Cisco Nexus 3600 NX-OS ラベル スイッチング構成ガイド、リリース 10.3（x）

初版：2022 年 8 月 19 日

最終更新：2023 年 1 月 5 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 REFERENCED IN THIS DOCUMENTATION ARE SUBJECT TO CHANGE WITHOUT NOTICE. EXCEPT AS MAY OTHERWISE BE AGREED BY CISCO IN WRITING, ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENTATION ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

The Cisco End User License Agreement and any supplemental license terms govern your use of any Cisco software, including this product documentation, and are located at: http://www.cisco.com/go/softwareterms.Cisco product warranty information is available at http://www.cisco.com/go/warranty. US Federal Communications Commission Notices are found here http://www.cisco.com/c/en/us/products/us-fcc-notice.html.

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 products and features described herein as in development or available at a future date remain in varying stages of development and will be offered on a when-and if-available basis. Any such product or feature roadmaps are subject to change at the sole discretion of Cisco and Cisco will have no liability for delay in the delivery or failure to deliver any products or feature roadmap items that may be set forth in this document.

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.

The documentation set for this product strives to use bias-free language. For the purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

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: www.cisco.com go trademarks. 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)

© 2022 Cisco Systems, Inc. All rights reserved.

目 次

はじめに

この前書きは、次の項で構成されています。

対象読者

このマニュアルは、Cisco Nexus スイッチの設置、設定、および維持に携わるネットワーク管理者を対象としています。

表記法

コマンドの説明には、次のような表記法が使用されます。

表記法	説明
bold	太字の文字は、表示どおりにユーザが入力するコマンドおよびキーワードです。
italic	イタリック体の文字は、ユーザが値を入力する引数です。
[x]	省略可能な要素（キーワードまたは引数）は、角かっこで囲んで示しています。
[x | y]	いずれか1つを選択できる省略可能なキーワードや引数は、角カッコで囲み、縦棒で区切って示しています。
{x | y}	必ずいずれか1つを選択しなければならない必須キーワードや引数は、波かっこで囲み、縦棒で区切って示しています。

表記法	説明
[x {y | z}]	角かっこまたは波かっこが入れ子になっている箇所は、任意または必須の要素内の任意または必須の選択肢であることを表します。角かっこ内の波かっこと縦棒は、省略可能な要素内で選択すべき必須の要素を示しています。
`variable`	ユーザが値を入力する変数であることを表します。イタリック体が使用できない場合に使用されます。
string	引用符を付けない一組の文字。string の前後には引用符を使用しません。引用符を使用すると、その引用符も含めて string とみなされます。

例では、次の表記法を使用しています。

表記法	説明
`screen フォント`	スイッチが表示する端末セッションおよび情報は、スクリーンフォントで示しています。
太字の screen フォント	ユーザが入力しなければならない情報は、太字のスクリーンフォントで示しています。
イタリック体の *screen* フォント	ユーザが値を指定する引数は、イタリック体の screen フォントで示しています。
<>	パスワードのように出力されない文字は、山カッコ（<>）で囲んで示しています。
[ ]	システム プロンプトに対するデフォルトの応答は、角カッコで囲んで示しています。
!、#	コードの先頭に感嘆符（!）またはポンド記号（#）がある場合には、コメント行であることを示します。

Cisco Nexus 3000 シリーズ スイッチの関連資料

Cisco Nexus 3000 シリーズ スイッチ全体のマニュアル セットは、次の URL にあります。

https://www.cisco.com/c/en/us/support/switches/nexus-3000-series-switches/tsd-products-support-series-home.html

マニュアルに関するフィードバック

このマニュアルに関する技術的なフィードバック、または誤りや記載もれなどお気づきの点がございましたら、HTML ドキュメント内のフィードバック フォームよりご連絡ください。ご協力をよろしくお願いいたします。

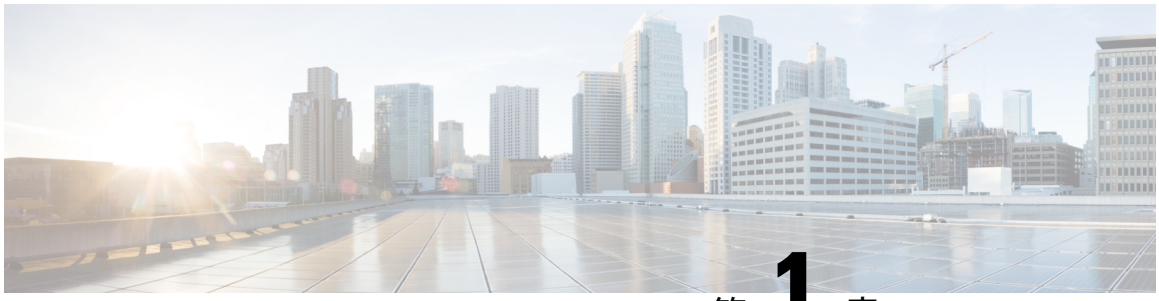
通信、サービス、およびその他の情報

- シスコからタイムリーな関連情報を受け取るには、Cisco Profile Manager でサインアップしてください。
- 重要な技術によりビジネスに必要な影響を与えるには、シスコ サービスにアクセスしてください。
- サービス リクエストを送信するには、シスコ サポートにアクセスしてください。
- 安全で検証済みのエンタープライズクラスのアプリケーション、製品、ソリューション、およびサービスを探して参照するには、Cisco Marketplace にアクセスしてください。
- 一般的なネットワーク、トレーニング、認定関連の出版物を入手するには、Cisco Press にアクセスしてください。
- 特定の製品または製品ファミリの保証情報を探すには、Cisco Warranty Finder にアクセスしてください。

Cisco Bug Search Tool

Cisco バグ検索ツール（BST）は、シスコ製品とソフトウェアの障害と脆弱性の包括的なリストを管理する Cisco バグ追跡システムへのゲートウェイとして機能する、Web ベースのツールです。BST は、製品とソフトウェアに関する詳細な障害情報を提供します。

通信、サービス、およびその他の情報



第 1 章

新規および変更情報

この章では、*[Cisco Nexus 3600* シリーズ *NX-OS* ラベル スイッチング構成ガイド、リリース *10.3*（*x*）（*Cisco Nexus 3600 Series NX-OS Label Switching Configuration Guide, Release 10.3(x)*）*]* に記載されている新機能および変更された機能に関するリリース固有の情報について説明します。

