD5 ダイジェストを追加します。ダイジェストによって、不正なルーティング メッセージがネットワーク ルーティング ドメインに入り込むのを防御できるため、IS-IS ルーティング プロトコル レベルでの認証が可能になります。

HMAC-MD5 認証の利点は次のとおりです。

- パスワードは、ルーティングメッセージを中断させずに新しいパスワードに変更できます。
- 中断なしで認証を移行できます。デバイスは、認証情報のない PDU や古い認証情報を持つ PDU を受け入れ、現在の認証情報を持つ PDU を送信します。このような移行は、認証なしの状態からあるタイプの認証に移行するとき、認証タイプを変更するとき、また認証キーを変更するときに便利です。

HMAC-SHA 認証

IS-IS では、MD5 認証またはクリアテキスト認証よりも安全性の高いセキュア ハッシュ アルゴリズム（SHA）認証（SHA-1、SHA-256、SHA-384、および SHA-512）がサポートされています。

HMAC-SHA 認証方式を有効にすると、共通ネットワークに接続されているすべてのデバイスで共有秘密キーが設定されます。各パケットでは、このキーを使用して、パケットに追加されるメッセージダイジェストを生成および検証します。メッセージダイジェストはパケットおよび秘密キーの単方向機能です。

ヒットレス アップグレード

使用するセキュリティ認証をあるタイプから別のタイプに移行する前に、次の手順を実行する必要があります。

1. すべてのデバイスに、その新しい認証タイプをサポートする新しいイメージをロードする必要があります。デバイスは、すべてのデバイスが新しい認証方式をサポートする新しいイメージでロードされ、さらにすべてのデバイスがその新しい認証方式を使用するように設定されるまで、元の認証方式を使用し続けます。
2. 現在のキーと新しいキーの両方を含むキーチェーンを追加します。たとえば、HMAC-MD5 から HMAC-SHA1-20 に移行する場合、現在のキーは HMAC-MD5 であり、新しいキーは HMAC-SHA1-20 です。IS-IS が現在のキーを送信しつづけるように、現在のキーが新しいキーよりも send-lifetime フィールドの終了日が遅いことを確認してください。IS-IS が両方のキーを受け入れるように、両方のキーの accept-lifetime 値を infinite に設定してください。
3. 手順 2 が完了したら、リンクまたはエリア内のすべてのデバイスについて、現在のキーをキーチェーンから削除できます。

NSF 認識

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

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

IS-IS グローバル パラメータ

次に、設定可能なオプションの IS-IS グローバルパラメータを示します。

- ルートマップによって制御されるデフォルトルートを設定することで、デフォルトルートを IS-IS ルーティングドメイン内に強制的に設定できます。ルートマップで設定可能な、その他のフィルタリングオプションも指定できます。
- 内部チェックサムエラーとともに受信された IS-IS リンクステートパケット（LSP）を無視したり、破損した LSP を消去するようにデバイスを設定できます。これにより、LSP の発信側は、LSP を再生成します。
- エリアおよびドメインにパスワードを割り当てられます。
- ルーティングテーブルでサマリーアドレスによって表される（経路集約に基づいた）集約アドレスを作成できます。他のルーティングプロトコルから学習したルートも集約できます。サマリーをアドバタイズするのに使用されるメトリックは、すべての個別ルートにおける最小のメトリックです。
- 過負荷ビットを設定できます。
- LSP リフレッシュインターバルおよび LSP がリフレッシュなしでデバイスデータベース内にとどまることができる最大時間を設定できます。
- LSP 生成に対するスロットリング タイマー、最短パス優先計算、および部分ルート計算を設定できます。
- IS-IS 隣接関係（アジャセンシー）がステートを変更（アップまたはダウン）する際に、デバイスがログメッセージを生成するように設定できます。
- ネットワーク内のリンクが、1500 バイト未満の最大伝送ユニット（MTU）サイズの場合、それでもルーティングが行われるように LSP MTU の値を低くできます。
- **partition avoidance** コマンドを使用して、レベル 1-2 境界デバイス、隣接レベル 1 デバイス、およびエンドホスト間で完全な接続が失われた場合に、エリアがパーティション化されるのを防ぐことができます。

IS-IS インターフェイス パラメータ

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

設定可能なインターフェイスレベルのパラメータは次のとおりです。

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

IS-IS の設定方法

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

IS-IS のデフォルト設定

表 *41 : IS-IS* のデフォルト設定

機能	デフォルト設定
リンクステート PDU（LSP）エラーを無視	イネーブル
IS-IS タイプ	従来型の IS-IS：ルータは、レベル 1（ステーション）とレベル2（エリア）両方のルータとして機能します。 マルチエリア IS-IS：IS-IS ルーティングプロセスの最初のインスタンスがレベル 1-2 ルータです。残りのインスタンスは、レベル 1 ルータです。
デフォルト情報送信元	ディセーブル
IS-IS 隣接関係のステート変更を記録	ディセーブル
LSP 生成スロットリング タイマー	連続した 2 つのオカレンス間の最大インターバル：5000 ミリ秒 初期 LSP 生成遅延：50 ミリ秒 最初と2番目のLSP生成の間のホールド時間：200 ミリ秒
LSP 最大ライフ タイム（リフレッシュなし）	LSP パケットが削除されるまで 1200 秒（20 分）
LSP リフレッシュ インターバル	900 秒（15 分）ごと
最大 LSP パケット サイズ	1497 バイト
NSF 認識	イネーブルレイヤ3デバイスでは、ハードウェアやソフトウェアの変更中に、隣接するノンストップ フォワーディング対応ルータからのパケットを転送し続けることができます。
部分ルート計算（PRC）スロットリング タイマー	最大 PRC 待機インターバル：5000 ミリ秒 トポロジの変更後の初期PRC計算遅延：50 ミリ秒 最初と 2 番目の PRC 計算の間のホールド時間：200 ミリ秒
パーティション回避	ディセーブル

機能	デフォルト設定
パスワード	エリアまたはドメインのパスワードが定義されておらず、認証はディセーブルになっています。
過負荷ビットの設定	ディセーブル。有効の際に引数が入力されない場合、過負荷ビットがただちに設定され、**no set-overload-bit** コマンドが入力されるまで設定されたままになります。
Shortest Path First（SPF）スロットリング タイマー	連続した SFP 間の最大インターバル：5000 ミリ秒 トポロジの変更後の初期 SFP 計算：200 ミリ秒 最初と2番目のSFP計算の間のホールド時間：50 ミリ秒
サマリー アドレス	ディセーブル

IS-IS ルーティングのイネーブル化

IS-IS をイネーブルにするには、各ルーティングプロセスに名前とネットワーク エンティティ タイトル（NET）を指定します。インターフェイス上でIS-ISルーティングをイネーブルにし、ルーティングプロセスの各インスタンスに対してエリアを指定します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**clns routing** 例： `Device(config)#clns routing`	デバイス上で ISO コネクションレス型ルーティングをイネーブルに設定します。

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

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

IS-IS グローバル パラメータの設定

グローバル IS-IS パラメータを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。

	コマンドまたはアクション	目的
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router isis** 例： `Device(config)#router isis`	IS-IS ルーティング プロトコルを指定し、ルータ コンフィギュレーション モードを開始します。
ステップ 4	**default-information originate** [**route-map** *map-name*] 例： `Device(config-router)#default-information originate route-map map1`	（任意）デフォルトルートを IS-IS ルーティング ドメインに強制的に設定します。**route-map** *map-name* コマンドを入力すると、にルーティングロセスによって有効なルートマップのデフォルトルートが生成されます。
ステップ 5	**ignore-lsp-errors** 例： `Device(config-router)#ignore-lsp-errors`	（任意）LSP を消去する代わりに、内部チェックサムにエラーがある LSP を無視するようにデバイスを設定します。このコマンドは、デフォルトでイネーブルになっています（破損した LSP はドロップされます）。破損した LSP を消去するには、ルータ コンフィギュレーション モードで **no ignore-lsp-errors** コマンドを入力します。
ステップ 6	**area-password** *password* 例： `Device(config-router)#area-password 1password`	（任意）レベル 1（ステーションルータ レベル）LSP に挿入されるエリア認証パスワードを設定します。
ステップ 7	**domain-password** *password* 例： `Device(config-router)#domain-password 2password`	（任意）レベル 2（エリアルータ レベル）LSP に挿入されるルーティング ドメイン認証パスワードを設定します。
ステップ 8	**summary-address** *address mask* [**level-1** | **level-1-2** | **level-2**] 例： `Device(config-router)#summary-address 10.1.0.0 255.255.0.0 level-2`	（任意）所定のレベルのアドレスのサマリーを作成します。

	コマンドまたはアクション	目的
ステップ 9	**set-overload-bit** [**on-startup** {*seconds* | **wait-for-bgp**}] 例： `Device(config-router)#set-overload-bit on-startup wait-for-bgp`	（任意）デバイスに問題がある場合に、他のデバイスが最短パス優先（SPF）計算でこのデバイスを無視するように過負荷ビットを設定します。 • （任意）**on-startup**：スタートアップ時だけ過負荷ビットを設定します。**on-startup** が指定されない場合、過負荷ビットが即座に設定され、**no set-overload-bit** コマンドを入力するまで設定されたままになります。**on-startup** が指定されている場合は、秒数または **wait-for-bgp** のどちらかを入力する必要があります。 • *seconds*：**on-startup** キーワードが設定されている場合、システム起動時に過負荷ビットが設定されて、指定した秒数の間設定されたままになります。指定できる範囲は 5 ～ 86400 秒です。 • **wait-for-bgp**：**on-startup** キーワードが設定されている場合、過負荷ビットがシステム起動時に設定され、BGP が収束するまで設定されたままになります。BGP が収束されたことが IS-IS に通知されない場合、IS-IS は 10 分後に過負荷ビットをオフにします。
ステップ 10	**lsp-refresh-interval** *seconds* 例： `Device(config-router)#lsp-refresh-interval 1080`	（任意）LSP リフレッシュインターバル（秒）を設定します。範囲は 1 ～ 65535 秒です。デフォルトでは、LSP リフレッシュを 900 秒（15 分）ごとに送信します。
ステップ 11	**max-lsp-lifetime** *seconds* 例： `Device(config-router)#max-lsp-lifetime 1000`	（任意）LSP パケットがリフレッシュされずにルータ データベース内に存続する最大時間を設定します。範囲は 1 ～ 65535 秒です。デフォルト値は 1200 秒（20 分）です。指定された時間間隔のあと、LSP パケットは削除されます。

	コマンドまたはアクション	目的
ステップ 12	**lsp-gen-interval** [**level-1** | **level-2**] *lsp-max-wait* [*lsp-initial-wait lsp-second-wait*] 例： `Device(config-router)#lsp-gen-interval level-2 2 50 100`	（任意）IS-IS 生成スロットリング タイマーを設定します。 • *lsp-max-wait*：生成される LAP の連続した 2 つのオカレンス間の最大インターバル（ミリ秒）。指定できる範囲は 1 ～ 120 ミリ秒です。デフォルト値は 5000 ミリ秒です。 • *lsp-initial-wait*：最初の LSP 生成遅延（ミリ秒）。指定できる範囲は 1 ～ 10000 ミリ秒です。デフォルト値は 50 ミリ秒です。 • *lsp-second-wait*：最初と 2 番目の LSP 生成間（ミリ秒）のホールド時間。指定できる範囲は 1 ～ 10000 ミリ秒です。デフォルト値は 200 ミリ秒です。
ステップ 13	**spf-interval** [**level-1** | **level-2**] *spf-max-wait* [*spf-initial-wait spf-second-wait*] 例： `Device(config-router)#spf-interval level-2 5 10 20`	（任意）IS-IS SPF スロットリングタイマーを設定します。 • *spf-max-wait*：連続する SFP 間（ミリ秒）の最大インターバル。指定できる範囲は 1 ～ 120 ミリ秒です。デフォルト値は 5000 ミリ秒です。 • *spf-initial-wait*：トポロジ変更後の最初の SFP 計算（ミリ秒）。指定できる範囲は 1 ～ 10000 ミリ秒です。デフォルト値は 50 ミリ秒です。 • *spf-second-wait*：最初と 2 番目の SFP 計算間（ミリ秒）のホールド時間。指定できる範囲は 1 ～ 10000 ミリ秒です。デフォルト値は 200 ミリ秒です。
ステップ 14	**prc-interval** *prc-max-wait* [*prc-initial-wait prc-second-wait*] 例：	（任意）IS-IS PRC スロットリングタイマーを設定します。 • *prc-max-wait*：2 つの連続する PRC 計算間の最大インターバル（ミリ

	コマンドまたはアクション	目的
	`Device(config-router)#prc-interval 5 10 20`	秒）。指定できる範囲は 1 ～ 120 ミリ秒です。デフォルト値は 5000 ミリ秒です。 • *prc-initial-wait*：トポロジ変更後の最初の PRC 計算遅延（ミリ秒）。指定できる範囲は 1 ～ 10,000 ミリ秒です。デフォルト値は 50 ミリ秒です。 • *prc-second-wait*：最初と 2 番目の PRC 計算間（ミリ秒）のホールドタイム。指定できる範囲は 1 ～ 10,000 ミリ秒です。デフォルト値は 200 ミリ秒です。
ステップ 15	**log-adjacency-changes** [**all**] 例： `Device(config-router)#log-adjacency-changes all`	（任意）IS-IS 隣接ステート変更をログするようルータを設定します。End System-to-Intermediate System PDU および LSP など、IS-IS hello に関連しないイベントにより生成されたすべての変更をログに含めるには、**all** を入力します。
ステップ 16	**lsp-mtu** *size* 例： `Device(config-router)#lsp mtu 1560`	（任意）最大 LSP パケットサイズ（バイト）を指定します。指定できる範囲は 128 ～ 4352 バイトです。デフォルト値は 1497 バイトです。 （注） ネットワーク内のリンクで MTU サイズが縮小された場合、ネットワーク内のすべてのデバイスで LSP MTU サイズを変更する必要があります。
ステップ 17	**partition avoidance** 例： `Device(config-router)#partition avoidance`	（任意）境界ルータ、すべての隣接レベル 1 ルータ、およびエンド ホスト間で、フル接続が切断された場合、IS-IS レベル 1-2 境界ルータがレベル 1 エリア プレフィックスをレベル 2 バックボーンにアドバタイズしないようにします。

	コマンドまたはアクション	目的
ステップ 18	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。

IS-IS インターフェイス パラメータの設定

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config)#interface gigabitethernet 1/0/1`	設定するインターフェイスを指定し、インターフェイス コンフィギュレーションモードを開始します。インターフェイスがまだレイヤ 3 インターフェイスとして設定されていない場合は、**no switchport** コマンドを入力してインターフェイスをレイヤ 3 モードに設定します。
ステップ 4	**isis metric** *default-metric* [**level-1** | **level-2**] 例： `Device(config-if)#isis metric 15`	（任意）指定したインターフェイスにメトリック（またはコスト）を設定します。指定できる範囲は 0 ～ 63 です。デフォルトは 10 です。レベルが入力されない場合は、レベル 1 ルータとレベル 2 ルータの両方にデフォルト値が適用されます。
ステップ 5	**isis hello-interval** {*seconds* | **minimal**} [**level-1** | **level-2**] 例：	（任意）デバイスが hello パケットを送信する間隔を指定します。デフォルトでは、hello インターバル *seconds* の 3 倍の値が、送信される hello パケットの

	コマンドまたはアクション	目的
	`Device(config-if)#isis hello-interval minimal`	*holdtime* としてアドバタイズされます。hello インターバルが狭まると、トポロジ変更の検出も速くなりますが、ルーティング トラフィック量は増大します。 • **minimal**：結果として得られるホールドタイムが 1 秒になるように、hello 乗数に基づいて hello 間隔が計算されます。 • *seconds*：指定できる範囲は 1 ー 65535 です。デフォルトは 10 秒です。
ステップ 6	**isis hello-multiplier** *multiplier* [**level-1** | **level-2**] 例： `Device(config-if)#isis hello-multiplier 5`	（任意）ネイバーが見落とすことができる IS-IS hello パケット数の最大値を指定します。見落とされたパケット数がこの値を超えると、デバイスは隣接がダウンしていると宣言します。指定できる範囲は 3 ～ 1000 です。デフォルトは 3 です。 （注） hello 乗数を小さくすると、高速コンバージェンスとなりますが、ルーティングが不安定になる場合があります。
ステップ 7	**isis csnp-interval** *seconds* [**level-1** | **level-2**] 例： `Device(config-if)#isis csnp-interval 15`	（任意）インターフェイスに IS-IS CSNP を設定します。指定できる範囲は 0 ～ 65535 です。デフォルトは 10 秒です。
ステップ 8	**isis retransmit-interval** *seconds* 例： `Device(config-if)#isis retransmit-interval 7`	（任意）ポイントツーポイントリンクの IS-IS LSP の再送信間隔（秒）を設定します。整数で、ネットワーク上の 2 つのルータ間で予測されるラウンドトリップ遅延よりも大きい値を指定してください。指定できる範囲は 0 ～ 65535 です。デフォルトは 5 秒です。
ステップ 9	**isis retransmit-throttle-interval** *milliseconds* 例：	（任意）IS-IS LSP 再送信スロットルインターバルを設定します。これは、IS-IS LSP がポイントツーポイントリンク上で再送信される最大レート（パ

	コマンドまたはアクション	目的
	`Device(config-if)#isis retransmit-throttle-interval 4000`	ケット間のミリ秒数）です。指定できる範囲は 0 ー 65535 です。デフォルトは **isis lsp-interval** コマンドによって決定されます。
ステップ 10	**isis priority** *value* [**level-1** | **level-2**] 例： `Device(config-if)#isis priority 50`	（任意）指定ルータの優先順位を設定します。指定できる範囲は 0 ～ 127 です。デフォルトは 64 です。
ステップ 11	**isis circuit-type** {**level-1** | **level-1-2** | **level-2-only**} 例： `Device(config-if)#isis circuit-type level-1-2`	（任意）指定されたインターフェイス上のネイバーで必要な隣接タイプを設定します（インターフェイスの回線タイプを指定します）。 • **level-1**：このノードとネイバーの両方に共通のエリアアドレスが少なくとも 1 つある場合、レベル 1 隣接関係が確立されます。 • **level-1-2**：ネイバーもレベル 1 およびレベル 2 の両方として設定されていて、少なくとも 1 つの共通のエリアがある場合、レベル 1 およびレベル 2 隣接関係が確立されます。共通のエリアがない場合は、レベル 2 隣接関係が確立されます。これはデフォルト設定です。これがデフォルトのオプションです。 • **level 2**：レベル 2 隣接関係が確立されます。ネイバールータがレベル 1 ルータである場合、隣接関係は確立されません。
ステップ 12	**isis password** *password* [**level-1** | **level-2**] 例： `Device(config-if)#isis password secret`	（任意）インターフェイスの認証パスワードを設定します。デフォルトでは、認証はディセーブルに設定されています。レベル 1 またはレベル 2 を指定すると、それぞれレベル 1 またはレベル 2 ルーティング用のパスワードだけがイネーブルになります。レベルを指定しない場合、デフォルトはレベル 1 およびレベル 2 です。

	コマンドまたはアクション	目的
ステップ 13	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。

IS-IS 認証の設定方法

ここでは、認証キーを生成する方法、インターフェイスの IS-IS 認証を設定する方法、およびインスタンスの IS-IS 認証を設定する方法について説明します。

認証キーの設定

複数のキーにライフタイムを設定できます。認証パケットを送信するために、最新の送信ライフタイム設定を持つキーが選択されます。複数のキーが同じ送信ライフタイム設定を持つ場合、キーはランダムに選択されます。受信した認証パケットを調べて受け入れるには、**accept-lifetime** コマンドを使用します。デバイスは、これらのライフタイムを認識している必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**key chain** *name-of-chain* 例： `Device(config)#key chain key10`	キーチェーンを識別し、キーチェーン コンフィギュレーション モードを開始します。
ステップ 4	**key** *number* 例： `Device(config-keychain)#key 2000`	キー番号を識別します。範囲は 0 ～ 65535 です。

	コマンドまたはアクション	目的
ステップ 5	**key-string** *text* 例： `Device(config-keychain-key)#Room 20, 10th floor`	キーストリングを確認します。ストリングには 1 ～ 80 文字の大文字および小文字の英数字を指定できますが、最初の文字に数字を指定できません。
ステップ 6	**accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} 例： `Device(config-keychain-key)#accept-lifetime 12:30:00 Jan 25 1009 infinite`	（任意）キーを受信できる期間を指定します。 *start-time* および *end-time* 構文には、*hh:mm:ss month date year* または *hh:mm:ss date month year* のいずれかを使用できます。デフォルトは、デフォルトの *start-time* 以降、無制限です。指定できる最初の日付は 1993 年 1 月 1 日です。デフォルトの *end-time* および **duration** は **infinite** です。
ステップ 7	**send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} 例： `Device(config-keychain-key)#accept-lifetime 23:30:00 Jan 25 1019 infinite`	（任意）キーを送信できる期間を指定します。 *start-time* および *end-time* 構文には、*hh:mm:ss month date year* または *hh:mm:ss date month year* のいずれかを使用できます。デフォルトの *start-time* は **infinite** で、指定できる最初の日付は 1993 年 1 月 1 日です。デフォルトの *end-time* および **duration** は **infinite** です。
ステップ 8	**cryptographic-algorithm** {**hmac-sha-1** | **hmac-sha-256** | **hmac-sha-384** | **hmac-sha-512** | **md5** } 例： `Device(config-keychain-key)#cryptographic-algorithm hmac-sha1-256`	（任意）暗号化アルゴリズムを指定します。
ステップ 9	**end** 例： `Device(config-keychain-key)#end`	特権 EXEC モードに戻ります。
ステップ 10	**show key chain** 例： `Device#show key chain`	認証キーの情報を表示します。

IS-IS インスタンスの HMAC-MD5 またはクリアテキスト認証の設定

ある認証方法から別の認証方法へ円滑に移行を実現し、IS-IS PDU の継続的な認証を可能にするには、ネットワークで通信する各デバイスでこの手順を実行します。

始める前に

認証文字列キーが生成されている必要があります。ネットワーク内のすべてのデバイスで同じ認証文字列キーを設定する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router isis** [**area** *tag*] 例： `Device(config)#router isis 1`	IP ルーティング プロトコルとして IS-IS を有効化し、必要に応じてプロセスにタグを割り当てます。ルータ コンフィギュレーション モードを開始します。
ステップ 4	**authentication send-only** [**level-1** | **level-2**] 例： `Device(config-router)#authentication send-only`	指定した IS-IS インスタンスについて送信された（受信ではなく）PDU に対してのみ認証が実行されるように指定します。
ステップ 5	**authentication mode** {**md5** | **text**}[**level-1** | **level-2**] 例： `Device(config-router)#authentication mode md5`	指定された IS-IS インスタンスについて PDU で使用される認証のタイプを指定します。 • **md5**：MD5 認証。 • **text**：クリアテキスト認証。
ステップ 6	**authentication key-chain** *name-of-chain* [**level-1** | **level-2**] 例： `Device(config-router)#authentication key-chain remote3754`	指定された IS-IS インスタンスについて認証が有効になります。

	コマンドまたはアクション	目的
ステップ 7	**no authentication send-only** 例： `Device(config-router)#no authentication send-only`	指定した IS-IS インスタンスについて送信および受信された PDU に対してのみ認証が実行されるように指定します。

IS-IS インターフェイスの HMAC-MD5 またはクリア テキスト認証の設定

ある認証方法から別の認証方法へ円滑に移行を実現し、IS-IS PDU の継続的な認証を可能にするには、ネットワークで通信する各デバイスでこの手順を実行します。

始める前に

認証文字列キーが生成されている必要があります。ネットワーク内のすべてのデバイスで同じ認証文字列キーを設定する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *type number* 例： `Device(config)#interface ethernet 0`	インターフェイスを設定します。
ステップ 4	**isis authentication send-only** [**level-1** | **level-2**] 例： `Device(config-if)#isis authentication send-only`	指定した IS-IS インターフェイスについて送信された（受信ではなく）PDU に対してのみ認証が実行されるように指定します。

	コマンドまたはアクション	目的
ステップ 5	**isis authentication mode** {**md5** | **text**}[**level-1** | **level-2**] 例： `Device(config-if)#isis authentication mode md5`	指定された IS-IS インスタンスについて PDU で使用される認証のタイプを指定します。 • **md5**：MD5 認証。 • **text**：クリアテキスト認証。
ステップ 6	**isis authentication key-chain** *name-of-chain* [**level-1** | **level-2**] 例： `Device(config-if)#isis authentication key-chain multistate87723`	指定された IS-IS インスタンスについて MD5 認証が有効になります。
ステップ 7	**no isis authentication send-only** 例： `Device(config-if)#no isis authentication send-only`	IS-IS インスタンスについて送信および受信された PDU に対してのみ認証が実行されるように指定します。

IS-IS のモニタリングおよびメンテナンス

ルーティング テーブル、キャッシュ、およびデータベースの内容など、特定の IS-IS の統計情報を表示できます。また、特定のインターフェイス、フィルタ、またはネイバーに関する情報も表示できます。

次の表に、IS-IS ルーティングを消去および表示するために使用する特権 EXEC コマンドを示します。

表 *42 : IS-IS show* コマンド

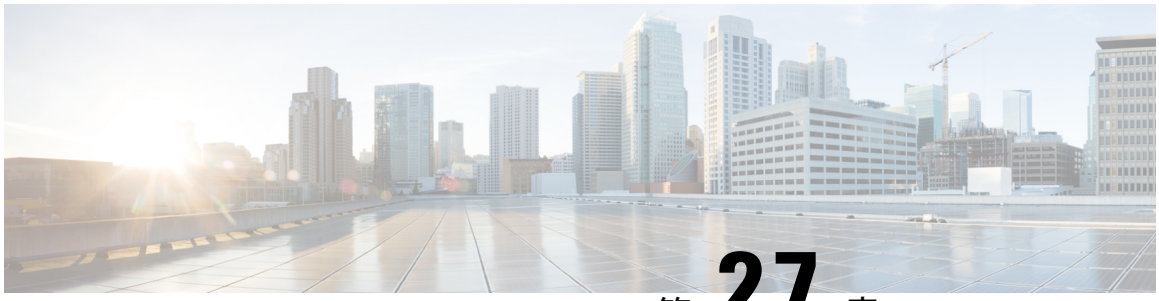
コマンド	目的
show ip route isis	IS-IS IP ルーティングテーブルの現在のステートを表示します。
show isis database	IS-IS リンクステート データベースを表示します。
show isis routes	IS-IS レベル 1 ルーティング テーブルを表示します。
show isis spf-log	IS-IS の SPF 計算の履歴を表示します。

コマンド	目的
show isis topology	すべてのエリア内の接続されたルータすべてのリストを表示します。
show route-map	設定済みのすべてのルートマップを表示するか、指定した 1 つのルートマップだけを表示します。
trace clns *[接続先（Destination）]*	ネットワークのパケットが指定された宛先までに経由するパスをトレースします。

IS-IS の機能情報

表 *43 : IS-IS* の機能情報

機能名	リリース	機能情報
Intermediate System-to-Intermediate System（IS-IS）	Cisco IOS XE Everest 16.6.1	この機能が導入されました。
	Cisco IOS XE Gibraltar 16.10.1	IS-IS は、セキュア ハッシュ アルゴリズム（SHA）認証（SHA-1、SHA-256、SHA-384、および SHA-512）をサポートするようになりました。



第 27 章

プロトコル独立機能

• プロトコル独立機能 （329 ページ）

プロトコル独立機能

分散型シスコ エクスプレス フォワーディング

シスコ エクスプレス フォワーディングに関する情報

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

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

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

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

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

シスコ エクスプレス フォワーディングの設定方法

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

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

⚠

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

ディセーブルである CEF または dCEF をグローバルにイネーブルにしたり、ソフトウェア転送トラフィックのインターフェイス上でイネーブルにするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**ip cef** 例： `Device(config)# ip cef`	非スタッキング スイッチで CEF の動作をイネーブルにします。 ステップ 4 に進みます。
ステップ 3	**ip cef distributed** 例： `Device(config)# ip cef distributed`	アクティブ スイッチで CEF の動作をイネーブルにします。

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

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

CEF トラフィック用のロードバランシングスキーム

CEFトラフィック用のロードバランシングスキームの設定に関する制約事項

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

CEF ロード バランシングの概要

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

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

CEF トラフィックに対する宛先別ロードバランシング

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

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

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

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

CEF トラフィックに対するロード バランシング アルゴリズム

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

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

CEF トラフィックに対するロードバランシングの設定方法

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

CEF の宛先別ロードバランシングの有効化または無効化

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

手順

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

	コマンドまたはアクション	目的
	`Device(config-if)# ip load-sharing per-destination`	**no ip load-sharing per-destination** コマンドを使用すると、インターフェイスで CEF の宛先別ロードバランシングが無効になります。
ステップ 5	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

CEF トラフィックに対するトンネル ロードバランシング アルゴリズムの選択

ネットワーク環境に少数の発信元と宛先のペアしか存在しない場合には、トンネル アルゴリズムを選択します。デバイスは、デフォルトではユニバーサル ロード シェアリングを実行するよう設定されています。

CEF トラフィック用にトンネル ロード バランシング アルゴリズムを選択するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device# enable`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip cef load-sharing algorithm** {**original** | **universal** [*id*] } 例： `Device(config)# ip cef load-sharing algorithm universal`	CEF のロードバランシング アルゴリズムを選択します。 • **original** キーワードは、発信元と宛先のハッシュに基づいて、ロード バランシング アルゴリズムとしてオリジナルアルゴリズムを設定します。 • **universal** キーワードは、ロード バランシング アルゴリズムとして、発信元と宛先および ID ハッシュを使用するものを設定します。

	コマンドまたはアクション	目的
		• *id* 引数は、固定 ID です。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

CEF トラフィックのロードバランシングの設定例

ここでは、CEF トラフィックのロードバランシングの設定例を示します。

例：CEF の宛先別ロードバランシングの有効化または無効化

CEF がイネーブルの場合、宛先別ロード バランシングはデフォルトでイネーブルです。次の例は、宛先単位のロード バランシングをディセーブルにする方法を示しています。

```
Device> enable
Device# configure terminal
Device(config)# interface Ethernet1/0/1
Device(config-if)# no ip load-sharing per-destination
Device(config-if)# end
```

等コスト ルーティング パスの個数

等コスト ルーティング パスに関する情報

同じネットワークへ向かう同じメトリックのルートが複数ルータに格納されている場合、これらのルートは等価コストを保有していると見なされます。ルーティング テーブルに複数の等コスト ルートが含まれる場合は、これらをパラレル パスと呼ぶこともあります。ネットワークへの等コスト パスがルータに複数格納されている場合、ルータはこれらを同時に使用できます。パラレル パスを使用すると、パスに障害が発生した場合に冗長性を確保できます。また、使用可能なパスにパケットの負荷を分散し、使用可能な帯域幅を有効利用することもできます。等コスト ルートは、スタック内の各スイッチでサポートされます。

等コスト ルートはルータによって自動的に取得、設定されますが、ルーティング テーブルの IP ルーティング プロトコルでサポートされるパラレル パスの最大数は制御可能です。スイッチ ソフトウェア では最大 32 の等コスト ルーティングが許可されていますが、スイッチ ハードウェアはルートあたり 17 パス以上は使用しません。

等コスト ルーティング パスの設定方法

手順

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

スタティック ユニキャスト ルート

スタティック ユニキャスト ルートに関する情報

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

ユーザによって削除されるまで、スタティック ルートはスイッチに保持されます。ただし、アドミニストレーティブ ディスタンスの値を割り当て、スタティック ルートをダイナミック ルーティング情報で上書きできます。各ダイナミック ルーティング プロトコルには、デフォルトのアドミニストレーティブ ディスタンスが設定されています（表 10 を参照）。ダイナミック ルーティング プロトコルの情報でスタティック ルートを上書きする場合は、スタティック ルートのアドミニストレーティブ ディスタンスがダイナミック プロトコルのアドミニストレーティブ ディスタンスよりも大きな値になるように設定します。

表 *44*: ダイナミック ルーティング プロトコルのデフォルトのアドミニストレーティブ ディスタンス

ルートの送信元	デフォルト距離
接続されているインターフェイス	0
スタティック ルート	1
EIGRP サマリー ルート	5
内部 EIGRP	90
IGRP	100
OSPF	110
不明	225

インターフェイスを指し示すスタティック ルートは、RIP、IGRP、およびその他のダイナミック ルーティング プロトコルを通してアドバタイズされます。**redistribute** スタティック ルータ コンフィギュレーション コマンドが、これらのルーティング プロトコルに対して指定されているかどうかは関係ありません。これらのスタティック ルートがアドバタイズされるのは、インターフェイスを指し示すスタティック ルートが接続された結果、静的な性質を失ったとルーティング テーブルで見なされるためです。ただし、network コマンドで定義されたネットワーク以外のインターフェイスに対してスタティック ルートを定義する場合は、ダイナミック ルーティング プロトコルに **redistribute** スタティック コマンドを指定しない限り、ルートはアドバタイズされません。

インターフェイスがダウンすると、ダウンしたインターフェイスを経由するすべてのスタティック ルートが IP ルーティング テーブルから削除されます。転送ルータのアドレスとして指定されたアドレスへ向かう有効なネクスト ホップがスタティック ルート内に見つからない場合は、IP ルーティング テーブルからそのスタティック ルートも削除されます。

スタティック ユニキャスト ルートの設定

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

スタティック ルートを設定するには、次の手順を実行します。

手順

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

次のタスク

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

デフォルトのルートおよびネットワーク

デフォルトのルートおよびネットワークに関する情報

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

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

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

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

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

デフォルトのルートおよびネットワークの設定方法

手順

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

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

ルーティング情報を再配信するためのルート マップ

ルート マップの概要

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

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

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

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

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

ルート マップの設定方法

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

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

手順

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

	コマンドまたはアクション	目的
		sequence number（任意）：同じ名前によってすでに設定されているルートマップのリスト内で、新しいルートマップの位置を指定する番号です。
ステップ 3	**match as-path** *path-list-number* 例： `Device(config-route-map)#match as-path 10`	BGP AS パス アクセス リストと照合します。
ステップ 4	**match community-list** *community-list-number* [**exact**] 例： `Device(config-route-map)# match community-list 150`	BGP コミュニティ リストのマッチングを行います。
ステップ 5	**match ip address** {*access-list-number* | *access-list-name*} [*...access-list-number* | *...access-list-name*] 例： `Device(config-route-map)# match ip address 5 80`	名前または番号を指定し、標準アクセス リストと照合します。1 ～ 199 の整数を指定できます。
ステップ 6	**match metric** metric-value 例： `Device(config-route-map)# match metric 2000`	指定されたルートメトリックと一致させます。*metric-value* には、0 ～ 4294967295 の値が指定された、EIGRP のメトリックを指定できます。
ステップ 7	**match ip next-hop** {*access-list-number* | *access-list-name*} [*...access-list-number* | *...access-list-name*] 例： `Device(config-route-map)# match ip next-hop 8 45`	指定されたアクセス リスト（番号 1 ～ 199）のいずれかで送信される、ネクストホップのルータアドレスと一致させます。
ステップ 8	**match tag** *tag value* [*...tag-value*] 例： `Device(config-route-map)# match tag 3500`	1 つまたは複数のルート タグ値からなるリスト内の指定されたタグ値と一致させます。0 ～ 4294967295 の整数を指定できます。

	コマンドまたはアクション	目的
ステップ 9	**match interface***type number* [*...type-number*] 例： `Device(config-route-map)# match interface gigabitethernet 1/0/1`	指定されたインターフェイスの 1 つから、指定されたネクスト ホップへのルートと一致させます。
ステップ 10	**match ip route-source** {*access-list-number* | *access-list-name*} [*...access-list-number* | *...access-list-name*] 例： `Device(config-route-map)# match ip route-source 10 30`	アドバタイズされた指定のアクセス リストによって指定したアドレスに一致します。
ステップ 11	**match route-type** {**local** | **internal** | **external** [**type-1** | **type-2**]} 例： `Device(config-route-map)# match route-type local`	指定された **route-type** と一致させます。 • **local**：ローカルに生成された BGP ルート。 • **internal**：OSPF エリア内およびエリア間ルート、または EIGRP 内部ルート。 • **external**：OSPF 外部ルート（タイプ 1 またはタイプ 2）または EIGRP 外部ルート。
ステップ 12	**set dampening** *halflife reuse suppress max-suppress-time* 例： `Device(config-route-map)# set dampening 30 1500 10000 120`	BGP ルート ダンプニング係数を設定します。
ステップ 13	**set local-preference** *value* 例： `Device(config-route-map)# set local-preference 100`	ローカル BGP パスに値を割り当てます。
ステップ 14	**set origin** {**igp** | **egp** *as* | **incomplete**} 例： `Device(config-route-map)#set origin igp`	BGP 送信元コードを設定します。

	コマンドまたはアクション	目的
ステップ 15	**set as-path** {**tag** | **prepend** *as-path-string*} 例： `Device(config-route-map)# set as-path tag`	BGP の自律システム パスを変更します。
ステップ 16	**set level** {**level-1** | **level-2** | **level-1-2** | **stub-area** | **backbone**} 例： `Device(config-route-map)# set level level-1-2`	ルーティング ドメインの指定エリアにアドバタイズされるルートのレベルを設定します。**stub-area** および **backbone** は、OSPF NSSA およびバックボーンエリアです。
ステップ 17	**set metric** *metric value* 例： `Device(config-route-map)# set metric 100`	再配布されるルートを指定するためのメトリック値を設定します（EIGRP のみ）。*metric value* は -294967295 ～ 294967295 の整数です。
ステップ 18	**set metric***bandwidth delay reliability loading mtu* 例： `Device(config-route-map)# set metric 10000 10 255 1 1500`	再配布されるルートを指定するためのメトリック値を設定します（EIGRP のみ）。 • *bandwidth*：0 ～ 4294967295 の範囲のルートのメトリック値または IGRP 帯域幅（キロビット/秒単位）。 • *delay*：0 ～ 4294967295 の範囲のルート遅延（10 マイクロ秒単位）。 • *reliability*：0 ～ 255 の数値で表されるパケット伝送の成功可能性。255 は信頼性が 100% であること、0 は信頼性がないことを意味します。 • *loading*：0 ～ 255 の数値で表されるルートの有効帯域幅（255 は 100% の負荷）。 • *mtu*：ルートの MTU の最小サイズ（バイト単位）。範囲は 0 ～ 4294967295 です。

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

ルート配信の制御方法

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

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

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

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

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

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

手順

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

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

ポリシーベース ルーティング

PBR の設定に関する制約事項

- ポリシーベースルーティング（PBR）は、トラフィックの GRE トンネルへの転送ではサポートされません。これは、任意のインターフェイスに適用される PBR と、トラフィックの GRE トンネルへの転送（PBR ネクストホップもしくはデフォルトのネクストホップまたは設定済みのインターフェイスを使用）に適用される PBR に適用されます。
- PBR は、GRE トンネル自体ではサポートされていません（GRE トンネル自体のもとで適用されます）。

ポリシーベース ルーティングの概要

PBR を使用すると、トラフィック フローに定義済みポリシーを設定できます。PBR を使用してルーティングをより細かく制御するには、ルーティング プロトコルから取得したルートの信頼度を小さくします。PBR は、次の基準に基づいて、パスを許可または拒否するルーティング ポリシーを設定したり、実装したりできます。

- 特定のエンド システムの ID
- アプリケーション
- プロトコル

PBR を使用すると、等価アクセスや送信元依存ルーティング、インタラクティブ対バッチ トラフィックに基づくルーティング、専用リンクに基づくルーティングを実現できます。たとえ

ば、在庫記録を本社に送信する場合は高帯域で高コストのリンクを短時間使用し、電子メールなど日常的に使用するアプリケーションデータは低帯域で低コストのリンクで送信できます。

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

- 許可とマークされているルート マップ文は次のように処理されます。
  - match コマンドは長さまたは複数の ACL で照合できます。ルート マップ文には複数の match コマンドを含めることができます。論理関数またはアルゴリズム関数は、許可または拒否の決定がされるまで、すべての match コマンドで実行されます。

    次に例を示します。

    match length A B

    match ip address acl1 acl2

    match ip address acl3

  パケットは、match length A B または acl1 または acl2 または acl3 により許可される場合に許可されます。
  - 決定が許可の場合は、set コマンドで指定されたアクションがパケットで適用されます。
  - 下された決定が拒否の場合は、PBR アクション（set コマンドで指定された）が適用されません。代わりに、処理ロジックが、シーケンス内の次のルートマップ文（シーケンス番号が次に高い文）に移動します。次の文が存在しない場合は、PBR 処理が終了し、パケットがデフォルトの IP ルーティング テーブルを使用してルーティングされます。
- PBR では、拒否としてマークされているルートマップ ステートメントはサポートされません。

標準 IP ACL を使用すると、アプリケーション、プロトコル タイプ、またはエンド ステーションに基づいて一致基準を指定するように、送信元アドレスまたは拡張 IP ACL の一致基準を指定できます。一致が見つかるまで、ルートマップにこのプロセスが行われます。一致が見つからない場合、通常の宛先ベースルーティングが行われます。match ステートメント リストの末尾には、暗黙の拒否ステートメントがあります。

match 句が満たされた場合は、set 句を使用して、パス内のネクスト ホップ ルータを識別する IP アドレスを指定できます。

PBR の設定方法

- マルチキャスト トラフィックには、ポリシーによるルーティングが行われません。PBR が適用されるのはユニキャスト トラフィックだけです。

- ルーテッド ポートまたは SVI 上で、PBR をイネーブルにできます。
- スイッチは一致長に基づき PBR をサポートします。
- レイヤ 3 モードの EtherChannel ポート チャネルにはポリシー ルート マップを適用できますが、EtherChannel のメンバーである物理インターフェイスには適用できません。適用しようとすると、コマンドが拒否されます。ポリシー ルート マップが適用されている物理インターフェイスは、EtherChannel のメンバーになることができません。
- スイッチまたはスイッチ スタックには最大 128 個の IP ポリシー ルート マップを定義できます。
- スイッチまたはスイッチ スタックには、PBR 用として最大 512 個のアクセス コントロール エントリ（ACE）を定義できます。
- ルート マップに一致基準を設定する場合は、次の注意事項に従ってください。
  - ローカル アドレス宛てのパケットを許可する ACL と照合させないでください。
- WCCP と PBR は、スイッチ インターフェイスで相互に排他的です。PBR がインターフェイスで有効になっているときは、WCCP を有効にできません。その反対の場合も同じで、WCCP がインターフェイスで有効になっているときは、PBR を有効にできません。
- PBR で使用されるハードウェア エントリ数は、ルート マップ自体、使用される ACL、ACL およびルート マップ エントリの順序によって異なります。
- TOS、DSCP、および IP Precedence に基づく PBR はサポートされません。
- set interface、set default next-hop、および set default interface はサポートされません。
- **ip next-hop recursive** および **ip next-hop verify availability** 機能は使用できません。next-hop は、直接接続される必要があります。
- set アクションのないポリシー マップはサポートされます。一致パケットは通常どおりにルーティングされます。
- match 句のないポリシー マップはサポートされます。set アクションはすべてのパケットに適用されます。