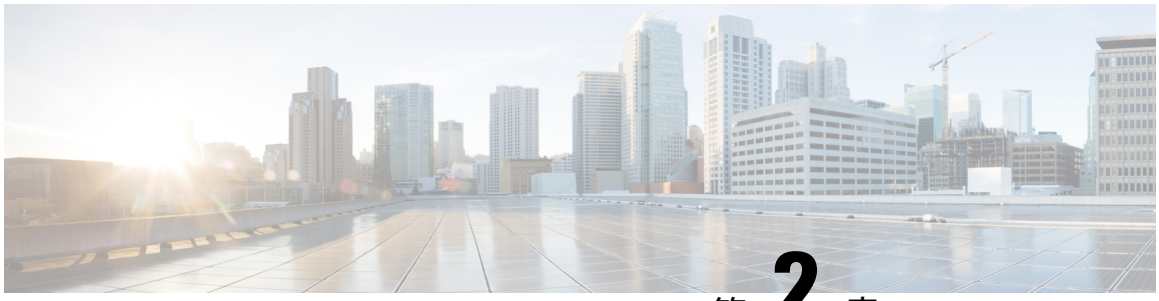
新規および変更情報

次の表は、*[Cisco Nexus 3600* シリーズ *NX-OS* ラベル スイッチング構成ガイド、リリース *10.3*（*x*）*]* に記載されている新機能と変更機能を要約したものです。それぞれの説明が記載されている箇所も併記されています。

表 *1 : NX-OS* リリース *10.3(x)* の新機能および変更された機能

特長	説明	変更が行われたリリース	参照先
NA	このリリースで追加された新機能はありません。	10.3(1)F	該当なし

新規および変更情報



第 2 章

セグメント ルーティングの設定

この章では、セグメント ルーティングの設定方法について説明します。

ライセンス要件

Cisco NX-OS ライセンス方式の推奨の詳細と、ライセンスの取得および適用の方法については、『*Cisco NX-OS Licensing Guide*』を参照してください。

セグメント ルーティングについて

セグメント ルーティングは、ソース ルーティングと同様に、パケットがたどるパスをパケット自体にエンコードする手法です。ノードは、制御された一連の命令（セグメント）によってパケットをステアリングするために、パケットの前にセグメント ルーティング ヘッダーを付加する各セグメントを識別するセグメント ID（SID）は、フラットな 32 ビットの符号なし整数からなる

セグメントのサブクラスであるボーダーゲートウェイプロトコル（BGP）セグメントは、BGP 転送命令を識別します。BGP セグメントには、プレフィックス セグメントと隣接セグメントの 2 つのグループがあります。プレフィックス セグメントは、利用可能なすべての等コスト マルチパス（ECMP）パスを使用して、宛先への最短パスを通るようパケットを誘導します。

隣接セグメントは、パケットをネイバーへの特定のリンクに誘導します。

セグメント ルーティング アーキテクチャは、MPLS データ プレーンに直接適用される

セグメント ルーティング アプリケーション モジュール

セグメントルーティングアプリケーション（SR-APP）モジュールは、セグメントルーティング機能を構成するために使用されます。セグメントルーティングアプリケーション（SR-APP）は、セグメントルーティングに関連するすべてのCLIを処理する独立した内部プロセスです。SRGB範囲を予約し、それについてクライアントに通知する役割を担います。また、プレフィックスから SID へのマッピングの維持も担当します。SR-APP サポートは、BGP、IS-IS、および OSPF プロトコルでも利用できます。

SR-APP モジュールは、以下の情報を保持します。

- セグメント ルーティングの動作状態
- セグメント ルーティングのグローバル ブロック範囲
- プレフィックス SID マッピング

詳細については、セグメント ルーティングの設定 （6 ページ）を参照してください。

セグメント ルーティングの注意事項と制限事項

セグメント ルーティングに関する注意事項および制約事項は、次のとおりです。

- MPLSセグメントルーティングは、物理イーサネットインターフェイスおよびポートチャネル バンドルで有効にできます。イーサネット サブインターフェイスまたは Switchedx Virtual Interfaces (SVI) ではサポートされていません。
- BGP は、next-hop-selfが有効な場合にのみ、iBGPルートリフレクタクライアントにSRGBラベルを割り当てます（たとえば、プレフィックスは、RR 上のローカル IP / IPv6 アドレスの 1 つであるネクスト ホップでアドバタイズされます）。RR で next-hop-self を構成すると、影響を受けるルートのネクスト ホップが変更されます（ルート マップ フィルタ処理の対象）。
- スタティック MPLS、MPLS セグメント ルーティング、および MPLS ストリッピングを同時に有効にすることはできません。
- スタティック MPLS、MPLS セグメント ルーティング、および MPLS ストリッピングは相互に排他的であるため、マルチホップ BGP の唯一のセグメント ルーティング アンダーレイはシングルホップ BGP です。 eBGP をオーバーレイとして実行する iBGP マルチホップトポロジはサポートされていません。
- 特定のインターフェイスへの転送がその後に続く MPLS ポップはサポートされていません。最後から 2 番目のホップ ポップ（PHP）は、コントロール プレーンが IPv4 黙示的 NULL ラベルをインストールした場合でも、ラベル FIB (LFIB) の out-label として明示的 NULL ラベルをインストールすれば回避できます。