デフォルトでは、PBR はスイッチ上で無効です。PBR を有効にするには、一致基準および結果アクションを指定するルートマップを作成する必要があります。次に、特定のインターフェイスでそのルート マップ用の PBR を有効にします。指定したインターフェイスに着信したパケットのうち、match 句と一致したものはすべて PBR の対象になります。

スイッチ（CPU）で生成されたパケットまたはローカルパケットは、通常どおりにポリシー ルーティングされません。スイッチ上でローカル PBR をグローバルに有効にすると、そのスイッチから送信されたすべてのユニキャストパケットがローカル PBR の影響を受けます。ローカル PBR に関してサポートされているプロトコルは、NTP、DNS、MSDP、SYSLOG、および TFTP です。ローカル PBR は、デフォルトで無効に設定されています。

手順

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

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

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

ルーティング情報のフィルタリング

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

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

受動インターフェイスの設定

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

多数のインターフェイスが存在するネットワークで、インターフェイスを手動でパッシブに設定する作業を回避するには、**passive-interface default** ルータ コンフィギュレーション コマンド

を使用し、すべてのインターフェイスをデフォルトでパッシブになるように設定します。このあとで、隣接関係が必要なインターフェイスを手動で設定します。

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

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**router** {**rip** | **ospf** | **eigrp**} 例： `Device(config)# router ospf`	ルータ コンフィギュレーション モードを開始します。
ステップ 3	**passive-interface** *interface-id* 例： `Device(config-router)# passive-interface gigabitethernet 1/0/1`	指定されたレイヤ3インターフェイス経由のルーティング アップデートの送信を抑制します。
ステップ 4	**passive-interface default** 例： `Device(config-router)# passive-interface default`	（任意）すべてのインターフェイスを、デフォルトでパッシブとなるように設定します。
ステップ 5	**no passive-interface** *interface type* 例： `Device(config-router)# no passive-interface gigabitethernet1/0/3 gigabitethernet 1/0/5`	（任意）隣接関係を送信する必要があるインターフェイスだけをアクティブにします。
ステップ 6	**network** *network-address* 例： `Device(config-router)# network 10.1.1.1`	（任意）ルーティングプロセス用のネットワーク リストを指定します。 *network-address* は IP アドレスです。
ステップ 7	**end** 例：	特権 EXEC モードに戻ります。

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

ルーティング アップデートのアドバタイズおよび処理の制御

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

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router** {**rip** | **eigrp**} 例： `Device(config)# router eigrp 10`	ルータ コンフィギュレーション モードを開始します。
ステップ 4	**distribute-list** {*access-list-number* | *access-list-name*} **out** [*interface-name* | *routing process* | *autonomous-system-number*] 例： `Device(config-router)# distribute 120 out gigabitethernet 1/0/7`	アクセス リスト内のアクションに応じて、ルーティングアップデート内のルートのアドバタイズを許可または拒否します。

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

ルーティング情報の送信元のフィルタリング

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

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

手順

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

	コマンドまたはアクション	目的
ステップ 3	**router** {**rip** | **ospf** | **eigrp**} 例： `Device(config)# router eigrp 10`	ルータ コンフィギュレーション モードを開始します。
ステップ 4	**distance** *weight* {*ip-address* {*ip-address mask*}} [*ip access list*] 例： `Device(config-router)# distance 50 10.1.5.1`	アドミニストレーティブ ディスタンスを定義します。 *weight*：アドミニストレーティブディスタンスは 10 ～ 255 の整数です。単独で使用した場合、*weight* はデフォルトのアドミニストレーティブ ディスタンスを指定します。ルーティング情報の送信元に他の指定がない場合に使用されます。アドミニストレーティブ ディスタンスが 255 のルートはルーティング テーブルに格納されません。 （任意）*ip access list*：着信ルーティング アップデートに適用される IP 標準または IP 拡張アクセス リストです。
ステップ 5	**end** 例： `Device(config-router)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show ip protocols** 例： `Device# show ip protocols`	指定されたルーティング プロセス用のデフォルトのアドミニストレーティブ ディスタンスを表示します。
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

認証キーの管理

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

前提条件

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

認証キーの設定方法

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

手順

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

	コマンドまたはアクション	目的
		す。デフォルトは、デフォルトの *start-time* 以降、無制限です。指定できる最初の日付は 1993 年 1 月 1 日です。デフォルトの *end-time* および **duration** は **infinite** です。
ステップ 6	**send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} 例： `Device(config-keychain)# accept-lifetime 23:30:00 Jan 25 1019 infinite`	（任意）キーを送信できる期間を指定します。 *start-time* および *end-time* 構文には、*hh:mm:ss Month date year* または *hh:mm:ss date Month year* のいずれかを使用できます。デフォルトは、デフォルトの *start-time* 以降、無制限です。指定できる最初の日付は 1993 年 1 月 1 日です。デフォルトの *end-time* および **duration** は **infinite** です。
ステップ 7	**end** 例： `Device(config-keychain)# end`	特権 EXEC モードに戻ります。
ステップ 8	**show key chain** 例： `Device# show key chain`	認証キーの情報を表示します。
ステップ 9	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

第 28 章

VRF-Lite の設定

- VRF-Lite について （359 ページ）
- VRF-Lite の設定に関するガイドライン （361 ページ）
- VRF-Lite の設定方法 （362 ページ）
- VRF-Lite に関する追加情報 （383 ページ）
- VRF-Lite 設定の確認 （383 ページ）
- VRF-Lite の設定例 （385 ページ）
- VRF-Lite に関するその他の参考資料 （388 ページ）
- マルチキャスト VRF-Lite の機能履歴と情報 （389 ページ）

VRF-Lite について

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

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

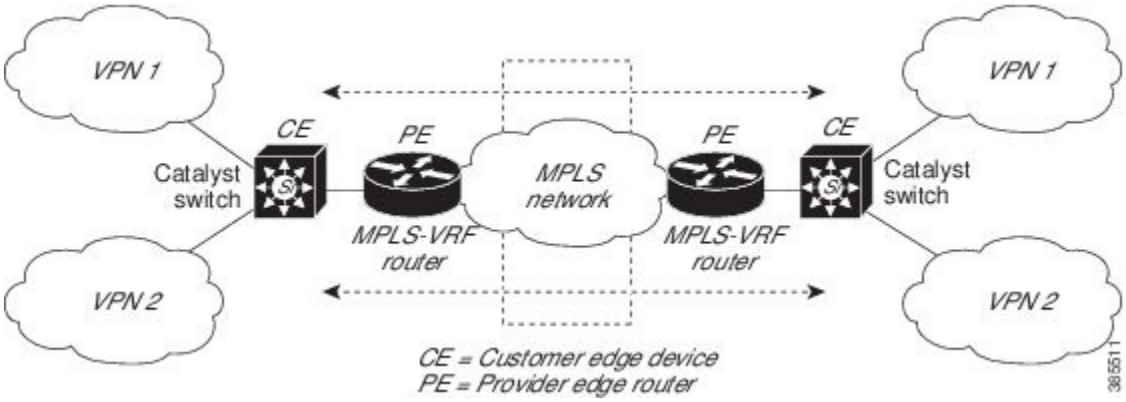
VRF-Lite には次のデバイスが含まれます。

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

VRF-lite を使用すると、複数の顧客が 1 つの CE を共有できます。また、1 つの物理リンクのみが CE と PE 間に使用されます。共有 CE は、お客様ごとに別々の VRF テーブルを維持し、独自のルーティング テーブルに基づいて、お客様ごとにパケットをスイッチングまたはルーティングします。VRF-lite は限定された PE の機能を CE デバイスに拡張して、個別の VRF テーブルを保守する機能を付与し、VPN のプライバシーおよびセキュリティをブランチ オフィスまで拡張します。

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

図 *14* : 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。

VRF-Lite の設定に関するガイドライン

IPv4 と IPv6

- VRF-Lite が設定されたスイッチは複数のカスタマーで共有され、すべてのカスタマーが独自のルーティング テーブルを持ちます。
- カスタマーは別々の VRF テーブルを使用するので、同じ IP アドレスを再利用できます。別々の VPN では IP アドレスの重複が許可されます。
- VRF-Lite では、複数のカスタマーが PE と CE の間で同一の物理リンクを共有できます。複数の VLAN を持つトランク ポートでは、パケットがお客様間で分離されます。すべてのカスタマーが独自の VLAN を持ちます。
- PE ルータでは、VRF-Lite の使用と複数の CE の使用には違いがありません。VRF-Lite について （359 ページ） では、複数の仮想レイヤ 3 インターフェイスが VRF-Lite デバイスに接続されています。
- Cisco Catalyst スイッチでは、物理ポートか VLAN SVI、またはその両方の組み合わせを使用して、VRF を設定できます。アクセス ポートまたはトランク ポート経由で SVI を接続できます。
- お客様は、別のお客様と重複しないかぎり、複数の VLAN を使用できます。お客様の VLAN は、スイッチに保存されている適切なルーティング テーブルの識別に使用される特定のルーティング テーブル ID にマッピングされます。
- レイヤ 3 TCAM リソースは、すべての VRF 間で共有されます。各 VRF が十分な CAM 領域を持つようにするには、**maximum routes** コマンドを使用します。
- VRF を使用した Cisco Catalyst スイッチは、1 つのグローバル ネットワークと複数の VRF をサポートできます。サポートされるルートの総数は、TCAM のサイズに制限されます。
- 1 つの VRF を IPv4 と IPv6 の両方に設定できます。
- 着信パケットの宛先アドレスが VRF テーブルにない場合、そのパケットはドロップされます。また、VRF ルートに TCAM 領域が十分にない場合、その VRF のハードウェア切り替えは無効になり、対応するデータ パケットがソフトウェアに送信されて処理されます。

IPv4 固有

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

IPv6 固有

- VRF 認識 OSPFv3、BGPv6、EIGRPv6、および IPv6 スタティックルーティングがサポートされます。
- VRF 認識 IPv6 ルート アプリケーションには、ping、telnet、ssh、tftp、ftp、およびトレースルートが含まれています（このリストには管理インターフェイスは含まれていません。これは、その下に IPv4 も IPv6 も設定できますが、別々に処理されます）。

VRF-Lite の設定方法

ここでは、VRF-Lite の設定について説明します。

IPv4 用の VRF-Lite の設定

ここでは、IPv4 用の VRF-Lite の設定について説明します。

VRF 認識サービスの設定

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

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

VRF 認識サービスには、次の特性があります。

- ユーザは、ユーザ指定の VRF 内のホストに ping を実行できます。
- ARP エントリは、個別の VRF で学習されます。ユーザは、特定の VRF の ARP エントリを表示できます。

ARP のユーザ インターフェイスの設定

手順

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

TACACS+ サーバ用の Per-VRF の設定

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

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

始める前に

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

手順

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

	コマンドまたはアクション	目的
ステップ 3	**ip vrf** *vrf-name* 例： `Device(config)# ip vrf vrf-name`	VRF テーブルを設定し、VRF コンフィギュレーション モードを開始します。
ステップ 4	**rd** *route-distinguisher* 例： `Device(config-vrf)# rd route-distinguisher`	VRF インスタンスに対するルーティングおよびフォワーディング テーブルを作成します。
ステップ 5	**exit** 例： `Device(config-vrf)# exit`	VRF コンフィギュレーション モードを終了します。
ステップ 6	**interface** *interface-name* 例： `Device(config)# interface interface-name`	インターフェイスを設定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 7	**vrf forwarding** *vrf-name* 例： `Device(config-if)# vrf forwarding vrf-name`	インターフェイスに VRF を設定します。
ステップ 8	**ip address** *ip-address mask* [**secondary**] 例： `Device(config-if)# ip address ip-address mask [secondary]`	インターフェイスに対するプライマリ IP アドレスまたはセカンダリ IP アドレスを設定します。
ステップ 9	**exit** 例： `Device(config-vrf)# exit`	インターフェイス コンフィギュレーション モードを終了します。
ステップ 10	**aaa group server tacacs+** *group-name* 例： `Device(config)# aaa group server tacacs+ tacacs1`	異なる TACACS+ サーバ ホストを別々のリストと方式にグループ化し、server-group コンフィギュレーション モードを開始します。
ステップ 11	**server-private** {*ip-address* | *name*} [**nat**] [**single-connection**] [**port** *port-number*] [**timeout** *seconds*] [**key** [**0** | **7**] *string*] 例： `Device(config-sg-tacacs+)# server-private 10.1.1.1 port 19 key cisco`	グループ サーバに対するプライベート TACACS+ サーバの IP アドレスを設定します。

	コマンドまたはアクション	目的
ステップ 12	**vrf forwarding** *vrf-name* 例： `Device(config-sg-tacacs+)# vrf forwarding vrf-name`	AAA TACACS+サーバグループの VRF リファレンスを設定します。
ステップ 13	**ip tacacs source-interface** *subinterface-name* 例： `Device(config-sg-tacacs+)# ip tacacs source-interface subinterface-name`	すべての発信 TACACS+ パケットに対して、指定されたインターフェイスの IP アドレスを使用します。
ステップ 14	**exit** 例： `Device(config-sg-tacacs)# exit`	server-group コンフィギュレーション モードを終了します。

例

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

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

マルチキャスト VRF の設定

手順

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

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

	コマンドまたはアクション	目的
ステップ 10	**ip address** *ip-address***mask** 例： `Device(config-if)# ip address ip-address mask`	レイヤ 3 インターフェイスの IP アドレスを設定します。
ステップ 11	**ip pim sparse-mode** 例： `Device(config-if)# ip pim sparse-mode`	VRF に関連付けられているレイヤ 3 インターフェイス上で、PIM をイネーブルにします。
ステップ 12	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。
ステップ 13	**show ip vrf** [**brief** | **detail** | **interfaces**] [*vrf-name*] 例： `Device# show ip vrf brief`	設定を確認します。設定した VRF に関する情報を表示します。
ステップ 14	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

例

次に、VRF テーブル内にマルチキャストを設定する例を示します。

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

VPN ルーティング セッションの設定

手順

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

	コマンドまたはアクション	目的
ステップ 2	**router ospf** *process-id* **vrf vrf-name** 例： `Device(config)# router ospf process-id vrf vrf-name`	OSPF ルーティングをイネーブルにし、VPN 転送テーブルを指定して、ルータ コンフィギュレーション モードを開始します。
ステップ 3	**log-adjacency-changes** 例： `Device(config-router)# log-adjacency-changes`	（任意）隣接状態（デフォルト）の変更を記録します。
ステップ 4	**redistribute bgp** *autonomous-system-number subnets* 例： `Device(config-router)# redistribute bgp autonomous-system-number subnets`	BGP ネットワークから OSPF ネットワークに情報を再配布するようにスイッチを設定します。
ステップ 5	**network** *network-number* **area** *area-id* 例： `Device(config-router)# network network-number area area-id`	OSPF が動作するネットワーク アドレスとマスク、およびそのネットワーク アドレスのエリア ID を定義します。
ステップ 6	**end** 例： `Device(config-router)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show ip ospf** *process-id* 例： `Device# show ip ospf process-id`	OSPF ネットワークの設定を確認します。
ステップ 8	copy running-config startup-config 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。 VPN 転送テーブルと OSPF ルーティング プロセスの関連付けを解除するには、**no router ospf** *process-id* **vrf** *vrf-name* グローバル コンフィギュレーション コマンドを使用します。

例

```
Device(config)# ip vrf VRF-RED
Device(config-vrf)# rd 1:1
Device(config-vrf)# exit
Device(config)# router eigrp virtual-name
Device(config-router)# address-family ipv4 vrf VRF-RED autonomous-system 1
Device(config-router-af)# network 10.0.0.0 0.0.0.255
Device(config-router-af)# topology base
```

```
Device(config-router-topology)# default-metric 10000 100 255 1 1500
Device(config-router-topology)# exit-af-topology
Device(config-router-af)# exit-address-family
```

BGP PE/CE ルーティング セッションの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**router bgp** *autonomous-system-number* 例： `Device(config)# router bgp autonomous-system-number`	その他の BGP ルータに渡された AS 番号で BGP ルーティング プロセスを設定し、ルータ コンフィギュレーション モードを開始します。
ステップ 3	**network** *network-number* **mask** *network-mask* 例： `Device(config-router)# network network-number mask network-mask`	BGP を使用してアナウンスするネットワークおよびマスクを指定します。
ステップ 4	**redistribute ospf** *process-id match internal* 例： `Device(config-router)# redistribute ospf process-id match internal`	OSPF 内部ルートを再配布するようにスイッチを設定します。
ステップ 5	**network** *network-number* **area** *area-id* 例： `Device(config-router)# network network-number area area-id`	OSPF が動作するネットワーク アドレスとマスク、およびそのネットワーク アドレスのエリア ID を定義します。
ステップ 6	**address-family ipv4 vrf** *vrf-name* 例： `Device(config-router-af)# address-family ipv4 vrf vrf-name`	PE から CE のルーティング セッションの BGP パラメータを定義し、VRF アドレス ファミリ モードを開始します。
ステップ 7	**neighbor** *address* **remote-as** *as-number* 例： `Device(config-router-af)# neighbor address remote-as as-number`	PE と CE ルータの間の BGP セッションを定義します。
ステップ 8	**neighbor** *address* **activate** 例：	IPv4 アドレス ファミリのアドバタイズメントをアクティブ化します。

	コマンドまたはアクション	目的
	`Device(config-router-af)# neighbor address activate`	
ステップ 9	**end** 例： `Device(config-router-af)# end`	特権 EXEC モードに戻ります。
ステップ 10	**show ip bgp** [**ipv4**] [**neighbors**] 例： `Device# show ip bgp [ipv4] [neighbors]`	BGP 設定を確認します。 BGP ルーティングプロセスを削除するには、**no router bgp** *autonomous-system-number* グローバル コンフィギュレーション コマンドを使用します。ルーティング特性を削除するには、コマンドにキーワードを指定してこのコマンドを使用します。

IPv4 VRF の設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**ip routing** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip vrf** *vrf-name* 例： `Device(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： `Device(config-vrf)# rd route-distinguisher`	ルート識別子を指定して VRF テーブルを作成します。自律システム番号と任意の数値（xxx:y）、または IP アドレスと任意の数値（A.B.C.D:y）のいずれかを入力します。
ステップ 5	**route-target** {**export** | **import** | **both**} *route-target-ext-community* 例：	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

	コマンドまたはアクション	目的
	`Device(config-vrf)# route-target {export | import | both} route-target-ext-community`	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： `Device(config-vrf)# import map route-map`	（任意）VRFにルートマップを対応付けます。
ステップ 7	**interface** *interface-id* 例： `Device(config-vrf)# interface interface-id`	インターフェイス コンフィギュレーション モードを開始して、VRFに対応付けるレイヤ3インターフェイスを指定します。インターフェイスにはルーテッドポートまたはSVIを設定できます。
ステップ 8	**vrf forwarding** *vrf-name* 例： `Device(config-if)# vrf forwarding vrf-name`	VRF をレイヤ3インターフェイスに対応付けます。
ステップ 9	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。
ステップ 10	**show ip vrf** [**brief** | **detail** | **interfaces**] [*vrf-name*] 例： `Device# 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： `Device# copy running-config startup-config`	（任意）コンフィギュレーションファイルに設定を保存します。 VRF とそのすべてのインターフェイスを削除するには、**no ip 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。VRFからインターフェイスを削除するには、**no vrf forwarding** インターフェイス コンフィギュレーション コマンドを使用します。

IPv6 用の VRF-Lite の設定

ここでは、IPv6 用の VRF-Lite の設定について説明します。

VRF 認識サービスの設定

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

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

VRF 認識サービスには、次の特性があります。

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

次のサービスは VRF 認識です。

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

PING のユーザ インターフェイスの設定

VRF 認識 ping を設定するには、次の作業を実行します。

手順

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

uRPF のユーザ インターフェイスの設定

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

手順

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

Traceroute のユーザ インターフェイスの設定

手順

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

Telnet および SSH のユーザ インターフェイスの設定

手順

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

NTP のユーザ インターフェイスの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**ntp server vrf** *vrf-name* **ipv6-host** 例： `Device(config)# ntp server vrf vrf-name ipv6-host`	指定された VRF で NTP サーバを設定します。
ステップ 3	**ntp peer vrf** *vrf-name* **ipv6-host** 例： `Device(config)# ntp peer vrf vrf-name ipv6-host`	指定された VRF で NTP ピアを設定します。