- BGP ラベル付きユニキャストおよびBGP セグメント ルーティングは、IPv6 プレフィックスではサポートされていません。
- BGP ラベル付きユニキャストおよびBGP セグメント ルーティングは、トンネル インターフェイス（GREおよびVXLANを含む）またはvPCアクセスインターフェイスではサポートされていません。
- MTU パス ディスカバリ（RFC 2923）は、MPLS ラベル スイッチド パス（LSP）またはセグメント ルーテッド パスではサポートされていません。
- BGP 設定コマンドの **neighbor-down fib-accelerate** および **suppress-fib-pending** は、MPLS プレフィックスではサポートされていません。
- セグメント ルーティング グローバル ブロック（SRGB）を再構成すると、BGP プロセスが自動的に再起動され、既存の URIB および ULIB エントリが更新されます。トラフィックの損失は数秒間発生するため、本番環境で SRGB を再構成しないでください。
- セグメントルーティング グローバルブロック（SRGB）が範囲に設定されているが、ルート マップ ラベル インデックス デルタ値が構成された範囲外にある場合、割り当てられたラベルはダイナミックに生成されます。たとえば、ルート マップのラベル インデックスが 9000 に設定されているときに SRGB が 16000 ～ 23999 の範囲に設定されている場合、ラベルはダイナミックに割り当てられます。
- ネットワークの拡張性のため、トップオブラック（TOR）または境界リーフ スイッチから接続されているプレフィクスをアドバタイズするマルチホップ BGP とともに階層型ルーティング設計を使用することを推奨します。
- BGP セッションは、MPLS LSP またはセグメント ルーテッド パスではサポートされていません。
- レイヤ 3 転送整合性チェッカーは、MPLS ルートではサポートされていません。
- セグメント ルーティング構成を削除すると、関連するすべてのセグメント ルーティング構成が削除されます。
- セグメント ルーティング上のレイヤ 3 VPN は、N3K-C3636C-R および N3K-C36180YC-R ラインカードを備えた Cisco Nexus 3600 プラットフォーム スイッチでサポートされています。
- ブート変数を設定してスイッチをリロードすることによって、Cisco Nexus デバイスを Cisco NX-OS リリース 9.3(1) から以前の NX-OS リリースにダウングレードすると、セグメント ルーティング mpls の以前のすべての構成が失われます。
- Cisco NX-OS リリース 9.3(1) から ISSD を実行する前に、セグメント ルーティング設定を無効にする必要があります。そうしないと、既存のセグメントルーティング構成が失われます。

セグメント ルーティングの設定

セグメント ルーティングの設定

始める前に

セグメントルーティングを設定する前に、以下の条件を満たしていることを確認してください。

- **segment-routing** コマンドを構成する前に、**install feature-set mpls**、**feature-set mpls** および **feature mpls segment-routing** コマンドが存在している必要があります。
- グローバルブロックが構成されている場合、指定された範囲が使用されます。それ以外の場合は、デフォルトの 16000 ～ 23999 の範囲が使用されます。
- BGP は、**set label-index**<*value*> 構成と新しい**connected-prefix-sid-map** CLI の両方を使用するようになりました。競合が発生した場合は、SR-APP の構成が優先されます。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル 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	**segment-routing**
	例：
	`switch(config)# segment-routing`
	`switch(config-sr)# mpls`
	`switch(config-sr-mpls)#`
		MPLS セグメント ルーティング機能を有効にします。このコマンドの **no** 形式は、MPLS セグメント ルーティング機能を無効化します。

ステップ 3	**connected-prefix-sid-map**
	例：
	`switch(config-sr-mpls)#`
	`connected-prefix-sid-map`
	`switch(config-sr-mpls)#`
		接続されたプレフィックス セグメント ID マッピングを設定します。

ステップ 4	**global-block** <*min*> <*max*>
	例：
	`switch(config-sr-mpls)# global-block`
	`<min> <max>`
	`switch(config-sr-mpls)#`
		セグメント ルーティング バインディングのグローバル ブロック範囲を指定します。

	コマンドまたはアクション	目的

ステップ 5	**connected-prefix-sid-map**
	例：
	`switch(config-sr-mpls)# connected-prefix-sid-map`
	`switch(config-sr-mpls-conn-pfsid)#`
		接続されたプレフィックス セグメント ID マッピングを設定します。

ステップ 6	**address-family ipv4** 例： `switch(config-sr-mpls-conn-pfsid)#address-family ipv4`	IPv4 アドレス ファミリを設定します。
ステップ 7	*<prefix>/<masklen>* [**index**|**absolute**] *<label>* 例： `switch(config-sr-mpls)# 2.1.1.5/32 absolute 201101`	オプションのキーワード **index** または **absolute** は、入力されたラベル値を SRGB へのインデックスとして解釈するか、絶対値として解釈するかを示します。

例

showコマンドについては、次の設定例を参照してください。

```
switch# show segment-routing mpls
Segment-Routing Global info

Service Name: segment-routing

State: Enabled

Process Id: 29123

Configured SRGB: 17000 - 24999

SRGB Allocation status: Alloc-Successful

Current SRGB: 17000 - 24999

Cleanup Interval: 60

Retry Interval: 180
```

次の CLI は、SR-APP に登録されているクライアントを表示します。クライアントが関心を登録した VRF がリストされます。

```
switch# show segment-routing mpls clients
            Segment-Routing Mpls Client Info

Client: isis-1
    PIB index: 1    UUID: 0x41000118    PID: 29463    MTS SAP: 412
    TIBs registered:
        VRF: default Table: base
```

```
Client: bgp-1
    PIB index: 2    UUID: 0x11b    PID: 18546    MTS SAP: 62252
    TIBs registered:
        VRF: default Table: base

Total Clients: 2
```

show segment-routing mpls ipv4 connected-prefix-sid-map CLI コマンドの例では、SRGB は、プレフィックス SID が構成された SRGB 内にあるかどうかを示します。**Indx** フィールドは、構成されたラベルがグローバルブロックへのインデックスであることを示します。**Abs** フィールドは、構成されたラベルが絶対値であることを示します。

SRGB フィールドに N が表示されている場合は、構成されたプレフィックス SID が SRGB 範囲内になく、SR-APP クライアントに提供されていないことを意味します。SRGB 範囲に入るプレフィックス SID のみが SR-APP クライアントに与えられます。

```
switch# show segment-routing mpls ipv4 connected-prefix-sid-map
            Segment-Routing Prefix-SID Mappings
Prefix-SID mappings for VRF default Table base
Prefix              SID   Type Range SRGB
13.11.2.0/24        713   Indx 1     Y
30.7.7.7/32         730   Indx 1     Y
59.3.24.0/30        759   Indx 1     Y
150.101.1.0/24      801   Indx 1     Y
150.101.1.1/32      802   Indx 1     Y
150.101.2.0/24      803   Indx 1     Y
1.1.1.1/32          16013 Abs  1     Y
```

次の CLI は **show running-config segment-routing** 出力を表示します。

```
switch# show running-config segment-routing ?

> Redirect it to a file
>> Redirect it to a file in append mode
all Show running config with defaults
| Pipe command output to filter

switch# show running-config segment-routing
switch# show running-config segment-routing

!Command: show running-config segment-routing
!Running configuration last done at: Thu Dec 12 19:39:52 2019
!Time: Thu Dec 12 20:06:07 2019

version 9.3(3) Bios:version 05.39
segment-routing
    mpls
        connected-prefix-sid-map
            address-family ipv4
                2.1.1.1/32 absolute 100100

switch#
```

インターフェイス上の MPLS のイネーブル化

MPLS はセグメント ルーティングで使用するインターフェイスで有効にすることができます。

始める前に

MPLS 機能セットは、**install feature-set mpls** および **feature-set mpls** コマンドを使用してインストールし、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	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル 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 slot/port*
	例：
	`switch(config)# interface ethernet 2/2`
	`switch(config-if)#`
		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 3	[**no**] **mpls ip forwarding** 例： `switch(config-if)# mpls ip forwarding`	指定されたインターフェイスで MPLS を有効にします。このコマンドの **no** 形式は、指定されたインターフェイスで MPLS を無効にします。
ステップ 4	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	実行設定を、スタートアップ設定にコピーします。

セグメント ルーティング グローバル ブロックの設定

セグメント ルーティング グローバル ブロック（SRGB）の開始と終了 MPLS ラベルは設定できます。

始める前に

MPLS 機能セットは、**install feature-set mpls** および **feature-set mpls** コマンドを使用してインストールし、有効にする必要があります。

MPLS セグメント ルーティング機能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	**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
	`switch# configure terminal` `switch(config)#`	

ステップ 2	[**no**] **segment-routing**
	例：
	`switch(config)# segment-routing`
	`switch(config-sr)# mpls`
		セグメント ルーティング コンフィギュレーション モードを開始し、16000 ～ 23999 のデフォルトの SRGB を有効にします。このコマンドの **no** 形式は、そのラベル ブロックの割り当てを解除します。
		設定されたダイナミック レンジがデフォルトの SRGB を保持できない場合、エラー メッセージが表示され、デフォルトの SRGB は割り当てられません。必要に応じて、次の手順で別の SRGB を設定できます。

ステップ 3	[**no**] **global-block** *beginning-label ending-label* 例： `switch(config-sr-mpls)# global-block 16000 471804`	SRGB の MPLS ラベル範囲を指定します。このコマンドは、**segment-routing** コマンドで設定されたデフォルトの SRGB ラベル範囲を変更する場合に使用します。 開始 MPLS ラベルと終了 MPLS ラベルの許容値は 16000 ～ 471804 です。**mpls label range** コマンドでは最小ラベルとして 16 が許可されますが、SRGB は 16000 からしか開始できません。 （注） **global-block** コマンドの最小値は 16000 から始まります。以前のリリースからアップグレードする場合は、アップグレードをトリガーする前に、サポートされている範囲内に収まるように SRGB を変更する必要があります。
ステップ 4	（任意） **show mpls label range** 例： `switch(config-sr-mpls)# show mpls label range`	SRGB の割り当てが成功した場合にのみ、SRGB を表示します。
ステップ 5	**show segment-routing**	設定されている SRGB を表示します。

	コマンドまたはアクション	目的
ステップ 6	**show segment-routing mpls** 例： `switch(config-sr-mpls)# show segment-routing mpls`	設定されている SRGB を表示します。
ステップ 7	（任意） **copy running-config startup-config** 例： `switch(config-sr-mpls)# copy running-config startup-config`	実行設定を、スタートアップ設定にコピーします。

セグメント ルーティングの構成例

このセクションの例は、2 台のルータ間の一般的な BGP プレフィックス SID 構成を示しています。

この例は、10.10.10.10/32 と 20.20.20.20/32 の BGP スピーカー構成を、それぞれ 10 と 20 のラベル インデックスでアドバタイズする方法を示しています。16000 ～ 23999 のデフォルトのセグメント ルーティング グローバル ブロック （SRGB）範囲を使用します。

```
hostname s1
install feature-set mpls
feature-set mpls

feature telnet
feature bash-shell
feature scp-server
feature bgp
feature mpls segment-routing

segment-routing
  mpls
  vlan 1
segment-routing
  mpls
    connected-prefix-sid-map
    address-family ipv4
    2.1.1.1/32 absolute 100100

route-map label-index-10 permit 10
  set label-index 10
route-map label-index-20 permit 10
  set label-index 20

vrf context management
  ip route 0.0.0.0/0 10.30.108.1

interface Ethernet1/1
  no switchport
  ip address 10.1.1.1/24
  no shutdown

interface mgmt0
  ip address dhcp
```

```
  vrf member management

interface loopback1
  ip address 10.10.10.10/32

interface loopback2
  ip address 20.20.20.20/32

line console
line vty

router bgp 1
  address-family ipv4 unicast
    network 10.10.10.10/32 route-map label-index-10
    network 20.20.20.20/32 route-map label-index-20
    allocate-label all
  neighbor 10.1.1.2 remote-as 2
    address-family ipv4 labeled-unicast
```

この例は、BGP スピーカーからの構成を受信する方法を示しています。

```
hostname s2
install feature-set mpls
feature-set mpls

feature telnet
feature bash-shell
feature scp-server
feature bgp
feature mpls segment-routing

segment-routing mpls
vlan 1

vrf context management
  ip route 0.0.0.0/0 10.30.97.1
  ip route 0.0.0.0/0 10.30.108.1

interface Ethernet1/1
  no switchport
  ip address 10.1.1.2/24
  ipv6 address 10:1:1::2/64
  no shutdown

interface mgmt0
  ip address dhcp
  vrf member management

interface loopback1
  ip address 2.2.2.2/32
line console

line vty

router bgp 2
  address-family ipv4 unicast
    allocate-label all
  neighbor 10.1.1.1 remote-as 1
    address-family ipv4 labeled-unicast
```