IPv6 VRF の設定

手順

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

	コマンドまたはアクション	目的
ステップ 2	**vrf definition** *vrf-name* 例： `Device(config)# vrf definition vrf-name`	VRF 名を指定し、VRF コンフィギュレーション モードを開始します。
ステップ 3	**rd** *route-distinguisher* 例： `Device(config-vrf)# rd route-distinguisher`	（任意）ルート識別子を指定して VRF テーブルを作成します。自律システム番号および任意の数（xxx:y）、または IP アドレスおよび任意の数（A.B.C.D:y）のいずれかを入力します。
ステップ 4	**address-family** *ipv4* | *ipv6* 例： `Device(config-vrf)# address-family ipv4 | ipv6`	（任意）デフォルトは IPv4 です。IPv6 の必須設定。
ステップ 5	**route-target** {**export** | **import** | **both**} *route-target-ext-community* 例： `Device(config-vrf)# route-target {export | import | both} route-target-ext-community`	指定された VRF のインポート、エクスポート、またはインポートおよびエクスポート ルート ターゲット コミュニティのリストを作成します。AS システム番号と任意の番号（xxx:y）または IP アドレスと任意の番号（A.B.C.D:y）を入力します。 （注） このコマンドは、BGP が動作している場合にのみ有効です。
ステップ 6	**exit-address-family** 例： `Device(config-vrf)# exit-address-family`	VRF アドレス ファミリ コンフィギュレーション モードを終了し、VRF コンフィギュレーション モードに戻ります。
ステップ 7	**vrf definition** *vrf-name* 例： `Device(config)# vrf definition vrf-name`	VRF コンフィギュレーション モードを開始します。
ステップ 8	**ipv6 multicast multitopology** 例： `Device(config-vrf-af)# ipv6 multicast multitopology`	マルチキャスト固有の RPF トポロジを有効にします。

	コマンドまたはアクション	目的
ステップ 9	**address-family ipv6 multicast** 例： `Device(config-vrf)# address-family ipv6 multicast`	マルチキャスト IPv6 アドレス ファミリを入力します。
ステップ 10	**end** 例： `Device(config-vrf-af)# end`	特権 EXEC モードに戻ります。

例

次に、VRF を設定する例を示します。

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

定義済み VRF へのインターフェイスの関連付け

手順

	コマンドまたはアクション	目的
ステップ 1	**interface** *interface-id* 例： `Device(config-vrf)# interface interface-id`	インターフェイス コンフィギュレーション モードを開始して、VRF に対応付けるレイヤ3インターフェイスを指定します。インターフェイスにはルーテッド ポートまたは SVI を設定できます。
ステップ 2	**no switchport** 例： `Device(config-if)# no switchport`	コンフィギュレーション モードからインターフェイスを削除します（物理インターフェイスの場合）。
ステップ 3	**vrf forwarding** *vrf-name* 例： `Device(config-if)# vrf forwarding vrf-name`	VRF をレイヤ 3 インターフェイスに対応付けます。

	コマンドまたはアクション	目的
ステップ 4	**ipv6 enable** 例： `Device(config-if)# ipv6 enable`	インターフェイスで IPv6 をイネーブルにします。
ステップ 5	**ipv6 address** *ip-address subnet-mask* 例： `Device(config-if)# ipv6 address ip-address subnet-mask`	インターフェイスの IPv6 アドレスを入力します。
ステップ 6	**show ipv6 vrf** [**brief** | **detail** | **interfaces**] [*vrf-name*] 例： `Device# show ipv6 vrf [brief | detail | interfaces] [vrf-name]`	設定を確認します。設定した VRF に関する情報を表示します。
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

例

次に、インターフェイスを VRF に関連付ける例を示します。

```
Switch(config-vrf)# interface ethernet0/1
Switch(config-if)# vrf forwarding red
Switch(config-if)# ipv6 enable
Switch(config-if)# ipv6 address 5000::72B/64
```

ルーティング プロトコル経由での VRF へのルートの入力

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

VRF スタティック ルートの設定

手順

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

	コマンドまたはアクション	目的
ステップ 2	**ipv6 route** [**vrf** *vrf-name*] *ipv6-prefix*/*prefix-length* {*ipv6-address* | **interface-type** *interface-number* [*ipv6-address*]} 例： `Device(config)# ipv6 route [vrf vrf-name] ipv6-prefix/prefix-length {ipv6-address | interface-type interface-number [ipv6-address]}`	VRF に固有のスタティック ルートを設定します。

例

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

OSPFv3 ルータ プロセスの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**router ospfv3** *process-id* 例： `Device(config)# router ospfv3 process-id`	IPv6 アドレス ファミリの OSPFv3 ルータ コンフィギュレーション モードを有効にします。
ステップ 3	**area** *area-ID* [**default-cot** | **nssa** | **stub**] 例： `Device(config-router)# area area-ID [default-cot | nssa | stub]`	OSPFv3 エリアを設定します。
ステップ 4	**router-id** *router-id* 例： `Device(config-router)# router-id router-id`	固定ルータ ID を使用します。
ステップ 5	**address-family ipv6 unicast vrf** *vrf-name* 例： `Device(config-router)# address-family ipv6 unicast vrf vrf-name`	vrf vrf-name の OSPFv3 の IPv6 アドレス ファミリ コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 6	**redistribute source-protocol** [*process-id*] **options** 例： `Device(config-router)# redistribute source-protocol [process-id] options`	あるルーティング ドメインから別のルーティング ドメインへ IPv6 ルートを再配布します。
ステップ 7	**end** 例： `Device(config-router)# end`	特権 EXEC モードに戻ります。

例

次に、OSPFv3 ルータ プロセスを設定する例を示します。

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

インターフェイス上での OSPFv3 のイネーブル化

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *type-number* 例： `Device(config-vrf)# interface type-number`	インターフェイスのタイプと番号を指定し、スイッチをインターフェイス コンフィギュレーション モードにします。
ステップ 3	**ospfv3** *process-id* **area** *area-id* **ipv6** [**instance** *instance-id*] 例： `Device(config-if)# ospfv3 process-id area area-ID ipv6 [instance instance-id]`	IPv6 AF を設定したインターフェイスで OSPFv3 を有効にします。
ステップ 4	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。

例

次に、インターフェイス上で OSPFv3 を有効にする例を示します。

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

EIGRPv6 ルーティング プロセスの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**router eigrp** *virtual-instance-name* 例： `Device(config)# router eigrp virtual-instance-name`	EIGRP ルーティング プロセスを設定し、ルータ コンフィギュレーション モードを開始します。
ステップ 3	**address-family ipv6 vrf** *vrf-name* **autonomous-system** *autonomous-system-number* 例： `Device(config-router)# address-family ipv6 vrf vrf-name autonomous-system autonomous-system-number`	EIGRP IPv6 VRF-Lite を有効にし、アドレス ファミリ コンフィギュレーション モードを開始します。
ステップ 4	**topology** {**base** | **topology-name tid number** 例： `Device(config-router-af)# topology {base | topology-name tid number`	指定されたトポロジ インスタンスで IP トラフィックをルーティングするよう EIGRP プロセスを設定し、アドレス ファミリ トポロジ コンフィギュレーション モードを開始します。
ステップ 5	**exit-aftopology** 例： `Device(config-router-af-topology)# exit-aftopology`	アドレス ファミリ トポロジ コンフィギュレーション モードを終了します。
ステップ 6	**eigrp router-id** *ip-address* 例： `Device(config-router)# eigrp router-id ip-address`	固定ルータ ID の使用を有効にします。

	コマンドまたはアクション	目的
ステップ 7	end 例： `Device(config-router)# end`	ルータ コンフィギュレーション モードを終了します。

例

次に、EIGRP ルーティング プロセスを設定する例を示します。

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

EBGPv6 ルーティング プロセスの設定

手順

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

	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： `Device(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： `Device(config-router-af)# neighbor ipv6-address peer-group peer-group-name`	BGP ネイバーの IPv6 アドレスをピア グループに割り当てます。
ステップ 7	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*[*%*]} **route-map** *map-name* {**in** | **out**} 例： `Device(config-router-af)# neighbor {ip-address | peer-group-name | ipv6-address[%]}route-map map-name {in | out}`	着信ルートまたは発信ルートにルート マップを適用します。ルート マップへの変更は、ピアリングがリセットされるまで、またはソフト リセットが実行されるまで、現在のピアでは有効になりません。soft キーワードと in キーワードを指定して clear bgp ipv6 コマンドを使用すると、ソフト リセットが実行されます。
ステップ 8	exit 例： `Device(config-router-af)# exit`	アドレス ファミリ コンフィギュレーション モードを終了し、ルータをルータ コンフィギュレーション モードに戻します。

例

次に、EBGPv6 を設定する例を示します。

```
Device(config)# router bgp 2
Device(config-router)# bgp router-id 2.2.2.2
Device(config-router)# bgp log-neighbor-changes
```

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

VRF-Lite に関する追加情報

ここでは、VRF-Lite に関する追加情報を提供します。

IPv4 と IPv6 間での VPN の共存

IPv4 を設定するための「以前の」CLI と、IPv6 用の「新しい」CLI 間には下位互換性があります。つまり、設定に両方の CLI を含めることができます。IPv4 CLI は、同じインターフェイス上で、VRF 内で定義されている IP アドレスとともにグローバル ルーティング テーブルで定義されている IPv6 アドレスも備える機能を保持しています。

次に例を示します。

```
vrf definition red
 rd 100:1
 address family ipv6
 route-target both 200:1
 exit-address-family
!
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

ここでは、VRF-Lite 設定を確認する手順について説明します。

IPv4 VRF-Lite ステータスの表示

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

コマンド	目的
Device# **show ip protocols vrf** *vrf-name*	VRF に対応付けられたルーティング プロトコル情報を表示します。
Device# **show ip route vrf** *vrf-name* [**connected**] [*protocol* [*as-number*]] [**list**] [**mobile**] [**odr**] [**profile**] [**static**] [**summary**][**supernets-only**]	VRF に対応付けられた IP ルーティング テーブル情報を表示します。
Device# **show 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。
Device# **bidir vrf** *instance-name a.b.c.d* | **active** | **bidriectional**| **count** | **interface** | **proxy** | **pruned** | **sparse** | **ssm** | **static** | **summary**	定義された VRF インスタンスに関する情報を表示します。

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

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

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

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

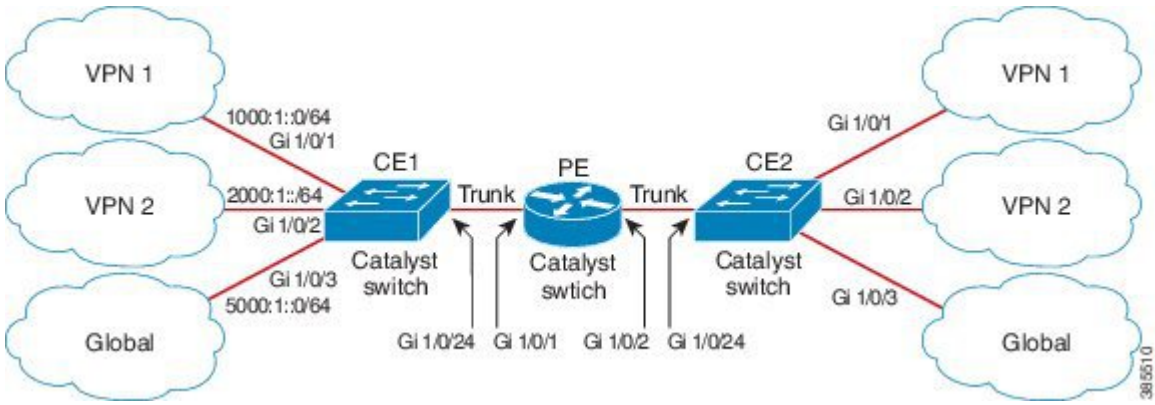
VRF-Lite の設定例

ここでは、VRF-Lite の設定例を示します。

IPv6 VRF-Lite の設定例

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

図 *15 : VRF-Lite* の設定例



CE1 スイッチの設定

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

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

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

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

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

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

interface GigabitEthernet 1/0/24
 switchport trunk encapsulation dot1q

switchport mode trunk
end

router ospfv3 100
 router-id 10.10.10.10
 !
 address-family ipv6 unicast vrf v1
  redistribute connected
  area 0 normal
 exit-address-family
!

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

PE スイッチの設定

```
ipv6 unicast-routing

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

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

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

interface Vlan800
 vrf forwarding v1
 ipv6 address 3000:1::7/64
 ospfv3 100 ipv6 area 0
```

```
!
interface Vlan900
 vrf forwarding v2
 ipv6 address 4000:1::7/64
 ospfv3 200 ipv6 area 0
!

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

interface GigabitEthernet 1/0/2
 switchport trunk encapsulation dot1q

switchport mode trunk
 exit

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

CE2 スイッチの設定

```
ipv6 unicast-routing

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

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

interface Vlan100
 vrf forwarding v1

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

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

interface GigabitEthernet 1/0/1
```

```
switchport access vlan 100
end

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

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

router ospfv3 100
 router-id 40.40.40.40
 !
 address-family ipv6 unicast vrf v1
  redistribute connected
  area 0 normal
 exit-address-family
!

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

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

VRF-Lite に関するその他の参考資料

関連資料

関連項目	マニュアル タイトル
この章で使用するコマンドの完全な構文および使用方法の詳細。	*Command Reference (Catalyst 9400 Series Switches)* の「IP マルチキャスト ルーティングのコマンド」の項を参照してください。

標準および RFC

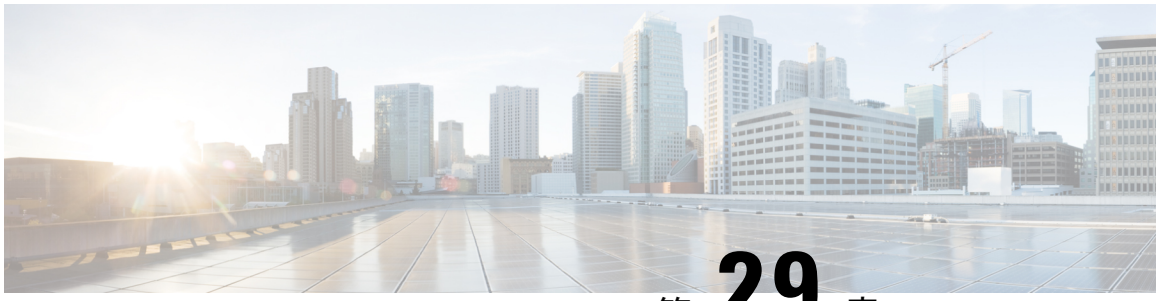
標準/RFC	タイトル
RFC 6763	『*DNS-Based Service Discovery*』
マルチキャスト DNS インターネット（ドラフト）	マルチキャスト

マルチキャスト VRF-Lite の機能履歴と情報

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

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

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



第 29 章

VRF 対応 PBR の設定

- VRF 対応 PBR に関する制約事項 （391 ページ）
- VRF 対応 PBR に関する情報 （391 ページ）
- VRF 対応 PBR の設定方法 （393 ページ）
- VRF 対応 PBR の設定例 （416 ページ）
- VRF 対応 PBR の機能情報 （424 ページ）

VRF 対応 PBR に関する制約事項

- ルートマップコマンドの **set global** と **set vrf** を、同じルートマップでいっしょに設定することはできません。
- 同じ PBR を複数の一意の VRF インターフェイスに適用することはできません。例外は、PBR ポリシーに **set global** または **set vrf** が set コマンドとして含まれている場合です。
- 異なるルートマップ コマンド オプション（**set ip vrf**、**set ip default vrf**、**set vrf**）を、同じシーケンスまたは異なるシーケンスで同じルートマップに設定することはできません。ルートマップで異なるシーケンス番号を使用して複数の一意のルート マップ コマンド オプション（**set vrf** など）を設定できます。

VRF 対応 PBR に関する情報

概要

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。

Cisco IOS XE 16.12.1 リリース以降では、VRF Lite インターフェイスで PBR を設定できます。

MPLS は、PBR が設定されている VRF Lite インターフェイスでは設定できません。

VRF 対応 PBR には次のタイプのものがあります。

- **継承 VRF**：継承 VRF の場合、VRF のコンテキストは入力インターフェイスに暗黙的に継承されます。パケットは VRF インターフェイスに入り、同じ VRF からポリシールーティングまたは転送されます。VRF ルーティングおよび転送テーブルは、設定されたルートポリシーをパケットに適用するためにルートルックアップが必要な場合に使用されます。
- **VRF 間**：VRF 間の場合は、VRF のコンテキストを明示的に指定する必要があります。この場合、パケットは VRF インターフェイスに入り、別の VRF インターフェイスにポリシールーティングまたは転送されます。
- **VRF からグローバル ルーティング テーブル**：パケットは VRF インターフェイスに入り、グローバル ルーティング テーブルからポリシールーティングまたは転送されます。グローバル ルーティング テーブルのコンテキストは、明示的に指定する必要があります。
- **グローバル ルーティング テーブルから VRF**：パケットはグローバルインターフェイスに入り、VRF インターフェイスからポリシールーティングまたは転送されます。

VRF 対応 PBR の set 句

次のいずれかのオプションにより、PBR パケットによる VRF 選択を有効にすることができます。

- ルートマップ
- グローバル ルーティング テーブル
- 指定された VRF

次の set 句を使用したルートマップコマンドにより、VRF インスタンスのパケットのポリシーベースのルーティングを有効にすることができます。

- **set ip vrf** ***vrf-name*** **next-hop** *ip-address* [*ip-address*]：VRF に対して指定されたネクストホップを使用して、ルートマップの一致基準を満たす IPv4 パケットのルーティング先を示します。
- **set ipv6 vrf** ***vrf-name*** **next-hop** *ip-address* [*ip-address*]：VRF に対して指定されたネクストホップを使用して、ルートマップの一致基準を満たす IPv6 パケットのルーティング先を示します。
- **set global**：グローバル ルーティング テーブルを使用してパケットをルーティングします。このコマンドは、特定の VRF に属する入力パケットをグローバル ルーティング テーブルを介してルーティングするために役立ちます。
- **set vrf**：特定の VRF テーブルを使用して、その VRF に属するインターフェイスのいずれかを介してパケットをルーティングします。VRF テーブルにルートがない場合、そのパケットはドロップされます。

- **set ip global next-hop**：PBR のルートマップの基準に一致する IPv4 パケットを転送するネクストホップを指定します。ネクストホップに到達するためにグローバル ルーティング テーブルを使用します。
- **set ipv6 global next-hop**：PBR のルートマップの基準に一致する IPv6 パケットを転送するネクストホップを指定します。ネクストホップに到達するためにグローバル ルーティング テーブルを使用します。
- **set ip default vrf** *vrf-name* **nexthop** *ip-address* [*ip-address*]：VRF のルーティングテーブルに IP アドレスが存在することを確認します。IP アドレスが存在する場合、パケットのポリシールーティングは行われず、ルーティングテーブルに基づいて転送されます。IP アドレスがルーティングテーブルに存在しない場合、パケットのポリシールーティングが行われ、指定されたネクストホップに送信されます。
- **set ipv6 default vrf** *vrf-name* **nexthop** *ip-address* [*ip-address*]：VRF のルーティングテーブルに IPv6 アドレスが存在することを確認します。IPv6 アドレスが存在する場合、パケットのポリシールーティングは行われず、ルーティングテーブルに基づいて転送されます。IPv6 アドレスがルーティングテーブルに存在しない場合、パケットのポリシールーティングが行われ、指定されたネクストホップに送信されます。
- **set ip default global**：グローバルルーティングに IPv4 VRF を設定します。
- **set ipv6 default global**：グローバルルーティングに IPv6 VRF を設定します。
- **set ip default next-hop**：PBR のルートマップの一致条件を満たした IPv4 パケットのうち、宛先に対する明示ルートが指定されていないものの送信先を指定します。
- **set ipv6 default next-hop**：ポリシールーティングのルートマップの一致条件を満たした IPv6 出力パケットのうち、宛先に対する明示ルートが指定されていないものの送信先を指定します。

VRF 対応 PBR の設定方法

ルートマップでの継承 VRF の設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例：	グローバル 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 access-list** {**standard** | **extended**} [*access-list-name* | *access-list-number*] 例： `Device(config)# ip access-list standard 10`	IP アクセスリストのタイプを指定し、対応するアクセスリストコンフィギュレーション モードを開始します。標準、拡張、または名前付きアクセスリストを指定できます。
ステップ 4	[*sequence-number*] {**permit** | **deny**} *protocol source source-wildcard destination destination-wildcard* 例： `Device(config-ipv4-acl)# 10 permit 133.33.33.0 0.0.0.255`	アクセスリストでパケットを許可または拒否する基準を定義します。
ステップ 5	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] 例： `Device(config-route-map)# route-map vrf1_vrf1 permit 10`	ポリシーベースルーティングを有効にするための条件を定義します。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 6	**match ip-address** {*acl-number* [*acl-number* | *acl-name*] | *acl-name* [*acl-name* | *acl-number*] } 例： `Device(config-route-map)# match ip address 10`	一致したパケットに対してポリシールーティングを実行します。IP アクセスリストと拡張 ACL がサポートされています。
ステップ 7	**match length** *min max* 例： `Device(config-route-map)# match length 64 1500`	パケット長と照合します。
ステップ 8	**set ip next-hop** *ip-address* [*ip-address*] 例： `Device(config-route-map)# set ip next-hop 135.35.35.2`	パケットをルーティングするためのネクスト ホップを指定します。
ステップ 9	**interface HundredGigE** *rack/slot/module/port* 例： `Device(config-if)# interface HundredGigE1/0/11`	100 ギガビット イーサネット 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。

	コマンドまたはアクション	目的
ステップ 10	**no switchport** 例： `Device(config-if)# no switchport`	インターフェイスをレイヤ 3 イーサネットインターフェイスとして設定します。
ステップ 11	**vrf forwarding** *vrf-name* 例： `Device(config-if)vrf forwarding vrf1`	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 12	**ip address** *ip-address subnet-mask* 例： `Device(config-if-vrf)ip address 100.1.1.1 255.255.255.0`	インターフェイスの IP アドレスを入力します。
ステップ 13	**ip policy route-map** *map-tag* 例： `Device(config-if) ip policy route-map vrf1_vrf1`	PBR で使用するルートマップを識別します。
ステップ 14	**end** 例： `Device(config-f)# 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。
ステップ 15	**interface HundredGigE** *rack/slot/module/port* 例： `Device(config)# interface HundredGigE1/0/25`	100 ギガビット イーサネット 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 16	**no switchport** 例： `Device(config-if)# no switchport`	インターフェイスをレイヤ 3 イーサネットインターフェイスとして設定します。
ステップ 17	**vrf forwarding** *vrf-name* 例： `Device(config-if)# vrf forwarding vrf1`	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 18	**ip address** *ip-address subnet-mask* 例： `Device(config-if-vrf)ip address 135.35.35.1 255.255.255.0`	インターフェイスの IP アドレスを入力します。

ルートマップでの IPv6 継承 VRF の設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip access-list** {**standard** | **extended**} [*access-list-name* | *access-list-number*] 例： `Device(config)# ipv6 access-list acl_vrf1`	IP アクセス リストのタイプを指定し、対応するアクセス リスト コンフィギュレーション モードを開始します。標準、拡張、または名前付きアクセス リストを指定できます。
ステップ 4	[*sequence-number*] {**permit** | **deny**} *protocol source source-wildcard destination destination-wildcard* 例： `Device(config-ipv6-acl)# 10 permit ipv6 1333::/64 2000::/64`	アクセス リストでパケットを許可または拒否する基準を定義します。
ステップ 5	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] 例： `Device(config-route-map)# route-map vrf1_vrf1_v6 permit 10`	ポリシーベース ルーティングを有効にするための条件を定義します。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 6	**match ip-address** {*acl-number* [*acl-number* | *acl-name*] | *acl-name* [*acl-name* | *acl-number*] } 例： `Device(config-route-map)# match ipv6 address acl_vrf1`	一致したパケットに対してポリシー ルーティングを実行します。IP アクセス リストと拡張 ACL がサポートされています。
ステップ 7	**match length min max** 例： `Device(config-route-map)# match length 64 1500`	パケット長と照合します。

	コマンドまたはアクション	目的
ステップ 8	**set ip next-hop** *ip-address*[*ip-address*] 例： `Device(config-route-map)# set ipv6 next-hop 1335::1`	IPv6 ルーティングパケットのネクストホップを指定します。
ステップ 9	**interface HundredGigE** *rack/slot/module/port* 例： `Device(config-if)# interface HundredGigE1/0/11`	100 ギガビットイーサネット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 10	**no switchport** 例： `Device(config-if)# no switchport`	インターフェイスをレイヤ 3 イーサネットインターフェイスとして設定します。
ステップ 11	**vrf forwarding** *vrf-name* 例： `Device(config-if)vrf forwarding vrf1`	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 12	**ip address** *ip-address subnet-mask* 例： `Device(config-if-vrf) ipv6 address 1000::1/64`	インターフェイスの IP アドレスを入力します。
ステップ 13	**ip policy route-map** *map-tag* 例： `Device(config-if)ipv6 policy route-map vrf1_vrf1_v6`	PBR で使用するルートマップを識別します。
ステップ 14	**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。
ステップ 15	**interface HundredGigE** *rack/slot/module/port* 例： `Device(config)# interface HundredGigE1/0/25`	100 ギガビットイーサネット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 16	**no switchport** 例： `Device(config-if)# no switchport`	インターフェイスをレイヤ 3 イーサネットインターフェイスとして設定します。
ステップ 17	**vrf forwarding** *vrf-name* 例：	VRF をレイヤ 3 インターフェイスに対応付けます。

	コマンドまたはアクション	目的
	`Device(config-if)# vrf forwarding vrf1`	
ステップ 18	**ip address** *ip-address subnet-mask* 例： `Device(config-if-vrf) ipv6 address 1335::2/64`	インターフェイスのIPアドレスを入力します。
ステップ 19	**ipv6 enable** 例： `Device(cofig-if) ipv6 enable`	明示的なIPv6アドレスが設定されていないインターフェイスにおけるIPv6処理をイネーブルにします。

ルートマップでの VRF 間の設定

始める前に

route-map コマンドの次の set 句を使用できます。

- **set ip vrf** ***vrf-name*****next-hop***ip-address*[*ip-address*]：VRF に対して指定されたネクストホップを使用して、ルートマップの一致基準を満たす IPv4 パケットのルーティング先を示します。
- **set ip default vrf** *vrf-name***nexthop***ip-address*[*ip-address*]：VRF のルーティングテーブルに IP アドレスが存在することを確認します。IPアドレスが存在する場合、パケットのポリシールーティングは行われず、ルーティングテーブルに基づいて転送されます。IPアドレスがルーティングテーブルに存在しない場合、パケットのポリシールーティングが行われ、指定されたネクストホップに送信されます。
- **set vrf**：特定の VRF テーブルを使用して、その VRF に属するインターフェイスのいずれかを介してパケットをルーティングします。VRF テーブルにルートがない場合、そのパケットはドロップされます。

手順

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

	コマンドまたはアクション	目的
ステップ 3	**ip access-list** {**standard** | **extended**} [*access-list-name* | *access-list-number*] 例： `Device# ip access-list standard 10`	IPアクセスリストのタイプを指定し、対応するアクセスリストコンフィギュレーション モードを開始します。標準、拡張、または名前付きアクセスリストを指定できます。
ステップ 4	[*sequence-number*] {**permit** | **deny**}*protocol source source-wildcard destination destination-wildcard* 例： `Device(config-ipv4-acl)# 10 permit 133.33.33.0 0.0.0.255`	アクセスリストでパケットを許可または拒否する基準を定義します。一致基準は、IP アドレス、IP アドレスの範囲、および他の IP パケット アクセスリストのフィルタリング オプションに基づいて定義できます。サポートされるアクセスリストは、名前付きアクセスリスト、番号付きアクセスリスト、標準アクセスリスト、および拡張アクセスリストです。一致基準の定義には Cisco IOS ソフトウェアのすべての IP アクセスリスト設定オプションを使用できます。
ステップ 5	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] 例： `Device(config-route-map)# route-map vrf1_vrf2 permit 10`	あるルーティング プロトコルから別のルーティング プロトコルへルートを再配布する条件を定義するか、ポリシー ルーティングをイネーブルにします。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 6	**match ip-address** {*acl-number* [*acl-number* | *acl-name*] | *acl-name* [*acl-name* | *acl-number*] } 例： `Device(config-route-map)# match ip address 10`	標準アクセスリストまたは拡張アクセスリストで宛先ネットワーク番号のアドレスが許可されているルートを配布し、一致したパケットのポリシー ルーティングを行います。 • •IP アクセスリストがサポートされます。 • •この例は、標準アクセスリスト 1 を使用して一致基準を定義するように、ルートマップを設定しています。
ステップ 7	**set ip vrf** *vrf-name* **next-hop** {*ip-address* [*ip-address*] | } • **set ip default vrf***vrf-name***next-hop** {*ip-address* [*ip-address*] | }	**set ip vrf** ***vrf-name*****next-hop***ip-address*[*ip-address*] コマンドは、VRF に対して指定されたネクストホップを使用して、ルート

	コマンドまたはアクション	目的

	• **set vrf***vrf-name*
	例：
	`Device(config-route-map)# set ip vrf vrf2 next-hop 135.35.35.2`
	`or`
	`Device(config-route-map)# set ip default vrf vrf2 next-hop 135.35.35.2`
	`or`
	`Device(config-route-map)# set vrf vrf2`
		マップの一致基準を満たすIPv4パケットのルーティング先を示します。
		default キーワードにより、VRFのルーティングテーブルにIPアドレスが存在することが確認されます。IPアドレスが存在する場合、パケットのポリシールーティングは行われず、ルーティングテーブルに基づいて転送されます。IPアドレスがルーティングテーブルに存在しない場合、パケットのポリシールーティングが行われ、指定されたネクストホップに送信されます。
		set vrf キーワードを使用すると、特定のVRFテーブルを使用して、そのVRFに属するインターフェイスのいずれかを介してパケットがルーティングされます。VRFテーブルにルートがない場合、そのパケットはドロップされます。

ステップ 8	**interface HundredGigE** *rack/slot/module/port* 例： `Device(config-if)# interface HundredGigE1/0/11`	100ギガビットイーサネット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 9	**no switchport** 例： `Device(config-if)# no switchport`	インターフェイスをレイヤ3イーサネットインターフェイスとして設定します。
ステップ 10	**vrf forwarding** *vrf-name* 例： `Device(config-if)# vrf forwarding vrf1`	VRFをレイヤ3インターフェイスに対応付けます。
ステップ 11	**ip address** *ip-address subnet-mask* 例： `Device(config-if-vrf)# ip address 100.1.1.1 255.255.255.0`	インターフェイスのIPアドレスを入力します。
ステップ 12	**ip policy route-map** *map-tag* 例： `Device(config-if)# ip policy route-map vrf1_vrf2`	PBRで使用するルートマップを識別します。

	コマンドまたはアクション	目的
ステップ 13	**end** 例： `Device(config-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。
ステップ 14	**interface HundredGigE** *rack/slot/module/port* 例： `Device(config)# interface HundredGigE1/0/25`	100 ギガビット イーサネット 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 15	**no switchport** 例： `Device(config-if)# no switchport`	インターフェイスをレイヤ 3 イーサネットインターフェイスとして設定します。
ステップ 16	**vrf forwarding** *vrf-name* 例： `Device(config-if)# vrf forwarding vrf2`	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 17	**ip address** *ip-address subnet-mask* 例： `Device(config-if-vrf)ip address 135.35.35.1 255.255.255.0`	インターフェイスの IP アドレスを入力します。

ルートマップでの IPv6 VRF 間の設定

始める前に

route-map コマンドの次の set 句を使用できます。

- **set ipv6 vrf** ***vrf-name*** **next-hop***ip-address*[*ip-address*]：VRF に対して指定されたネクストホップを使用して、ルートマップの一致基準を満たす IPv6 パケットのルーティング先を示します。
- **set ip default vrf** *vrf-name***nexthop***ip-address*[*ip-address*]：VRF のルーティングテーブルに IP アドレスが存在することを確認します。IP アドレスが存在する場合、パケットのポリシールーティングは行われず、ルーティングテーブルに基づいて転送されます。IP アドレスがルーティングテーブルに存在しない場合、パケットのポリシールーティングが行われ、指定されたネクストホップに送信されます。
- **set vrf**：特定の VRF テーブルを使用して、その VRF に属するインターフェイスのいずれかを介してパケットをルーティングします。VRF テーブルにルートがない場合、そのパケットはドロップされます。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip access-list** {**standard** | **extended**} [*access-list-name* | *access-list-number*] 例： `Device# ipv6 access-list acl_vrf1`	IP アクセス リストのタイプを指定し、対応するアクセス リスト コンフィギュレーション モードを開始します。標準、拡張、または名前付きアクセス リストを指定できます。
ステップ 4	[*sequence-number*] {**permit** | **deny**} *protocol source source-wildcard destination destination-wildcard* 例： `Device(config-ipv6-acl)# 10 permit ipv6 1333::/64 2000::/64`	アクセス リストでパケットを許可または拒否する基準を定義します。一致基準は、IPv6 アドレス、IPv6 アドレスの範囲、および他の IPv6 パケット アクセス リストのフィルタリング オプションに基づいて定義できます。サポートされるアクセス リストは、名前付きアクセス リスト、番号付きアクセス リスト、標準アクセス リスト、および拡張アクセス リストです。一致基準の定義には Cisco IOS ソフトウェアのすべての IPv6 アクセス リスト設定オプションを使用できます。
ステップ 5	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] 例： `Device(config-route-map)# route-map vrf1_vrf2_v6 permit 10`	あるルーティング プロトコルから別のルーティング プロトコルへルートを再配布する条件を定義するか、ポリシー ルーティングをイネーブルにします。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 6	**match ip-address** {*acl-number* [*acl-number* | *acl-name*] | *acl-name* [*acl-name* | *acl-number*] } 例： `Device(config-route-map)# match ipv6 address acl_vrf1`	標準アクセス リストまたは拡張アクセス リストで宛先ネットワーク番号のアドレスが許可されているルートを配布し、一致したパケットのポリシー ルーティングを行います。

	コマンドまたはアクション	目的
		• •IPv6アクセスリストがサポートされます。 • •この例は、標準アクセスリスト 1 を使用して一致基準を定義するように、ルートマップを設定しています。

ステップ 7	**set ip vrf** *vrf-name* **next-hop** {*ip-address* [*ip-address*]	}
		• **set ip default vrf***vrf-name* **next-hop** {*ip-address* [*ip-address*]
			}
			• **set vrf***vrf-name*
			例：
			`Device(config-route-map)# set ipv6 vrf vrf2 next-hop 1335::1`
			or
			`Device(config-route-map)# set ipv6 default vrf vrf2 next-hop 1335::1`
			or
			`Device(config-route-map)# set vrf vrf2`
				set ipv6 vrf ***vrf-name*****next-hop***ip-address*[*ip-address*] コマンドは、VRF に対して指定されたネクストホップを使用して、ルートマップの一致基準を満たす IPv4 パケットのルーティング先を示します。
				default キーワードにより、VRF のルーティングテーブルに IP アドレスが存在することが確認されます。IP アドレスが存在する場合、パケットのポリシールーティングは行われず、ルーティングテーブルに基づいて転送されます。IP アドレスがルーティングテーブルに存在しない場合、パケットのポリシールーティングが行われ、指定されたネクストホップに送信されます。

ステップ 8	**interface HundredGigE** *rack/slot/module/port* 例： `Device(config-if)# interface HundredGigE1/0/11`	100 ギガビット イーサネット 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 9	**no switchport** 例： `Device(config-if)# no switchport`	インターフェイスをレイヤ 3 イーサネットインターフェイスとして設定します。
ステップ 10	**vrf forwarding** *vrf-name* 例： `Device(config-if)# vrf forwarding vrf1`	VRF をレイヤ 3 インターフェイスに対応付けます。
ステップ 11	**ip address** *ip-address subnet-mask* 例： `Device(config-if-vrf)# ipv6 address 1000::1/64`	インターフェイスの IP アドレスを入力します。

	コマンドまたはアクション	目的
ステップ 12	**ip policy route-map** *map-tag* 例： `Device(config-if)# ipv6 policy route-map vrf1_vrf2_v6`	PBR で使用するルートマップを識別します。
ステップ 13	**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。
ステップ 14	**interface HundredGigE** *rack/slot/module/port* 例： `Device(config)# interface HundredGigE1/0/25`	100 ギガビット イーサネット 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 15	**no switchport** 例： `Device(config-if)# no switchport`	インターフェイスをレイヤ 3 イーサネットインターフェイスとして設定します。
ステップ 16	**vrf forwarding** *vrf-name* 例： `Device(config-if)vrf forwarding vrf2`	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 17	**ip address** *ip-address subnet-mask* 例： `Device(config-if-vrf) ipv6 address 1335::2/64`	インターフェイスの IP アドレスを入力します。
ステップ 18	**ipv6 enable** 例： `Device(cofig-if) ipv6 enable`	明示的な IPv6 アドレスが設定されていないインターフェイスにおける IPv6 処理をイネーブルにします。

ルートマップでのVRFからグローバルルーティングテーブルオプションの設定

始める前に

route-map コマンドの次の set 句を使用できます。

- **set ip global next hop**：PBR のルートマップの一致基準を満たす IPv4/IPv6 パケットのうち、グローバル ルーティング テーブルが使用されるものの転送先を指定します。
- **set global**：グローバル ルーティング テーブルを使用してパケットをルーティングします。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip access-list** {**standard** | **extended**} [*access-list-name* | *access-list-number*] 例： `Device# ip access-list standard 10`	IP アクセスリストのタイプを指定し、対応するアクセスリスト コンフィギュレーション モードを開始します。標準、拡張、または名前付きアクセスリストを指定できます。
ステップ 4	[*sequence-number*] {**permit** | **deny**} *protocol source source-wildcard destination destination-wildcard* 例： `Device(config-ipv4-acl)# 10 permit 133.33.33.0 0.0.0.255`	アクセスリストでパケットを許可または拒否する基準を定義します。一致基準は、IP アドレス、IP アドレスの範囲、および他の IP パケット アクセスリストのフィルタリング オプションに基づいて定義できます。サポートされるアクセスリストは、名前付きアクセスリスト、番号付きアクセスリスト、標準アクセスリスト、および拡張アクセスリストです。一致基準の定義には Cisco IOS ソフトウェアのすべての IP アクセスリスト設定オプションを使用できます。
ステップ 5	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] 例： `Device(config-route-map)# route-map vrf1_global permit 10`	あるルーティング プロトコルから別のルーティング プロトコルへルートを再配布する条件を定義するか、ポリシー ルーティングをイネーブルにします。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 6	**match ip-address** {*acl-number* [*acl-number* | *acl-name*] | *acl-name* [*acl-name* | *acl-number*] } 例： `Device(config-route-map)# match ip address 10`	標準アクセスリストまたは拡張アクセスリストで宛先ネットワーク番号のアドレスが許可されているルートを転送し、一致したパケットのポリシー ルーティングを行います。

	コマンドまたはアクション	目的
		• •IPアクセスリストがサポートされます。 • •この例は、標準アクセスリスト 1 を使用して一致基準を定義するように、ルートマップを設定しています。
ステップ 7	**set ip default global next-hop** *ip-address* [*ip-address*] • **set global** 例： `Device(config-route-map)# set ip default global next-hop 135.35.35.2 or Device(config-route-map)# set global`	パケットをルーティングするためのネクスト ホップを指定します。
ステップ 8	**interface HundredGigE** *rack/slot/module/port* 例： `Device(config-if)# interface HundredGigE1/0/11`	100 ギガビット イーサネット 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 9	**no switchport** 例： `Device(config-if)# no switchport`	インターフェイスをレイヤ 3 イーサネットインターフェイスとして設定します。
ステップ 10	**vrf forwarding** *vrf-name* 例： `Device(config-if)# vrf forwarding vrf1`	VRF をレイヤ 3 インターフェイスに対応付けます。
ステップ 11	**ip address** *ip-address subnet-mask* 例： `Device(config-if-vrf)#ip address 100.1.1.1 255.255.255.0`	インターフェイスの IP アドレスを入力します。
ステップ 12	**ip policy route-map** *map-tag* 例： `Device(config-if)# ip policy route-map vrf1_global`	PBR で使用するルートマップを識別します。
ステップ 13	**end** 例： `Device(config-f)# 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。

	コマンドまたはアクション	目的
ステップ 14	**interface HundredGigE** *rack/slot/module/port* 例： `Device(config)# interface HundredGigE1/0/25`	100 ギガビット イーサネット 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 15	**no switchport** 例： `Device(config-if)# no switchport`	インターフェイスをレイヤ 3 イーサネットインターフェイスとして設定します。
ステップ 16	**ip address** *ip-address subnet-mask* 例： `Device(config-if-vrf)ip address 135.35.35.1 255.255.255.0`	インターフェイスの IP アドレスを入力します。

ルートマップでの IPv6 VRF からグローバル ルーティング テーブル オプションの設定

始める前に

route-map コマンドの次の set 句を使用できます。

- **set ipv6 global next hop**：PBR のルートマップの一致基準を満たす IPv6 パケットのうち、グローバル ルーティング テーブルが使用されるものの転送先を指定します。
- **set global**：グローバル ルーティング テーブルを使用してパケットをルーティングします。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip access-list** {**standard** | **extended**} [*access-list-name* | *access-list-number*]	IP アクセス リストのタイプを指定し、対応するアクセス リスト コンフィギュレーション モードを開始します。標

	コマンドまたはアクション	目的
	例： `Device# ipv6 access-list acl_vrf1`	準、拡張、または名前付きアクセスリストを指定できます。
ステップ 4	[*sequence-number*]{**permit** | **deny**}*protocol source source-wildcard destination destination-wildcard* 例： `Device(config-ipv6-acl)# 10 permit ipv6 1333::/64 2000::/64`	アクセスリストでパケットを許可または拒否する基準を定義します。一致基準は、IP アドレス、IP アドレスの範囲、および他の IP パケット アクセスリストのフィルタリング オプションに基づいて定義できます。サポートされるアクセスリストは、名前付きアクセスリスト、番号付きアクセスリスト、標準アクセスリスト、および拡張アクセスリストです。一致基準の定義には Cisco IOS ソフトウェアのすべての IP アクセスリスト設定オプションを使用できます。
ステップ 5	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] 例： `Device(config-route-map)# route-map vrf1_global_v6 permit 10`	あるルーティング プロトコルから別のルーティング プロトコルへルートを再配布する条件を定義するか、ポリシー ルーティングをイネーブルにします。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 6	**match ip-address** {*acl-number* [*acl-number* | *acl-name*] | *acl-name* [*acl-name* | *acl-number*] } 例： `Device(config-route-map)# match ipv6 address acl_vrf1`	標準アクセスリストまたは拡張アクセスリストで宛先ネットワーク番号のアドレスが許可されているルートを転送し、一致したパケットのポリシー ルーティングを行います。 • •IP アクセスリストがサポートされます。 • •この例は、標準アクセスリスト 1 を使用して一致基準を定義するように、ルートマップを設定しています。

ステップ 7	**set ip default global next-hop** *ip-address* [*ip-address*]
	• **set global**
	例：
	`Device(config-route-map)# set ipv6 default global next-hop 1335::1`
	or
	`Device(config-route-map)# set global`
		パケットをルーティングするためのネクスト ホップを指定します。

	コマンドまたはアクション	目的
ステップ 8	**interface HundredGigE** *rack/slot/module/port* 例： `Device(config-if)# interface HundredGigE1/0/11`	100 ギガビット イーサネット 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 9	**no switchport** 例： `Device(config-if)# no switchport`	インターフェイスをレイヤ 3 イーサネットインターフェイスとして設定します。
ステップ 10	**vrf forwarding** *vrf-name* 例： `Device(config-if)vrf forwarding vrf1`	VRF をレイヤ 3 インターフェイスに対応付けます。
ステップ 11	**ip address** *ip-address subnet-mask* 例： `Device(config-if-vrf) ipv6 address 1000::1/64`	インターフェイスの IP アドレスを入力します。
ステップ 12	**ip policy route-map** *map-tag* 例： `Device(config-if)ipv6 policy route-map vrf1_global_v6`	PBR で使用するルートマップを識別します。
ステップ 13	**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。
ステップ 14	**interface HundredGigE** *rack/slot/module/port* 例： `Device(config)# interface HundredGigE1/0/25`	100 ギガビット イーサネット 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 15	**no switchport** 例： `Device(config-if)# no switchport`	インターフェイスをレイヤ 3 イーサネットインターフェイスとして設定します。
ステップ 16	**ip address** *ip-address subnet-mask* 例： `Device(config-if-vrf) ipv6 address 1335::2/64`	インターフェイスの IP アドレスを入力します。

	コマンドまたはアクション	目的
ステップ 17	**ipv6 enable** 例： `Device(cofig-if) ipv6 enable`	明示的なIPv6アドレスが設定されていないインターフェイスにおけるIPv6処理をイネーブルにします。

ルートマップでのグローバル ルーティング テーブルから VRF の設定

始める前に

route-map コマンドの次の set 句を使用できます。

- **set ip vrf** ***vrf-name*****next-hop***ip-address*[*ip-address*]：VRF に対して指定されたネクストホップを使用して、ルートマップの一致基準を満たす IPv4 パケットのルーティング先を示します。
- **set ip default vrf** *vrf-name***nexthop***ip-address*[*ip-address*]：VRF のルーティングテーブルに IP アドレスが存在することを確認します。IPアドレスが存在する場合、パケットのポリシールーティングは行われず、ルーティングテーブルに基づいて転送されます。IPアドレスがルーティングテーブルに存在しない場合、パケットのポリシールーティングが行われ、指定されたネクストホップに送信されます。
- **set vrf**：特定の VRF テーブルを使用して、その VRF に属するインターフェイスのいずれかを介してパケットをルーティングします。VRF テーブルにルートがない場合、そのパケットはドロップされます。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip access-list** {**standard** | **extended**} [*access-list-name* | *access-list-number*] 例： `Device# ip access-list standard 10`	IPアクセスリストのタイプを指定し、対応するアクセスリスト コンフィギュレーション モードを開始します。標準、拡張、または名前付きアクセスリストを指定できます。

	コマンドまたはアクション	目的
ステップ 4	[*sequence-number*] {**permit** | **deny**} *protocol source source-wildcard destination destination-wildcard* 例： `Device(config-ipv4-acl)# 10 permit 133.33.33.0 0.0.0.255`	アクセスリストでパケットを許可または拒否する基準を定義します。一致基準は、IP アドレス、IP アドレスの範囲、および他の IP パケット アクセスリストのフィルタリング オプションに基づいて定義できます。サポートされるアクセスリストは、名前付きアクセスリスト、番号付きアクセスリスト、標準アクセスリスト、および拡張アクセスリストです。一致基準の定義には Cisco IOS ソフトウェアのすべての IP アクセスリスト設定オプションを使用できます。
ステップ 5	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] 例： `Device(config-route-map)# route-map global_vrf permit 10`	あるルーティングプロトコルから別のルーティングプロトコルへルートを転送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ます。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 6	**match ip-address** {*acl-number* [*acl-number* | *acl-name*] | *acl-name* [*acl-name* | *acl-number*] } 例： `Device(config-route-map)# match ip address 10`	標準アクセスリストまたは拡張アクセスリストで宛先ネットワーク番号のアドレスが許可されているルートを転送し、一致したパケットのポリシー ルーティングを行います。 • •IP アクセスリストがサポートされます。 • •この例は、標準アクセスリスト 1 を使用して一致基準を定義するように、ルートマップを設定しています。

ステップ 7	**set ip vrf** *vrf-name* **next-hop** *ip-address* [*ip-address*]
	• **set ip default vrf***vrf-name***next-hop** {*ip-address* [*ip-address*]
	• **set vrf***vrf-name*
	例：
	`Device(config-route-map)# set ip vrf vrf2 next-hop 135.35.35.2`
	or
	`Device(config-route-map)# set ip default vrf vrf2 next-hop 135.35.35.2`
		set ip vrf ***vrf-name*****next-hop***ip-address*[*ip-address*] コマンドは、VRF に対して指定されたネクストホップを使用して、ルートマップの一致基準を満たす IPv4 パケットのルーティング先を示します。
		default キーワードにより、VRF のルーティングテーブルに IP アドレスが存在することが確認されます。IP アドレスが存在する場合、パケットのポリシー

	コマンドまたはアクション	目的
	or Device(config-route-map)# set vrf vrf2	ルーティングは行われず、ルーティングテーブルに基づいて転送されます。IP アドレスがルーティングテーブルに存在しない場合、パケットのポリシールーティングが行われ、指定されたネクストホップに送信されます。 **set vrf** キーワードを使用すると、特定の VRF テーブルを使用して、その VRF に属するインターフェイスのいずれかを介してパケットがルーティングされます。VRF テーブルにルートがない場合、そのパケットはドロップされます。
ステップ 8	**interface HundredGigE** *rack/slot/module/port* 例： Device(config-if)# interface HundredGigE1/0/11	100 ギガビット イーサネット 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 9	**no switchport** 例： Device(config-if)# no switchport	インターフェイスをレイヤ 3 イーサネットインターフェイスとして設定します。
ステップ 10	**ip address** *ip-address subnet-mask* 例： Device(config-if-vrf)ip address 100.1.1.1 255.255.255.0	インターフェイスの IP アドレスを入力します。
ステップ 11	**ip policy route-map** *map-tag* 例： Device(config-if) ip policy route-map global_vrf1	PBR で使用するルートマップを識別します。
ステップ 12	**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。
ステップ 13	**interface HundredGigE** *rack/slot/module/port* 例： Device(config)# interface HundredGigE1/0/25	100 ギガビット イーサネット 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。

	コマンドまたはアクション	目的
ステップ 14	**no switchport** 例： `Device(config-if)# no switchport`	インターフェイスをレイヤ 3 イーサネットインターフェイスとして設定します。
ステップ 15	**vrf forwarding** *vrf-name* 例： `Device(config-if)# vrf forwarding vrf2`	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 16	**ip address** *ip-address subnet-mask* 例： `Device(config-if-vrf)# ip address 135.35.35.1 255.255.255.0`	インターフェイスの IP アドレスを入力します。

ルートマップでの IPv6 グローバル ルーティング テーブルから VRF の設定

始める前に

route-map コマンドの次の set 句を使用できます。

- **set ipv6 vrf** ***vrf-name*** **next-hop***ip-address*[*ip-address*]：VRF に対して指定されたネクストホップを使用して、ルートマップの一致基準を満たす IPv6 パケットのルーティング先を示します。
- **set ip default vrf** *vrf-name***nexthop***ip-address*[*ip-address*]：VRF のルーティングテーブルに IP アドレスが存在することを確認します。IP アドレスが存在する場合、パケットのポリシールーティングは行われず、ルーティングテーブルに基づいて転送されます。IP アドレスがルーティングテーブルに存在しない場合、パケットのポリシールーティングが行われ、指定されたネクストホップに送信されます。
- **set vrf**：特定の VRF テーブルを使用して、その VRF に属するインターフェイスのいずれかを介してパケットをルーティングします。VRF テーブルにルートがない場合、そのパケットはドロップされます。

手順

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

	コマンドまたはアクション	目的
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip access-list** {**standard** | **extended**} [*access-list-name* | *access-list-number*] 例： `Device# ipv6 access-list acl_vrf1`	IP アクセスリストのタイプを指定し、対応するアクセスリスト コンフィギュレーション モードを開始します。標準、拡張、または名前付きアクセスリストを指定できます。
ステップ 4	[*sequence-number*] {**permit** | **deny**} *protocol source source-wildcard destination destination-wildcard* 例： `Device(config-ipv6-acl)# 10 permit ipv6 1333::/64 2000::/64`	アクセスリストでパケットを許可または拒否する基準を定義します。一致基準は、IP アドレス、IP アドレスの範囲、および他の IP パケット アクセスリストのフィルタリング オプションに基づいて定義できます。サポートされるアクセスリストは、名前付きアクセスリスト、番号付きアクセスリスト、標準アクセスリスト、および拡張アクセスリストです。一致基準の定義には Cisco IOS ソフトウェアのすべての IP アクセスリスト設定オプションを使用できます。
ステップ 5	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] 例： `Device(config-route-map)# route-map global_vrf_v6 permit 10`	あるルーティングプロトコルから別のルーティングプロトコルへルートを転送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ます。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 6	**match ip-address** {*acl-number* [*acl-number* | *acl-name*] | *acl-name* [*acl-name* | *acl-number*] } 例： `Device(config-route-map)# match ipv6 address acl_vrf1`	標準アクセスリストまたは拡張アクセスリストで宛先ネットワーク番号のアドレスが許可されているルートを転送し、一致したパケットのポリシー ルーティングを行います。 • •IPv6 アクセスリストがサポートされます。 • •この例は、標準アクセスリスト 1 を使用して一致基準を定義するように、ルートマップを設定しています。

	コマンドまたはアクション	目的

ステップ 7	**set ip vrf** *vrf-name* **next-hop** *ip-address* [*ip-address*]
	• **set ip default vrf***vrf-name***next-hop** {*ip-address* [*ip-address*]
	• **set vrf***vrf-name*
	例：
	`Device(config-route-map)# set ipv6 vrf vrf2 next-hop 1335::1`
	`or`
	`Device(config-route-map)# set ipv6 default vrf vrf2 next-hop 1335::1`
	`or`
	`Device(config-route-map)# set vrf vrf2`
		set ipv6 vrf ***vrf-name*****next-hop***ip-address*[*ip-address*] コマンドは、VRF に対して指定されたネクストホップを使用して、ルートマップの一致基準を満たす IPv4 パケットのルーティング先を示します。
		default キーワードにより、VRF のルーティングテーブルに IP アドレスが存在することが確認されます。IP アドレスが存在する場合、パケットのポリシールーティングは行われず、ルーティングテーブルに基づいて転送されます。IP アドレスがルーティングテーブルに存在しない場合、パケットのポリシールーティングが行われ、指定されたネクストホップに送信されます。
		set vrf キーワードを使用すると、特定の VRF テーブルを使用して、その VRF に属するインターフェイスのいずれかを介してパケットがルーティングされます。VRF テーブルにルートがない場合、そのパケットはドロップされます。

ステップ 8	**interface HundredGigE** *rack/slot/module/port* 例： `Device(config-if)# interface HundredGigE1/0/11`	100 ギガビット イーサネット 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 9	**no switchport** 例： `Device(config-if)# no switchport`	インターフェイスをレイヤ 3 イーサネットインターフェイスとして設定します。
ステップ 10	**ip address** *ip-address subnet-mask* 例： `Device(config-if-vrf)# ipv6 address 1000::1/64`	インターフェイスの IP アドレスを入力します。
ステップ 11	**ip policy route-map** *map-tag* 例： `Device(config-if)# ipv6 policy route-map global_vrf_v6`	PBR で使用するルートマップを識別します。

	コマンドまたはアクション	目的
ステップ 12	**end** 例： `Device(config-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。
ステップ 13	**interface HundredGigE** *rack/slot/module/port* 例： `Device(config)# interface HundredGigE1/0/25`	100 ギガビット イーサネット 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 14	**no switchport** 例： `Device(config-if)# no switchport`	インターフェイスをレイヤ 3 イーサネット インターフェイスとして設定します。
ステップ 15	**vrf forwarding** *vrf-name* 例： `Device(config-if)# vrf forwarding vrf2`	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 16	**ip address** *ip-address subnet-mask* 例： `Device(config-if-vrf)# ipv6 address 1335::2/64`	インターフェイスの IP アドレスを入力します。
ステップ 17	**ipv6 enable** 例： `Device(cofig-if)# ipv6 enable`	明示的な IPv6 アドレスが設定されていないインターフェイスにおける IPv6 処理をイネーブルにします。

VRF 対応 PBR の設定例

例：ルートマップにおける VRF インターフェイスの継承 VRF としての設定

次に、ルートマップで VRF インターフェイスを継承 VRF として設定する例を示します。