この例は、BGP スピーカーからの構成を表示する方法を示しています。この例の **show** コマンドは、16000 ～ 23999 の SRGB 範囲のラベル 16010 にマッピングされているラベル インデックス 10 のプレフィックス 10.10.10.10 を表示します。

```
switch# show bgp ipv4 labeled-unicast 10.10.10.10/32

BGP routing table information for VRF default, address family IPv4 Label Unicast
BGP routing table entry for 10.10.10.10/32, version 7
Paths: (1 available, best #1)
Flags: (0x20c001a) on xmit-list, is in urib, is best urib route, is in HW, , has label
  label af: version 8, (0x100002) on xmit-list
  local label: 16010

  Advertised path-id 1, Label AF advertised path-id 1
  Path type: external, path is valid, is best path, no labeled nexthop, in rib
  AS-Path: 1 , path sourced external to AS
    10.1.1.1 (metric 0) from 10.1.1.1 (10.10.10.10)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 0
      Prefix-SID Attribute: Length: 10
        Label Index TLV: Length 7, Flags 0x0 Label Index 10

  Path-id 1 not advertised to any peer
  Label AF advertisement
  Path-id 1 not advertised to any peer
```

この例は、BGP スピーカーで出力ピア エンジニアリングを構成する方法を示しています。

```
hostname epe-as-1
install feature-set mpls
feature-set mpls

feature telnet
feature bash-shell
feature scp-server
feature bgp
feature mpls segment-routing

segment-routing mpls
vlan 1

vrf context management
  ip route 0.0.0.0/0 10.30.97.1
  ip route 0.0.0.0/0 10.30.108.1

interface Ethernet1/1
  no switchport
  ip address 10.1.1.1/24
  no shutdown

interface Ethernet1/2
  no switchport
  ip address 11.1.1.1/24
  no shutdown

interface Ethernet1/3
  no switchport
  ip address 12.1.1.1/24
  no shutdown

interface Ethernet1/4
  no switchport
  ip address 13.1.1.1/24
```

```
  no shutdown

interface Ethernet1/5
  no switchport
  ip address 14.1.1.1/24
  no shutdown
```

次に、show ip route vrf 2 コマンドの例を示します。

```
show ip route vrf 2
IP Route Table for VRF "2"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

41.11.2.0/24, ubest/mbest: 1/0
    *via 1.1.1.9%default, [20/0], 13:26:48, bgp-2, external, tag 11 (mpls-vpn)
42.11.2.0/24, ubest/mbest: 1/0, attached
    *via 42.11.2.1, Vlan2, [0/0], 13:40:52, direct
42.11.2.1/32, ubest/mbest: 1/0, attached
    *via 42.11.2.1, Vlan2, [0/0], 13:40:52, local
```

次に、**show forwarding route vrf 2** コマンドの例を示します。

```
slot  1
=======

IPv4 routes for table 2/base

------------------+----------------------------------+-----------------+--------------+--------------
Prefix            | Next-hop                         | Interface       |
Labels          | Partial Install
------------------+----------------------------------+-----------------+--------------+--------------
0.0.0.0/32          Drop                               Null0
127.0.0.0/8         Drop                               Null0
255.255.255.255/32  Receive                            sup-eth1
*41.11.2.0/24       27.1.31.4                          Ethernet1/3
PUSH  30002 492529
                    27.1.32.4                          Ethernet1/21
PUSH  30002 492529
                    27.1.33.4                          port-channel23
PUSH  30002 492529
                    27.11.31.4                         Ethernet1/3.11
PUSH  30002 492529
                    27.11.33.4                         port-channel23.11
PUSH  30002 492529
                    37.1.53.4                          Ethernet1/53/1
PUSH  29002 492529
                    37.1.54.4                          Ethernet1/54/1
PUSH  29002 492529
                    37.2.53.4                          Ethernet1/53/2
PUSH  29002 492529
                    37.2.54.4                          Ethernet1/54/2
PUSH  29002 492529
                    80.211.11.1                        Vlan801
PUSH  30002 492529
```

次に、**show bgp l2vpn evpn summary** コマンドの例を示します。

```
show bgp l2vpn evpn summary
BGP summary information for VRF default, address family L2VPN EVPN
BGP router identifier 2.2.2.3, local AS number 2
BGP table version is 17370542, L2VPN EVPN config peers 4, capable peers 1
1428 network entries and 1428 paths using 268464 bytes of memory
BGP attribute entries [476/76160], BGP AS path entries [1/6]
BGP community entries [0/0], BGP clusterlist entries [0/0]
476 received paths for inbound soft reconfiguration
476 identical, 0 modified, 0 filtered received paths using 0 bytes

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
1.1.1.1         4    11       0       0        0    0    0 23:01:53 Shut (Admin)
1.1.1.9         4    11    4637    1836 17370542    0    0 23:01:40 476
1.1.1.10        4    11       0       0        0    0    0 23:01:53 Shut (Admin)
1.1.1.11        4    11       0       0        0    0    0 23:01:52 Shut (Admin)
```

次に、**show bgp l2vpn evpn** コマンドの例を示します。

```
show bgp l2vpn evpn 41.11.2.0
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 14.1.4.1:115
BGP routing table entry for [5]:[0]:[0]:[24]:[41.11.2.0]:[0.0.0.0]/224, version 17369591
Paths: (1 available, best #1)
Flags: (0x000002) on xmit-list, is not in l2rib/evpn, is not in HW

  Advertised path-id 1
  Path type: external, path is valid, received and used, is best path
             Imported to 2 destination(s)
  AS-Path: 11 , path sourced external to AS
    1.1.1.9 (metric 0) from 1.1.1.9 (14.1.4.1)
      Origin incomplete, MED 0, localpref 100, weight 0
      Received label 492529
      Extcommunity: RT:2:20

  Path-id 1 not advertised to any peer

Route Distinguisher: 2.2.2.3:113
BGP routing table entry for [5]:[0]:[0]:[24]:[41.11.2.0]:[0.0.0.0]/224, version 17369595
Paths: (1 available, best #1)
Flags: (0x000002) on xmit-list, is not in l2rib/evpn, is not in HW

  Advertised path-id 1
  Path type: external, path is valid, is best path
             Imported from 14.1.4.1:115:[5]:[0]:[0]:[24]:[41.11.2.0]:[0.0.0.0]/224
  AS-Path: 11 , path sourced external to AS
    1.1.1.9 (metric 0) from 1.1.1.9 (14.1.4.1)
```