```
Device(config)# ip access-list standard 10
Device(config-ipv4-acl)# 10 permit 133.33.33.0 0.0.0.255
Device(config-route-map)# route-map vrf1_vrf1 permit 10
Device(config-route-map)# match ip address 10
Device(config-route-map)# match length 64 1500
Device(config-route-map)#  set ip next-hop 135.35.35.2
Device(config-if)# interface HundredGigE1/0/11
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf1
Device(config-if-vrf)# ip address 100.1.1.1 255.255.255.0
```

```
Device(config-if)# ip policy route-map vrf1_vrf1
Device(config-if)# end
Device(config)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf1
Device(config-if-vrf)# ip address 135.35.35.1 255.255.255.0
```

例：ルートマップにおける IPv6 VRF インターフェイスの継承 VRF としての設定

次に、ルートマップで IPv6 VRF インターフェイスを継承 VRF として設定する例を示します。

```
Device(config)# ipv6 access-list acl_vrf1
Device(config-ipv4-acl)# sequence 10 permit ipv6 1333::/64 2000::/64
Device(config-route-map)# route-map vrf1_vrf1_v6 permit 10
Device(config-route-map)# match ipv6 address acl_vrf1
Device(config-route-map)# match length 64 1500
Device(config-route-map)# set ipv6  next-hop 1335::1
Device(config-if)# interface HundredGigE1/0/11
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf1
Device(config-if)# ipv6 address 1000::1/64
Device(config-if)# ipv6 policy route-map vrf1_vrf1_v6

Device(config-if)# end
Device(config)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf1
Device(config-if-vrf)# ipv6 address 1335::2/64
Device(config-if-vrf)# ipv6 enable
```

例：set ip vrf 句を使用したルートマップにおける VRF インターフェイスの VRF 間としての設定

次に、**set ip vrf** 句を使用してルートマップで VRF インターフェイスを VRF 間として設定する例を示します。

```
Device# ip access-list standard 10
Device(config-ipv4-acl)# 10 permit 133.33.33.0 0.0.0.255
Device(config-route-map)# route-map vrf1_vrf2 permit 10
Device(config-route-map)# match ip address 10
Device(config-route-map)# set ip  vrf vrf2 next-hop 135.35.35.2
Device(config-if)# interface HundredGigE1/0/11
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf1
Device(config-if-vrf)# ip address 100.1.1.1 255.255.255.0
Device(config-if)# ip policy route-map vrf1_vrf1
Device(config-if)# end
Device(config)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf2
Device(config-if-vrf)# ip address 135.35.35.1 255.255.255.0
```

例：set ip vrf 句を使用したルートマップにおける VRF インターフェイスの IPv6 VRF 間としての設定

次に、**set ip vrf** 句を使用してルートマップで IPv6 VRF インターフェイスを VRF 間として設定する例を示します。

```
Device# ipv6 access-list acl_vrf1
Device(config-ipv4-acl)# sequence 10 permit ipv6 1333::/64 2000::/64
Device(config-route-map)# route-map vrf1_vrf2_v6 permit 10
Device(config-route-map)# match ipv6 address acl_vrf1
Device(config-route-map)# set ipv6 vrf vrf2 next-hop 1335::1
Device(config-if)# interface HundredGigE1/0/11
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf1
Device(config-if)# ipv6 address 1000::1/64
Device(config-if)# ipv6 policy route-map vrf1_vrf1_v6
Device(config-if)# end
Device(config)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf2
Device(cofig-if-vrf)# ipv6 address 1335::2/64
Device(config-if-vrf)# ipv6 enable
```

例：set ip default vrf 句を使用したルートマップにおける VRF インターフェイスの VRF 間としての設定

次に、**set ip vrf** 句を使用してルートマップで VRF インターフェイスを VRF 間として設定する例を示します。

```
Device# ip access-list standard 10
Device(config-ipv4-acl)# 10 permit 133.33.33.0 0.0.0.255
Device(config-route-map)# route-map vrf1_vrf2 permit 10
Device(config-route-map)# match ip address 10
Device(config-route-map)# set ip default vrf vrf2 next-hop 135.35.35.2
Device(config-if)# interface HundredGigE1/0/11
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf1
Device(config-if-vrf)#ip address 100.1.1.1 255.255.255.0
Device(config-if-vrf)# ip policy route-map vrf1_vrf2
Device(config-if-vrf)# end
Device(config-if)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf2
Device(config-if-vrf)# ip address 135.35.35.1 255.255.255.0
```

例：set ip default vrf 句を使用したルートマップにおける IPv6 VRF インターフェイスの VRF 間としての設定

次に、**set ip vrf** 句を使用してルートマップで IPv6 VRF インターフェイスを VRF 間として設定する例を示します。

```
Device# ipv6 access-list acl_vrf1
Device(config-ipv6-acl)# sequence 10 permit ipv6 1333::/64 2000::/64
Device(config-route-map)# route-map vrf1_vrf2_v6 permit 10
```

```
Device(config-route-map)#  match ipv6 address acl_vrf1
Device(config-route-map)#  set ipv6  default vrf vrf2 next-hop 1335::1
Device(config-if)# interface HundredGigE1/0/11
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf1
Device(config-if-vrf)# ipv6 address 1000::1/64
Device(config-if-vrf)# ipv6 policy route-map vrf1_vrf2_v6
Device(config-if-vrf)# end
Device(config-if)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf2
Device(config-if-vrf)# ipv6 address 1335::2/64
Device(config-if-vrf)# ipv6 enable
```

例：set vrf 句を使用したルートマップにおける VRF インターフェイスの VRF 間としての設定

次に、**set vrf** 句を使用してルートマップで VRF インターフェイスを VRF 間として設定する例を示します。

```
Device# ip access-list standard 10
Device(config-ipv4-acl)# 10 permit 133.33.33.0 0.0.0.255
Device(config-route-map)# route-map vrf1_vrf2 permit 10
Device(config-route-map)# match ip address 10
Device(config-route-map)# set vrf vrf2
Device(config-if)# interface HundredGigE1/0/11
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf1
Device(config-if-vrf)# ip address 100.1.1.1 255.255.255.0
Device(config-if)# ip policy route-map vrf1_vrf2
Device(config-if)# end
Device(config)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf2
Device(config-if-vrf)# ip address 135.35.35.1 255.255.255.0
```

例：set vrf 句を使用したルートマップにおける IPv6 VRF インターフェイスの VRF 間としての設定

次に、**set vrf** 句を使用してルートマップで IPv6 VRF インターフェイスを VRF 間として設定する例を示します。

```
Device# ipv6 access-list acl_vrf1
Device(config-ipv4-acl)# sequence 10 permit ipv6 1333::/64 2000::/64
Device(config-route-map)# route-map vrf1_vrf2_v6 permit 10
Device(config-route-map)# match ipv6 address acl_vrf1
Device(config-route-map)# set vrf vrf2
Device(config-if)# interface HundredGigE1/0/11
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf1
Device(config-if)# ipv6 address 1000::1/64
Device(confog-f)# ipv6 policy route-map vrf1_vrf2_v6
Device(config-if)# end
Device(config)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf2
```

```
Device(config-if-vrf)# ipv6 address 1335::2/64
Device(config-if-vrf)# ipv6 enable
```

例：set ip default global 句を使用したルートマップでの VRF からグローバル ルーティング テーブルの設定

次に、**set ip default global** 句を使用してパケットをルートマップで VRF からグローバル ルーティング テーブルに設定する例を示します。

```
Device# ip access-list standard 10
Device(config-ipv4-acl)# 10 permit 133.33.33.0 0.0.0.255
Device(config-route-map)# route-map vrf1_global permit 10
Device(config-route-map)# match ip address 10
Device(config-route-map)# set ip default global next-hop 135.35.35.2
Device(config-if)# interface HundredGigE1/0/11
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf1
Device(config-if-vrf)# ip address 100.1.1.1 255.255.255.0
Device(config-if)# ip policy route-map vrf1_global
Device(config-if)# end
Device(config)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if-vrf)# ip address 135.35.35.1 255.255.255.0
```

例：set ip default global 句を使用したルートマップでの IPv6 VRF からグローバル ルーティング テーブルの設定

次に、**set ip default global** 句を使用してパケットをルートマップで IPv6 VRF からグローバル ルーティング テーブルに設定する例を示します。