セグメント ルーティングと IS-IS プロトコル

IS-IS について

IS-IS は、ISO（国際標準化機構）/IEC（国際電気標準化会議）10589 および RFC 1995 に基づく IGP（内部ゲートウェイ プロトコル）です。Cisco NX-OS は、インターネット プロトコル バージョン 4（IPv4）および IPv6 をサポートします。IS-IS はネットワーク トポロジの変化を検出し、ネットワーク上の他のノードへのループフリー ルートを計算できる、ダイナミック リンクステート ルーティング プロトコルです。各ルータは、ネットワークの状態を記述するリンクステートデータベースを維持し、設定された各リンクにパケットを送信してネイバーを検出します。IS-IS はネットワークを介して各ネイバーにリンクステート情報をフラッディングします。ルータもすべての既存ネイバーを通じて、リンクステートデータベースのアドバタイズメントおよびアップデートを送信します。

IS-IS プロトコルでのセグメント ルーティングは、次をサポートしています。

- IPv4
- レベル 1、レベル 2、およびマルチレベルのルーティング
- プレフィックス SID
- ドメイン ボーダー ノード用の同じループバック インターフェイス上の複数の IS-IS インスタンス
- 隣接関係用の隣接関係 SID

IS-IS プロトコルでのセグメント ルーティングの設定

セグメント ルーティングは IS-IS プロトコルで設定できます。

始める前に

次の条件が満たされると、IS-IS セグメント ルーティングが完全に有効になります。

- **mpls segment-routing** 機能が有効になっていること。
- IS-IS 機能が有効になっていること。
- セグメント ルーティングが、IS-IS の下で少なくとも 1 つのアドレス ファミリに対して有効になっていること。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal**	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**router isis** *instance-tag*	instance tag を設定して、新しい IS-IS インスタンスを作成します。
ステップ 3	**net** *network-entity-title*	この IS-IS インスタンスに対応する NET を設定します。
ステップ 4	**address-family** *ipv4* **unicast**	アドレス ファミリ設定モードを開始します。
ステップ 5	**segment-routing mpls**	セグメント ルーティングを IS-IS プロトコルで設定します。 （注） • IS-IS コマンドは、IPv4 アドレス ファミリでのみサポートされます。IPv6 アドレス ファミリではサポートされていません。 • SR プレフィックスの他のプロトコルから ISIS への再配布はサポートされていません。すべてのプレフィックス SID インターフェイスで **ip router isis** コマンドを有効にする必要があります。

セグメント ルーティングと OSPFv2 プロトコル

OSPF について

Open Shortest Path First（OSPF）は、Internet Engineering Task Force（IETF）の OSPF ワーキング グループによって開発された内部ゲートウェイ プロトコル（IGP）です。OSPF は特に IP ネットワーク向けに設計されており、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ギング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。

OSPF プロトコルのセグメント ルーティング設定は、プロセス レベルまたはエリア レベルで適用できます。プロセス レベルでセグメント ルーティングを設定すると、すべてのエリアで有効になります。ただし、エリア レベルごとに有効または無効にすることもできます。

OSPF プロトコルでのセグメント ルーティングは、次をサポートしています。

- OSPFv2 のコントロール プレーン
- マルチエリア
- ループバック インターフェイス上のホスト プレフィックスの IPv4 プレフィックス SID
- 隣接関係用の隣接関係 SID

隣接関係 SID のアドバタイズメント

OSPF は、セグメント ルーティング隣接関係 SID のアドバタイズメントをサポートしています。隣接関係セグメント識別子（Adj-SID）は、セグメント ルーティングにおけるルータ隣接関係を表します。

セグメントルーティング対応ルータは、隣接関係ごとにAdj-SIDを割り当てることができ、この SID を拡張不透明リンク LSA で伝送するように Adj-SID サブ TLV が定義されます。

OSPF は、OSPF 隣接関係が 2 つの方法または完全な状態にある場合、各 OSPF ネイバーに隣接関係 SID を割り当てます。OSPF は、セグメント ルーティングが有効になっている場合にのみ隣接関係 SID を割り当てます。隣接関係 SID のラベルは、システムによって動的に割り当てられます。これにより、ローカルでしか有効でないため、設定ミスの可能性がなくなります。

接続されたプレフィックス SID

OSPFv2 は、ループバック インターフェイスに関連付けられたアドレスのプレフィックス SID のアドバタイズをサポートします。これを実現するために、OSPFは、不透明な拡張プレフィックス LSA で拡張プレフィックス サブ TLV を使用します。OSPF がネイバーからこの LSA を受信すると、SR ラベルは、拡張プレフィックス サブ TLV に存在する情報に基づいて、受信したプレフィックスに対応する RIB に追加されます。

設定では、セグメント ルーティングを OSPF で有効にする必要があり、OSPF で設定されたループバック インターフェイスに対応して、セグメントルーティング モジュールでプレフィックス-SID マッピングが必要です。

（注） SID は、ループバック アドレスに対してのみ、またエリア内およびエリア間プレフィックス タイプに対してのみアドバタイズされます。外部プレフィックスまたは NSSA プレフィックスの SID 値はアドバタイズされません。

エリア間のプレフィックス伝播

エリア境界を越えたセグメント ルーティング サポートを提供するには、エリア間で SID 値を伝播するために OSPF が必要です。OSPF は、エリア間のプレフィックス到達可能性をアドバタイズするときに、プレフィックスの SID がアドバタイズされているかどうかを確認します。通常、SID 値はルータから取得され、送信元エリアのプレフィックスへの最適なパスに寄与します。この場合、OSPF はその SID を使用してエリア間でアドバタイズを行います。SID 値がエリア内のベスト パスに寄与するルータによってアドバタイズされない場合、OSPF は送信元エリア内の他のルータからの SID 値を使用します。

セグメント ルーティングのグローバル範囲の変更

OSPF は、SID/ラベル範囲 TLV のアドバタイズに関して、そのセグメント ルーティング機能をアドバタイズします。OSPFv2 では、SID/ラベル範囲 TLV はルータ情報 LSA で伝えられます。

セグメント ルーティングのグローバル範囲設定は、「segment-routing mpls」設定の下にあります。OSPF プロセスが来たら、segment-routing からグローバル範囲の値を取得し、その後の変更はそれに伝播する必要があります。

OSPF セグメント ルーティングが設定されている場合、OSPF は、OSPF セグメント ルーティングの動作状態を有効にする前に、セグメント ルーティング モジュールとのインタラクションをリクエストする必要があります。SRGB 範囲が作成されていない場合、OSPF は有効になりません。SRGB 変更イベントが発生した場合、OSPF は、そのサブブロック エントリで対応する変更を行います。

SID エントリの競合処理

理想的な状況では、各プレフィックスに一意の SID エントリが割り当てられている必要があります。

SID エントリと関連付けられているプレフィックス エントリの間に競合がある場合は、次のいずれかの方法を使用して競合を解決します。

- 1 つのプレフィックスに複数の SID：同じプレフィックスが異なる SID を持つ複数の送信元によってアドバタイズされる場合、OSPF はそのプレフィックスのラベルのないパスをインストールします。OSPF は、到達可能なルータからの SID のみを考慮し、到達不能なルーターからの SID は無視します。1 つのプレフィックスに対して複数の SID がアドバタイズされると、競合と見なされ、そのプレフィックスの接続領域に SID はアドバタイズされません。同様のロジックは、バックボーン エリアと非バックボーン エリアの間でエリア間プレフィックスを伝搬するときにも使用されます。
- SID の範囲外：SID 範囲に収まらない SID の場合、RIB の更新時にラベルは使用されません。

インターフェイスでの MPLS 転送

セグメントルーティングがインターフェイスを使用する前に、MPLS転送を有効にする必要があります。OSPF は、インターフェイスでの MPLS 転送を有効にする役割を担います。

セグメントルーティングが OSPF トポロジに対して有効になっている場合、またはOSPF セグメントルーティングの動作状態が有効になっている場合、OSPF は、OSPF トポロジがアクティブである任意のインターフェイスに対して MPLS を有効にします。同様に、OSPF トポロジのセグメント ルーティングが無効になっている場合、OSPF は、そのトポロジのすべてのインターフェイスで MPLS 転送を無効にします。

MPLS転送は、IPIP/GRE トンネルを終端するインターフェイスではサポートされていません。

OSPFv2 でのセグメント ルーティングの設定

セグメント ルーティングを OSPFv2 プロトコルで設定します。

始める前に

OSPFv2 でセグメント ルーティングを設定する前に、次の条件が満たされていることを確認してください。

- OSPFv2 機能が有効になっている。
- セグメント ルーティング機能が有効になっている。
- セグメント ルーティングが OSPF で有効になっている。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル 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	[**no**]**router ospf** *process* 例： `switch(config)# router ospf test`	OSPF モードを有効にします。
ステップ 3	**segment-routing** 例： `switch(config-router)# segment-routing mpls`	OSPF でのセグメントルーティング機能を設定します。

OSPF ネットワークでのセグメント ルーティングの設定：エリア レベル

始める前に

OSPF ネットワークでセグメント ルーティングを設定する前に、ネットワーク上で OSPF 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	**router ospf process** 例： `switch(config)# router ospf test`	OSPF モードを有効にします。
ステップ 2	**area** <*area id*> **segment-routing** [**mpls** | **disable**] 例： `switch(config-router)# area 1 segment-routing mpls`	特定の領域にセグメント ルーティング MPLS モードを設定します。
ステップ 3	[**no**]**area** <*area id*> **segment-routing** [**mpls** | **disable**] 例： `switch(config-router)#area 1 segment-routing disable`	指定されたエリアのセグメントルーティング mpls モードを無効にします。
ステップ 4	**show ip ospf** プロセス **segment-routing** 例： `switch(config-router)# show ip ospf test segment-routing`	OSPF の下で SR を設定するための出力を示します。

OSPF のプレフィックス SID の設定

ここでは、各インターフェイスでプレフィックスセグメント ID（SID）を設定する方法について説明します。

始める前に

セグメント ルーティングを対応するアドレス ファミリで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	**configure terminal** 例： `switch# configure terminal`	グローバル コンフィギュレーション モードを開始します
ステップ 2	[**no**]**router ospf** *process* 例： `switch(config)# router ospf test`	OSPF を設定します。

ステップ 3	**segment-routing**
	例：
	`switch(config-router)# segment-routing`
	`switch(config-sr)#mpls`
	`switch(config-sr-mpls)#`
		OSPF でのセグメント ルーティング機能を設定します。

ステップ 4	**interface loopback** *interface_number* 例： `switch(config-sr-mpls)# Interface loopback 0`	OSPF が有効になっているインターフェイスを指定します。
ステップ 5	**ip address 1.1.1.1/32** 例： `switch(config-sr-mpls)# ip address 1.1.1.1/32`	ospf インターフェイスで設定された IP アドレスを指定します。
ステップ 6	**ip router ospf 1 area 0** 例： `switch(config-sr-mpls)# ip router ospf 1 area 0`	エリア内のインターフェイスで有効になっている OSPF を指定します。

ステップ 7	**segment-routing**
	例：
	`switch(config-router)#segment-routing`
	`(config-sr)#mpls`
		SR モジュールの下でプレフィックス SID マッピングを設定します。

ステップ 8	**connected-prefix-sid-map**
	例：
	`switch(config-sr-mpls)#`
	`connected-prefix-sid-map`
	`switch(config-sr-mpls-conn-pfxsid)#`
		セグメントルーティング モジュールの下でプレフィックス SID マッピングを設定します。

ステップ 9	**address-family ipv4**
	例：
	`switch(config-sr-mpls-conn-pfxsid)#`
	`address-family ipv4`
	`switch(config-sr-mpls-conn-pfxsid-af)#`
		OSPF インターフェイスで設定されている IPv4 アドレス ファミリを指定します。

	コマンドまたはアクション	目的
ステップ 10	**1.1.1.1/32 index 10** 例： `switch(config-sr-mpls-conn-af)# 1.1.1.1/32 index 10`	SID 100 にアドレス 1.1.1.1/32 を関連付けます。
ステップ 11	**exit** 例： `switch(config-sr-mpls-conn-af)# exit`	セグメントルーティングモードを終了し、コンフィギュレーション端末モードに戻ります。

プレフィックス属性 N-flag-clear の設定

OSPF は、その不透明 LSA に拡張プレフィックス TLV を介してプレフィックス SID をアドバタイズします。これはプレフィックスのフラグを伝送します。そのうちの1つはNフラグ（ノード）で、プレフィックスに沿って送信されたトラフィックが、LSAを発信するルータ宛てであることを示します。このフラグは通常、ルータのループバックのホスト ルートをマークします。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `switch# configure terminal switch(config)#`	グローバル 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 loopback3** 例： `switch(config)# interface loopback3`	インターフェイス ループバックを指定します。
ステップ 3	**ip ospf prefix-attributes n-flag-clear** 例： `switch#(config-if)# ip ospf prefix-attributes n-flag-clear`	プレフィックスN-flagをクリアします。