```
Device# ipv6 access-list acl_vrf1
Device(config-ipv4-acl)#  sequence 10 permit ipv6 1333::/64 2000::/64
Device(config-route-map)# route-map vrf1_global_v6 permit 10
Device(config-route-map)# match ipv6 address acl_vrf1
Device(config-route-map)#  set ipv6 default  global next-hop 1335::1
Device(config-if)# interface HundredGigE1/0/11
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf1
Device(config-if)# ipv6 address 1000::1/64
Device(config-if)# ipv6 policy route-map vrf1_global_v6
Device(config-if)# end
Device(config)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if-vrf)# ipv6 address 1335::2/64
Device(config-if-vrf)# ipv6 enable
```

例：set global 句を使用したルートマップでの VRF からグローバル ルーティング テーブルの設定

次に、**set global** 句を使用してパケットをルートマップで VRF からグローバル ルーティング テーブルに設定する例を示します。

```
Device# ip access-list standard 10
Device(config-ipv4-acl)# 10 permit 133.33.33.0 0.0.0.255
Device(config-route-map)# route-map vrf1_global permit 10
Device(config-route-map)# match ip address 10
Device(config-route-map)# set global
Device(config-if)# interface HundredGigE1/0/11
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf1
Device(config-if-vrf)# ip address 100.1.1.1 255.255.255.0
Device(config-if)# ip policy route-map vrf1_global
Device(config-if)# end
Device(config)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if-vrf)ip address 135.35.35.1 255.255.255.0
```

例：set global 句を使用したルートマップでの IPv6 VRF からグローバル ルーティング テーブルの設定

次に、**set global** 句を使用してパケットをルートマップで IPv6 VRF からグローバル ルーティング テーブルに設定する例を示します。

```
Device# ipv6 access-list acl_vrf1
Device(config-ipv6-acl)#  sequence 10 permit ipv6 1333::/64 2000::/64
Device(config-route-map)# route-map vrf1_global_v6 permit 10
Device(config-route-map)# match ipv6 address acl_vrf1
Device(config-route-map)# set global
Device(config-if)# interface HundredGigE1/0/11
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf1
Device(config-if-vrf)# ipv6 address 1000::1/64
Device(config-if)# ipv6 policy route-map vrf1_global_v6
Device(config-if)# end
Device(config)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if-vrf)# ipv6 address 1335::2/64
Device(config-if-vrf)# ipv6 enable
```

例：set ip vrf 句を使用したルートマップでのグローバル ルーティング テーブルから VRF の設定

次に、**set ip vrf** 句を使用してパケットのルーティングと転送をルートマップでグローバル ルーティング テーブルから VRF に設定する例を示します。

```
Device# ip access-list standard 10
Device(config-ipv4-acl)# 10 permit 133.33.33.0 0.0.0.255
Device(config-route-map)# route-map global_vrf permit 10
Device(config-route-map)# match ip address 10
Device(config-route-map)# set ip vrf vrf2 next-hop 135.35.35.2
Device(config-if)# interface HundredGigE1/0/11
Device(config-if)# no switchport
Device(config-if-vrf)# ip address 100.1.1.1 255.255.255.0
Device(config-if)# ip policy route-map global_vrf
Device(config-if)# end
Device(config)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf2
Device(config-if-vrf)# ip address 135.35.35.1 255.255.255.0
```

例：set ipv6 vrf 句を使用したルートマップでのグローバル ルーティング テーブルから IPv6 VRF の設定

次に、**set ipv6 vrf** 句を使用してパケットのルーティングと転送をルートマップでグローバル ルーティング テーブルから IPv6 VRF に設定する例を示します。

```
Device# ipv6 access-list acl_vrf1
Device(config-ipv4-acl)# sequence 10 permit ipv6 1333::/64 2000::/64
Device(config-route-map)# route-map global_vrf_v6 permit 10
Device(config-route-map)# match ipv6 address acl_vrf1
Device(config-route-map)# set ipv6 vrf vrf2 next-hop 1335::1
Device(config-if)# interface HundredGigE1/0/11
Device(config-if)# no switchport
Device(config-if-vrf)# ipv6 address 1000::1/64
Device(config-if)#  ipv6 policy route-map global_vrf_v6
Device(config-if)# end
Device(config)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf2
Device(config-if-vrf)# ipv6 address 1335::2/64
Device(config-if-vrf)# ipv6 enable
```

例：set ip default vrf 句を使用したルートマップでのグローバル ルーティング テーブルから VRF の設定

次に、**set ip vrf** 句を使用してパケットのルーティングと転送をルートマップでグローバル ルーティング テーブルから VRF に設定する例を示します。

```
Device# ip access-list standard 10
Device(config-ipv4-acl)# 10 permit 133.33.33.0 0.0.0.255
Device(config-route-map)# route-map global_vrf permit 10
Device(config-route-map)# match ip address 10
Device(config-route-map)# set ip default vrf vrf2 next-hop 135.35.35.2
Device(config-if)# interface HundredGigE1/0/11
Device(config-if-vrf)# ip address 100.1.1.1 255.255.255.0
Device(config-if-vrf)#ip policy route-map global_vrf
Device(config-if)# end
Device(config)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf2
Device(config-if-vrf)# ip address 135.35.35.1 255.255.255.0
```

例：set ipv6 default vrf 句を使用したルートマップでのグローバル ルーティング テーブルから IPv6 VRF の設定

次に、**set ipv6 default vrf** 句を使用してパケットのルーティングと転送をルートマップでグローバル ルーティング テーブルから VRF に設定する例を示します。

```
Device# ipv6 access-list acl_vrf1
Device(config-ipv4-acl)# sequence 10 permit ipv6 1333::/64 2000::/64
Device(config-route-map)# route-map global_vrf_v6 permit 10
Device(config-route-map)# match ipv6 address acl_vrf1
Device(config-route-map)# set ipv6 default vrf vrf2 next-hop 1335::1
Device(config-if)# interface HundredGigE1/0/11
```

```
Device(config-if-vrf)#  ipv6 address 1000::1/64
Device(config-if-vrf)# ipv6 policy route-map global_vrf_v6
Device(config-if)# end
Device(config)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf2
Device(config-if-vrf)# ipv6 address 1335::2/64
Device(config-if-vrf)# ipv6 enable
```

例：set vrf 句を使用したルートマップでのグローバル ルーティング テーブルから VRF の設定

次に、**set vrf** 句を使用してパケットのルーティングと転送をルートマップでグローバル ルーティング テーブルから VRF に設定する例を示します。

```
Device# ip access-list standard 10
Device(config-ipv4-acl)# 10 permit 133.33.33.0 0.0.0.255
Device(config-route-map)# route-map global_vrf permit 10
Device(config-route-map)# match ip address 10
Device(config-route-map)# set vrf vrf2
Device(config-if)# interface HundredGigE1/0/11
Device(config-if)# no switchport
Device(config-if-vrf)# ip address 100.1.1.1 255.255.255.0
Device(config-if)# ip policy route-map global_vrf
Device(config-if)# end
Device(config)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf2
Device(config-if-vrf)#ip address 135.35.35.1 255.255.255.0
```

例：set vrf 句を使用したルートマップでのグローバル ルーティング テーブルから IPv6 VRF の設定

次に、**set vrf** 句を使用してパケットのルーティングと転送をルートマップでグローバル ルーティング テーブルから IPv6 VRF に設定する例を示します。

```
Device# ipv6 access-list acl_vrf1
Device(config-ipv4-acl)# sequence 10 permit ipv6 1333::/64 2000::/64
Device(config-route-map)# route-map global_vrf_v6 permit 10
Device(config-route-map)# match ipv6 address acl_vrf1
Device(config-route-map)# set vrf vrf2
Device(config-if)# interface HundredGigE1/0/11
Device(config-if)# no switchport
Device(config-if-vrf)#  ipv6 address 1000::1/64
Device(config-if)#  ipv6 policy route-map global_vrf_v6
Device(config-if)# end
Device(config)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf2
Device(config-if-vrf)#ipv6 address 1335::2/64
Device(config-if-verf)# ipv6 enable
```

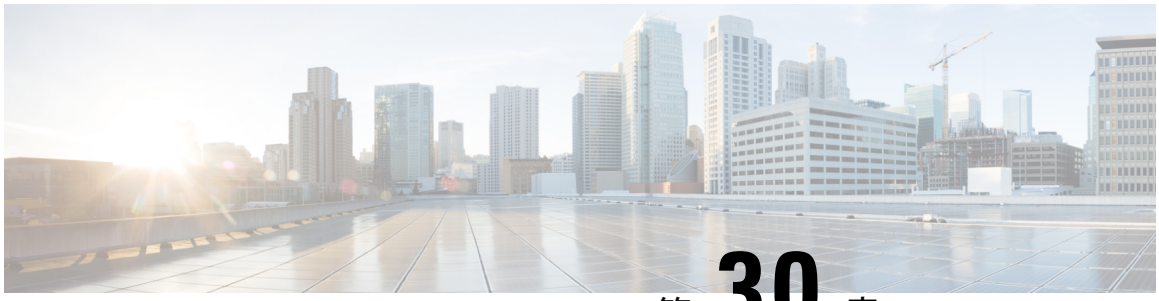
VRF 対応 PBR の機能情報

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

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

表 *45 : VRF* 対応 *PBR* の機能情報

機能名	リリース	機能情報
VRF 対応 PBR	Cisco IOS XE Gibraltar 16.12.1	この機能が導入されました。



第 30 章

Multi-VRF CE の設定

- Multi-VRF CE に関する情報 （425 ページ）
- Multi-VRF CE の設定方法 （429 ページ）
- Multi-VRF CE の設定方法 （434 ページ）
- VRF 認識サービスの設定 （440 ページ）
- Multi-VRF CE の設定例 （449 ページ）
- マルチ VRF CE の機能情報 （453 ページ）

Multi-VRF CE に関する情報

バーチャル プライベート ネットワーク（VPN）は、ISP バックボーン ネットワーク上でお客様にセキュアな帯域幅共有を提供します。VPN は、共通ルーティング テーブルを共有するサイトの集合です。カスタマー サイトは、1 つまたは複数のインターフェイスでサービス プロバイダー ネットワークに接続され、サービス プロバイダーは、VRF テーブルと呼ばれる VPN ルーティング テーブルと各インターフェイスを関連付けます。

スイッチが で稼働している場合、スイッチはカスタマーエッジ（CE）デバイスの Multiple VPN Routing/Forwarding（Multi-VRF）インスタンスをサポートします（Multi-VRF CE）。サービス プロバイダーは、Multi-VRF CE により、重複する IP アドレスで複数の VPN をサポートできます。

（注） スイッチでは、VPN のサポートのためにマルチプロトコル ラベル スイッチング（MPLS）が使用されません。

Multi-VRF CE の概要

Multi-VRF CE は、サービス プロバイダーが複数の VPN をサポートし、VPN 間で IP アドレスを重複して使用できるようにする機能です。Multi-VRF CE は入力インターフェイスを使用して、さまざまな VPN のルートを区別し、1 つまたは複数のレイヤ 3 インターフェイスと各 VRF を関連付けて仮想パケット転送テーブルを形成します。VRF 内のインターフェイスは、イーサ

ネット ポートのように物理的なもの、または VLAN SVI のように論理的なものにもできますが、複数の VRF に属すことはできません。

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

Multi-VRF CE には、次のデバイスが含まれます。

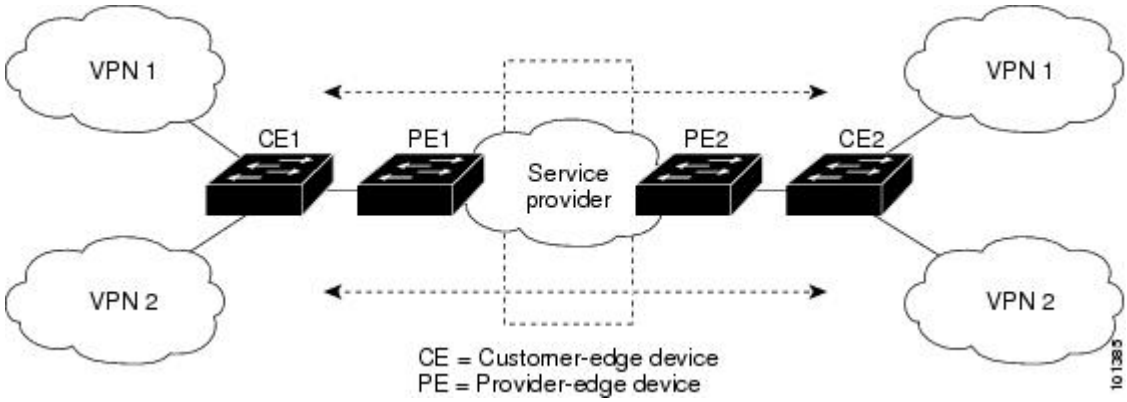
- お客様は、CE デバイスにより、1 つまたは複数のプロバイダー エッジ（PE）ルータへのデータ リンクを介してサービス プロバイダー ネットワークにアクセスできます。CE デバイスは、サイトのローカル ルートをルータにアドバタイズし、リモート VPN ルートをそこから学習します。スイッチを CE に設定することができます。
- PE ルータは、スタティック ルーティング、または BGP、RIPv2、OSPF、EIGRP などのルーティング プロトコルを使用して、CE デバイスとルーティング情報を交換します。PE は、直接接続している VPN に対する VPN ルートのみを保守する必要があります。そのため、すべてのサービス プロバイダー VPN ルートを PE が保守する必要はありません。各 PE ルータは、直接接続しているサイトごとに VRF を維持します。すべてのサイトが同じ VPN に存在する場合は、PE ルータの複数のインターフェイスを 1 つの VRF に関連付けることができます。各 VPN は、指定された VRF にマッピングされます。PE ルータは、ローカル VPN ルートを CE から学習したあとで、IBGP を使用して別の PE ルータと VPN ルーティング情報を交換します。
- CE デバイスに接続していないサービス プロバイダー ネットワークのルータは、プロバイダー ルータやコア ルータになります。

Multi-VRF CE では、複数のお客様が 1 つの CE を共有でき、CE と PE の間で 1 つの物理リンクだけが使用されます。共有 CE は、お客様ごとに別々の VRF テーブルを維持し、独自のルーティング テーブルに基づいて、お客様ごとにパケットをスイッチングまたはルーティングします。Multi-VRF CE は、制限付きの PE 機能を CE デバイスに拡張して、別々の VRF テーブルを維持し、VPN のプライバシーおよびセキュリティをブランチ オフィスに拡張します。

ネットワーク トポロジ

次の図に、スイッチを複数の仮想 CE として使用した構成例を示します。このシナリオは、中小企業など、VPN サービスの帯域幅要件の低いお客様に適しています。この場合、スイッチにはマルチ VRF CE のサポートが必要です。Multi-VRF CE はレイヤ 3 機能なので、VRF のそれぞれのインターフェイスはレイヤ 3 インターフェイスである必要があります。

図 *16* : 複数の仮想 *CE* として機能するスイッチ



CE スイッチは、レイヤ3インターフェイスを VRF に追加するコマンドを受信すると、Multi-VRF CE 関連のデータ構造で VLAN ID と Policy Label（PL）の間に適切なマッピングを設定し、VLAN ID と PL を VLAN データベースに追加します。

Multi-VRF CE を設定すると、レイヤ 3 フォワーディング テーブルは、次の 2 つのセクションに概念的に分割されます。

- Multi-VRF CE ルーティング セクションには、さまざまな VPN からのルートが含まれます。
- グローバル ルーティング セクションには、インターネットなど、VPN 以外のネットワークへのルートが含まれます。

さまざまな VRF の VLAN ID はさまざまな PL にマッピングされ、処理中に VRF を区別するために使用されます。レイヤ 3 設定機能では、学習した新しい VPN ルートごとに、入力ポートの VLAN ID を使用して PL を取得し、Multi-VRF CE ルーティング セクションに PL および新しいルートを挿入します。ルーテッド ポートからパケットを受信した場合は、ポート内部 VLAN ID 番号が使用されます。SVI からパケットを受信した場合は、VLAN 番号が使用されます。

パケット転送処理

Multi-VRF CE 対応ネットワークのパケット転送処理は次のとおりです。

- スイッチは、VPN からパケットを受信すると、入力 PL 番号に基づいてルーティング テーブルを検索します。ルートが見つかると、スイッチはパケットを PE に転送します。
- 入力 PE は、CE からパケットを受信すると、VRF 検索を実行します。ルートが見つかると、ルータは対応する MPLS ラベルをパケットに追加し、MPLS ネットワークに送信します。
- 出力 PE は、ネットワークからパケットを受信すると、ラベルを除去してそのラベルを使用し、正しい VPN ルーティング テーブルを識別します。次に、通常のルート検索を実行します。ルートが見つかると、パケットを正しい隣接デバイスに転送します。

- CE は、出力 PE からパケットを受信すると、入力 PL を使用して正しい VPN ルーティング テーブルを検索します。ルートが見つかると、パケットを VPN 内で転送します。

ネットワーク コンポーネント

VRF を設定するには、VRF テーブルを作成し、VRF に関連するレイヤ 3 インターフェイスを指定します。次に、VPN、および CE と PE 間でルーティング プロトコルを設定します。プロバイダ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。Multi-VRF CE ネットワークには、次の 3 つの主要コンポーネントがあります。

- VPN ルート ターゲット コミュニティ：VPN コミュニティのその他すべてのメンバのリスト。VPN コミュニティ メンバーごとに VPN ルート ターゲットを設定する必要があります。
- VPN コミュニティ PE ルータのマルチプロトコル BGP ピアリング：VPN コミュニティのすべてのメンバーに VRF 到達可能性情報を伝播します。VPN コミュニティのすべての PE ルータで BGP ピアリングを設定する必要があります。
- VPN 転送：VPN サービス プロバイダー ネットワークを介し、全 VPN コミュニティ メンバー間で、全トラフィックを伝送します。

VRF 認識サービス

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

VRF 認識サービスは、プラットフォームに依存しないモジュールに実装されます。VRF とは、Cisco IOS 内の複数のルーティング インスタンスを意味します。各プラットフォームには、サポートする VRF 数に関して独自の制限があります。

VRF 認識サービスには、次の特性があります。

- ユーザは、ユーザ指定の VRF 内のホストに ping を実行できます。
- ARP エントリは、個別の VRF で学習されます。ユーザは、特定の VRF の ARP エントリを表示できます。

Multi-VRF CE の設定方法

Multi-VRF CE のデフォルト設定

表 *46 : VRF* のデフォルト設定

機能	デフォルト設定
VRF	ディセーブルVRF は定義されていません。
マップ	インポート マップ、エクスポート マップ、ルート マップは定義されていません。
VRF 最大ルート数	ファスト イーサネット スイッチ：8000 ギガビット イーサネット スイッチ：12000
転送テーブル	インターフェイスのデフォルトは、グローバル ルーティング テーブルです。

Multi-VRF CE の設定時の注意事項

（注）

Multi-VRF CE を使用するには、スイッチで をイネーブルにする必要があります。

- Multi-VRF CE を含むスイッチは複数のお客様によって共有され、各お客様には独自のルーティング テーブルがあります。
- お客様は別々の VRF テーブルを使用するので、同じ IP アドレスを再利用できます。別々の VPN では IP アドレスの重複が許可されます。
- Multi-VRF CE では、複数のお客様が、PE と CE の間で同じ物理リンクを共有できます。複数の VLAN を持つトランク ポートでは、パケットがお客様間で分離されます。それぞれのお客様には独自の VLAN があります。
- Multi-VRF CE ではサポートされない MPLS-VRF 機能があります。ラベル交換、LDP 隣接関係、ラベル付きパケットはサポートされません。
- PE ルータの場合、Multi-VRF CE の使用と複数の CE の使用に違いはありません。図 41-6 では、複数の仮想レイヤ 3 インターフェイスが Multi-VRF CE デバイスに接続されています。
- スイッチでは、物理ポートか VLAN SVI、またはその両方の組み合わせを使用して、VRF を設定できます。SVI は、アクセス ポートまたはトランク ポートで接続できます。
- お客様は、別のお客様と重複しないかぎり、複数の VLAN を使用できます。お客様の VLAN は、スイッチに保存されている適切なルーティング テーブルの識別に使用される特定のルーティング テーブル ID にマッピングされます。
- スイッチは、1 つのグローバルネットワークおよび最大 256 の VRF をサポートします。
- Cisco Catalyst 9200 シリーズ スイッチの各モデルでサポートされる VRF の数は次のとおりです。

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

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

- CE と PE の間では、ほとんどのルーティング プロトコル（BGP、OSPF、RIP、およびスタティック ルーティング）を使用できます。ただし、次の理由から External BGP（EBGP）を使用することを推奨します。
  - BGP では、複数の CE とのやり取りに複数のアルゴリズムを必要としません。
  - BGP は、さまざまな管理者によって稼働するシステム間でルーティング情報を渡すように設計されています。
  - BGP では、ルートの属性を CE に簡単に渡すことができます。
- Multi-VRF CE は、パケットのスイッチング レートに影響しません。
- VPN マルチキャストはサポートされません。
- プライベート VLAN で VRF をイネーブルにできます（逆も同様です）。
- インターフェイスでポリシーベース ルーティング（PBR）がイネーブルになっている場合は、VRF をイネーブルにできません（逆も同様です）。
- インターフェイスで Web Cache Communication Protocol（WCCP）がイネーブルになっている場合は、VRF をイネーブルにできません（逆も同様です）。

VRF の設定

次の操作を行ってください。

手順

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

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

	コマンドまたはアクション	目的
ステップ 10	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 11	**show ip vrf** [**brief** | **detail** | **interfaces**] [*vrf-name*] 例： `Device#show ip vrf interfaces vpn1`	設定を確認します。設定したVRFに関する情報を表示します。
ステップ 12	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーションファイルに設定を保存します。

Multi-VRF CE の設定方法

マルチキャスト VRF の設定

手順

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

	コマンドまたはアクション	目的
ステップ 4	**ip vrf** *vrf-name* 例： `Device(config)#ip vrf vpn1`	VRF 名を指定し、VRF コンフィギュレーション モードを開始します。
ステップ 5	**rd** *route-distinguisher* 例： `Device(config-vrf)#rd 100:2`	ルート識別子を指定して VRF テーブルを作成します。AS 番号と任意の番号（xxx:y）または IP アドレスと任意の番号（A.B.C.D:y）を入力します。
ステップ 6	**route-target** {**export** | **import** | **both**} *route-target-ext-community* 例： `Device(config-vrf)#route-target import 100:2`	指定された VRF のインポート、エクスポート、またはインポートおよびエクスポート ルート ターゲット コミュニティのリストを作成します。AS システム番号と任意の番号（xxx:y）または IP アドレスと任意の番号（A.B.C.D:y）を入力します。*route-target-ext-community* は、ステップ 4 で入力した *route-distinguisher* と同一にする必要があります。
ステップ 7	**import map** *route-map* 例： `Device(config-vrf)#import map importmap1`	（任意）VRF にルートマップを対応付けます。
ステップ 8	**ip multicast-routing vrf** *vrf-name* **distributed** 例： `Device(config-vrf)#ip multicast-routing vrf vpn1 distributed`	（任意）VRF テーブルでグローバル マルチキャスト ルーティングをイネーブルにします。
ステップ 9	**interface** *interface-id* 例： `Device(config-vrf)#interface gigabitethernet 1/0/2`	VRF に関連付けるレイヤ 3 インターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。インターフェイスはルーテッド ポートまたは SVI に設定できます。
ステップ 10	**ip vrf forwarding** *vrf-name* 例： `Device(config-if)#ip vrf forwarding vpn1`	VRF をレイヤ 3 インターフェイスに対応付けます。

	コマンドまたはアクション	目的
ステップ 11	**ip address** *ip-address* **mask** 例： `Device(config-if)#ip address 10.1.5.1 255.255.255.0`	レイヤ 3 インターフェイスの IP アドレスを設定します。
ステップ 12	**ip pim sparse-dense mode** 例： `Device(config-if)#ip pim sparse-dense mode`	VRF に関連付けられているレイヤ 3 インターフェイス上で、PIM をイネーブルにします。
ステップ 13	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 14	**show ip vrf** [**brief** | **detail** | **interfaces**] [*vrf-name*] 例： `Device#show ip vrf detail vpn1`	設定を確認します。設定した VRF に関する情報を表示します。
ステップ 15	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーションファイルに設定を保存します。

VPN ルーティング セッションの設定

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

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router ospf** *process-id* **vrf** *vrf-name* 例： `Device(config)#router ospf 1 vrf vpn1`	OSPF ルーティングをイネーブルにして VPN 転送テーブルを指定し、ルータ コンフィギュレーション モードを開始します。
ステップ 4	**log-adjacency-changes** 例： `Device(config-router)#log-adjacency-changes`	（任意）隣接ステートの変更を記録します。これは、デフォルトの状態です。
ステップ 5	**redistribute bgp** *autonomous-system-number* **subnets** 例： `Device(config-router)#redistribute bgp 10 subnets`	BGP ネットワークから OSPF ネットワークに情報を再配布するようにスイッチを設定します。
ステップ 6	**network** *network-number* **area** *area-id* 例： `Device(config-router)#network 1 area 2`	OSPF が動作するネットワーク アドレスとマスク、およびそのネットワーク アドレスのエリア ID を定義します。
ステップ 7	**end** 例： `Device(config-router)#end`	特権 EXEC モードに戻ります。
ステップ 8	**show ip ospf** *process-id* 例： `Device#show ip ospf 1`	OSPF ネットワークの設定を確認します。

	コマンドまたはアクション	目的
ステップ 9	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

BGP PE/CE ルーティング セッションの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**router bgp** *autonomous-system-number* 例： `Device(config)#router bgp 2`	その他の BGP ルータに AS 番号を渡す BGP ルーティング プロセスを設定し、ルータ コンフィギュレーション モードを開始します。
ステップ 3	**network** *network-number* **mask** *network-mask* 例： `Device(config-router)#network 5 mask 255.255.255.0`	BGP を使用してアナウンスするネットワークおよびマスクを指定します。
ステップ 4	**redistribute ospf** *process-id* **match internal** 例： `Device(config-router)#redistribute ospf 1 match internal`	OSPF 内部ルートを再配布するようにスイッチを設定します。
ステップ 5	**network** *network-number* **area** *area-id* 例： `Device(config-router)#network 5 area 2`	OSPF が動作するネットワーク アドレスとマスク、およびそのネットワーク アドレスのエリア ID を定義します。

	コマンドまたはアクション	目的
ステップ 6	**address-family ipv4 vrf** *vrf-name* 例： `Device(config-router)#address-family ipv4 vrf vpn1`	PE/CE ルーティング セッションの BGP パラメータを定義し、VRF アドレス ファミリ モードを開始します。
ステップ 7	**neighbor** *address* **remote-as** *as-number* 例： `Device(config-router)#neighbor 10.1.1.2 remote-as 2`	PE と CE ルータの間の BGP セッションを定義します。
ステップ 8	**neighbor** *address* **activate** 例： `Device(config-router)#neighbor 10.2.1.1 activate`	IPv4 アドレス ファミリのアドバタイズメントをアクティブ化します。
ステップ 9	**end** 例： `Device(config-router)#end`	特権 EXEC モードに戻ります。
ステップ 10	**show ip bgp** [**ipv4**] [**neighbors**] 例： `Device#show ip bgp ipv4 neighbors`	BGP 設定を確認します。
ステップ 11	**copy running-config startup-config** 例： `Device#copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

Multi-VRF CE のモニタリング

表 *47 : Multi-VRF CE* 情報を表示するコマンド

コマンド	目的
show ip protocols vrf *vrf-name*	VRF に対応付けられたルーティング プロトコル情報を表示します。
show ip route vrf *vrf-name* [**connected**] [*protocol* [*as-number*]] [**list**] [**mobile**] [**odr**] [**profile**] [**static**] [**summary**] [**supernets-only**]	VRF に対応付けられた IP ルーティング テーブル情報を表示します。

コマンド	目的
show ip vrf [**brief** | **detail** | **interfaces**] [*vrf-name*]	定義された VRF インスタンスに関する情報を表示します。

VRF 認識サービスの設定

次のサービスは、VRF 認識です。

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

ARP 用 VRF 認識サービスの設定

手順

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

ping 用 VRF 認識サービスの設定

手順

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

SNMP 用 VRF 認識サービスの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**snmp-server trap authentication vrf** 例： `Device(config)#snmp-server trap authentication vrf`	VRF で、パケットに対して SNMP トラップをイネーブルにします。
ステップ 4	**snmp-server engineID remote** *host* **vrf** *vpn-instance engine-id string* 例： `Device(config)#snmp-server engineID remote 172.16.20.3 vrf vpn1 80000009030000B064EFE100`	スイッチ上で、リモート SNMP エンジンの名前を設定します。
ステップ 5	**snmp-server host** *host* **vrf** *vpn-instance* **traps** *community* 例： `Device(config)#snmp-server host 172.16.20.3 vrf vpn1 traps comaccess`	SNMP トラップ動作の受信側、および SNMP トラップの送信に使用される VRF テーブルを指定します。
ステップ 6	**snmp-server host** *host* **vrf** *vpn-instance* **informs** *community* 例： `Device(config)#snmp-server host 172.16.20.3 vrf vpn1 informs comaccess`	SNMP 通知動作の受信先を指定し、SNMP 通知の送信に使用される VRF テーブルを指定します。

	コマンドまたはアクション	目的
ステップ 7	**snmp-server user** *user group* **remote** *host* **vrf** *vpn-instance security model* 例： `Device(config)#snmp-server user abcd remote 172.16.20.3 vrf vpn1 priv v2c 3des secure3des`	SNMP アクセス用に、VRF 上にあるリモート ホストの SNMP グループにユーザを追加します。
ステップ 8	**end** 例： `Device(config-if)#end`	特権 EXEC モードに戻ります。

NTP 用 VRF 認識サービスの設定

NTP 用の VRF 認識サービスの設定には、NTP サーバと、NTP サーバに接続された NTP クライアント インターフェイスの設定が含まれます。

始める前に

NTP クライアントとサーバの間の接続を確認します。NTP サーバに接続されているクライアント インターフェイスで有効な IP アドレスおよびサブネットを設定します。

NTP クライアントでの NTP 用 VRF 認識サービスの設定

NTP サーバに接続されているクライアント インターフェイスで次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • プロンプトが表示されたらパスワードを入力します。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例：	VRF に関連付けるレイヤ 3 インターフェイスを指定し、インターフェイス

	コマンドまたはアクション	目的
	`Device(config)#interface gigabitethernet 1/0/1`	コンフィギュレーションモードを開始します。
ステップ 4	**vrf forwarding** *vrf-name* 例： `Device(config-if)#vrf forwarding A`	VRF をレイヤ 3 インターフェイスに対応付けます。
ステップ 5	**ip address** *ip-address subnet-mask* 例： `Device(config-if)#ip address 1.1.1.1 255.255.255.0`	インターフェイスの IP アドレスを入力します。
ステップ 6	**no shutdown** 例： `Device(config-if)#no shutdown`	インターフェイスをイネーブルにします。
ステップ 7	**exit** 例： `Device(config-if)exit`	インターフェイス コンフィギュレーション モードを終了します。
ステップ 8	**ntp authentication-key** *number* **md5** *md5-number* 例： `Device(config)#ntp authentication-key 1 md5 cisco123`	認証キーを定義します。デバイスが時刻源と同期するのは、時刻源がこれらの認証キーのいずれかを持ち、 **ntp trusted-key number** コマンドによってキー番号が指定されている場合だけです。 （注） 認証キー番号と MD5 パスワードは、クライアントとサーバの両方で同じである必要があります。
ステップ 9	**ntp authenticate** 例： `Device(config)#ntp authenticate`	NTP 認証機能をイネーブルにします。NTP 認証はデフォルトでディセーブルになっています。
ステップ 10	**ntp trusted-key** *key-number* 例： `Device(config)#ntp trusted-key 1`	NTP クライアントで同期をとれるようにするために、NTP サーバによってその NTP パケットで提供される必要がある 1 つ以上のキーを指定します。trusted key の範囲は 1 ～ 65535 です。このコマンドにより、NTP クライアントが、信頼されていない NTP サーバと誤って

	コマンドまたはアクション	目的
		同期する、ということが防止されます。
ステップ 11	**ntp server vrf** *vrf-name* 例： `Device(config)#ntp server vrf A 1.1.1.2 key 1`	指定された VRF で NTP サーバを設定します。

NTP サーバでの NTP 用 VRF 認識サービスの設定

NTP サーバで次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ntp authentication-key** *number* **md5** *passowrd* 例： `Device(config)#ntp authentication-key 1 md5 cisco123`	認証キーを定義します。デバイスが時刻源と同期するのは、時刻源がこれらの認証キーのいずれかを持ち、 **ntp trusted-key number** コマンドによってキー番号が指定されている場合だけです。 （注） 認証キー番号と MD5 パスワードは、クライアントとサーバの両方で同じである必要があります。
ステップ 4	**ntp authenticate** 例： `Device(config)#ntp authenticate`	NTP 認証機能をイネーブルにします。NTP 認証はデフォルトでディセーブルになっています。

	コマンドまたはアクション	目的
ステップ 5	**ntp trusted-key** *key-number* 例： `Device(config)#ntp trusted-key 1`	NTP クライアントで同期をとれるようにするために、NTP サーバによってその NTP パケットで提供される必要がある 1 つ以上のキーを指定します。trusted key の範囲は 1 ～ 65535 です。このコマンドにより、NTP クライアントが、信頼されていない NTP サーバと誤って同期する、ということが防止されます。
ステップ 6	**interface** *interface-id* 例： `Device(config)#interface gigabitethernet 1/0/3`	VRF に関連付けるレイヤ 3 インターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 7	**vrf forwarding** *vrf-name* 例： `Device(config-if)#vrf forwarding A`	VRF をレイヤ 3 インターフェイスに対応付けます。
ステップ 8	**ip address** *ip-address subnet-mask* 例： `Device(config-if)#ip address 1.1.1.2 255.255.255.0`	インターフェイスの IP アドレスを入力します。
ステップ 9	**exit** 例： `Device(config-if)exit`	インターフェイス コンフィギュレーション モードを終了します。

uRPF 用 VRF 認識サービスの設定

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

手順

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

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

VRF 認識 RADIUS の設定

VRF 認識 RADIUS を設定するには、まず RADIUS サーバ上で AAA をイネーブルにする必要があります。『*Per VRF AAA Feature Guide*』で説明されているとおり、スイッチで **ip vrf forwarding** *vrf-name* サーバグループ コンフィギュレーション コマンドと **ip radius source-interface** グローバル コンフィギュレーション コマンドがサポートされます。

syslog 用 VRF 認識サービスの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device>enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**logging on** 例： `Device(config)#logging on`	ストレージルータ イベント メッセージのロギングを、イネーブルまたは一時的にディセーブルにします。
ステップ 4	**logging host** *ip-address* **vrf** *vrf-name* 例： `Device(config)#logging host 10.10.1.0 vrf vpn1`	ロギング メッセージが送信される Syslog サーバのホストアドレスを指定します。
ステップ 5	**logging buffered** *logging buffered size* **debugging** 例： `Device(config)#logging buffered critical 6000 debugging`	メッセージを内部バッファにロギングします。
ステップ 6	**logging trap debugging** 例： `Device(config)#logging trap debugging`	Syslog サーバに送信されるロギング メッセージを制限します。
ステップ 7	**logging facility** *facility* 例： `Device(config)#logging facility user`	ロギング ファシリティにシステム ロギング メッセージを送信します。
ステップ 8	**end** 例：	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
	`Device(config-if)#end`	

traceroute 用 VRF 認識サービスの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**traceroute vrf** *vrf-name ipaddress* 例： `Device(config)#traceroute vrf vpn2 10.10.1.1`	宛先アドレスを取得する VPN VRF の名前を指定します。

FTP および TFTP 用 VRF 認識サービスの設定

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

手順

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

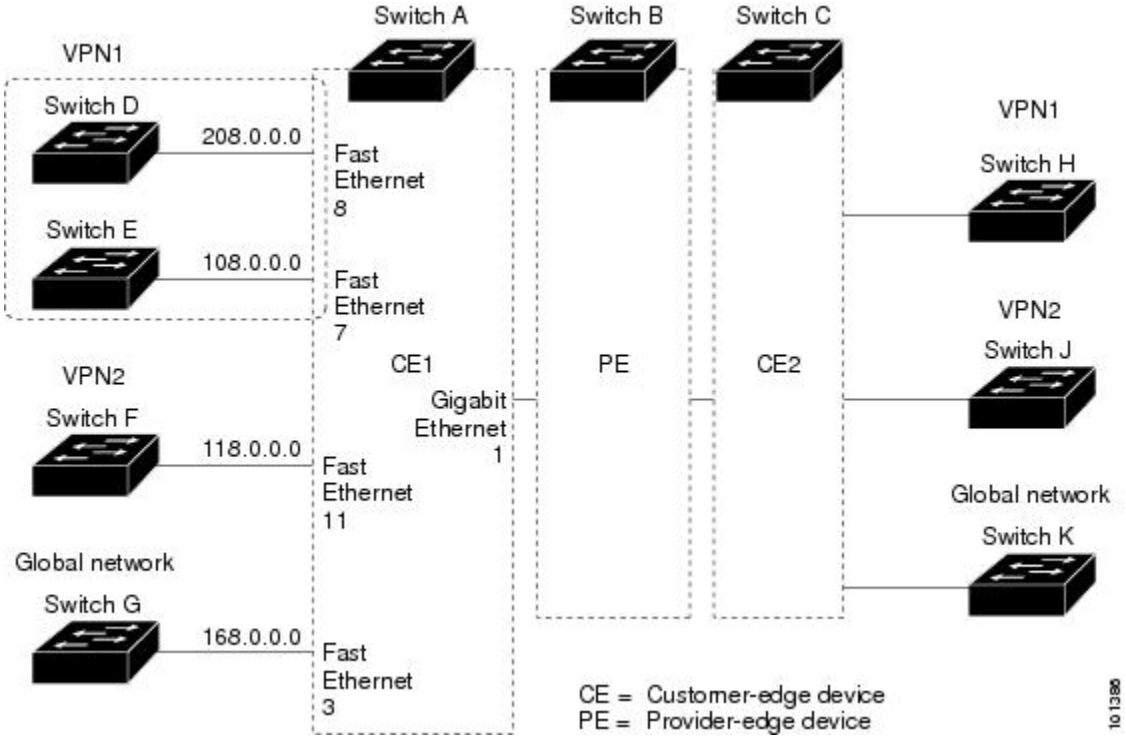
	コマンドまたはアクション	目的
ステップ 3	**ip ftp source-interface** *interface-type interface-number* 例： `Device(config)#ip ftp source-interface gigabitethernet 1/0/2`	FTP 接続の発信元 IP アドレスを指定します。
ステップ 4	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 5	**configure terminal** 例： `Device#configure terminal`	グローバル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

Multi-VRF CE の設定例

Multi-VRF CE の設定例

VPN1、VPN2、およびグローバル ネットワークで使用されるプロトコルは OSPF です。CE/PE 接続には BGP が使用されます。図のあとに続く出力は、スイッチを CE スイッチ A として設定する例、およびカスタマー スイッチ D と F の VRF 設定を示しています。CE スイッチ C とその他のカスタマー スイッチを設定するコマンドは含まれていませんが、内容は同様です。

図 *17 : Multi-VRF CE* の設定例



スイッチ A では、ルーティングをイネーブルにして VRF を設定します。

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

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

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

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

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

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

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

VPN1 と VPN2 で OSPF ルーティングを設定します。

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

CE/PE ルーティングに BGP を設定します。

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

```
Device(config-router-af)#network 8.8.1.0 mask 255.255.255.0
Device(config-router-af)#end
```

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

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

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

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

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

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

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

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

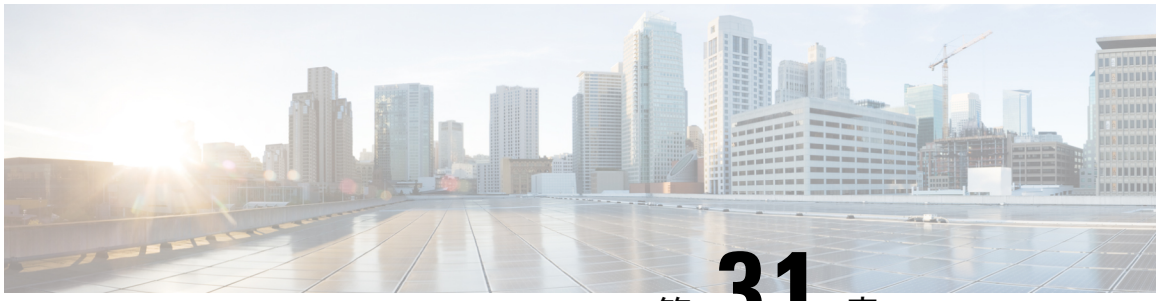
```
Device#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)#ip vrf v1
Device(config-vrf)#rd 100:1
Device(config-vrf)#route-target export 100:1
Device(config-vrf)#route-target import 100:1
Device(config-vrf)#exit

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

Device(config)#interface Loopback2
```

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

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

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

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

マルチ VRF CE の機能情報

表 *48* : マルチ *VRF CE* の機能情報

機能名	リリース	機能情報
マルチ VRF CE	Cisco IOS XE Everest 16.6.1	この機能が導入されました



第 31 章

ユニキャスト リバース パス転送の設定

- ユニキャスト リバース パス転送の設定 （455 ページ）
- IPv6 ユニキャスト リバース パス転送の設定 （455 ページ）

ユニキャスト リバース パス転送の設定

ユニキャスト リバース パス転送（ユニキャスト RPF）機能は、検証可能な送信元 IP アドレスが不足している IP パケットを廃棄することで、間違ったまたは偽造（スプーフィングされた）送信元 IP アドレスがネットワークに流れて発生する問題を軽減するのに役立ちます。たとえば、Smurf や Tribal Flood Network（TFN）など、多くの一般的なタイプの DoS 攻撃は、偽造された、または次々に変わる送信元 IP アドレスを使用して、攻撃を突き止めたりフィルタすることを攻撃者が阻止できるようにします。パブリック アクセスを提供するインターネット サービス プロバイダー（ISP）の場合、uRPF が IP ルーティング テーブルと整合性の取れた有効な送信元アドレスを持つパケットだけを転送することによって、そのような攻撃をそらします。この処理により、ISP のネットワーク、その顧客、および残りのインターネットが保護されます。

（注）

- ユニキャスト RPF は、 でサポートされています。

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

IPv6 ユニキャスト リバース パス転送の設定

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

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

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

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