OSPF のプレフィックス SID の設定例

この例は、OSPF のプレフィックス SID の設定を示しています。

```
Router ospf 10
     Segment-routing mpls
Interface loop 0
    Ip address 1.1.1.1/32
    Ip router ospf 10 area 0
Segment-routing
```

```
Mpls
   connected-prefix-sid-m
       address-family ipv4
           1.1.1.1/32 index 10
```

BGP を使用したプレフィックス SID の構成

network コマンドにマッチするルートのラベル インデックスを設定できます。これにより、**set label-index** コマンドを含むルート マップで構成されているローカル プレフィックスに対して BGP プレフィックス SID がアドバタイズされます。ただし、ローカル プレフィックスを指定する **network** コマンドでルート マップが指定されていることが必要です。（**[[ネットワーク（network）]** コマンドの詳細については、Cisco Nexus 3600 Series NX-OS Unicast Routing Configuration Guide の「Configuring Basic BGP」の章を参照してください。)（(For more information on the network command, see the "Configuring Basic BGP" chapter in the Cisco Nexus 3600 Series NX-OS Unicast Routing Configuration Guide.)）]

（注） ルートマップが **network** コマンド以外のコンテキストで指定されている場合、ルートマップ ラベル インデックスは無視されます。また、プレフィックスが **allocate-label route-map** *route-map-name* コマンドで設定されているかどうかに関係なく、ルート マップ ラベル インデックスを使用してプレフィックスにラベルが割り当てられます。

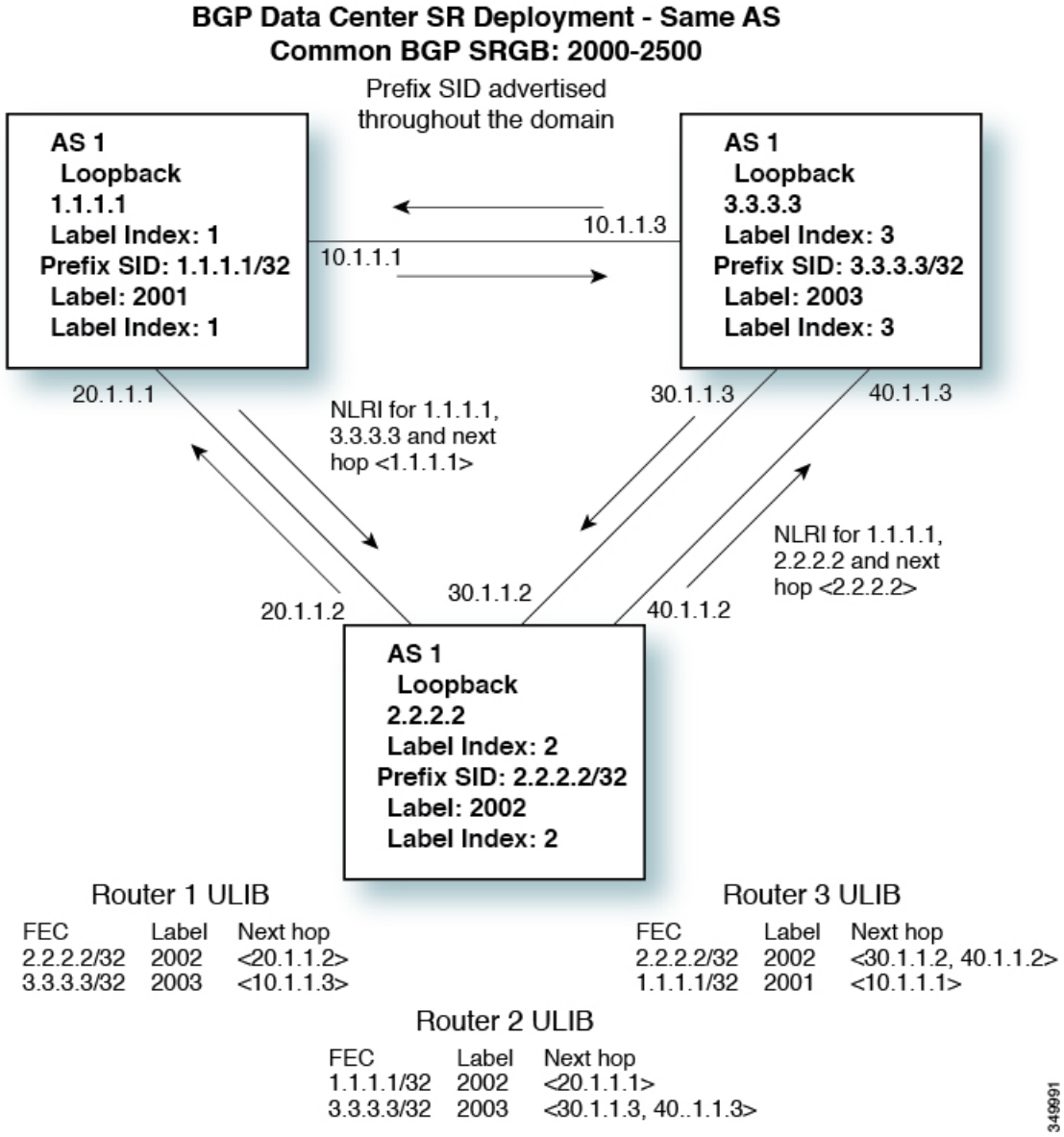
BGP プレフィックス SID

セグメント ルーティングをサポートするためには、BGP が BGP プレフィックスのセグメント ID（SID）をアドバタイズできなければなりません。BGP プレフィックス SID は常にセグメント ルーティング BGP ドメイン内でグローバルであり、命令を識別し、BGP によって計算された ECMP 対応のベストパスを介して、パケットを関連するプレフィックスに転送します。BGP プレフィックス SID は、BGP プレフィックス セグメントを識別します。

BGP プレフィックス SID の展開例

以下の簡単な例では、3 つのルーターすべてが iBGP を実行し、ネットワーク層到達可能性情報（NRLI）を互いにアドバタイズしています。また、ルーターは、ルーター 2.2.2.2 と 3.3.3.3 の間に ECMP を提供するネクスト ホップとして、ループバック インターフェイスをアドバタイズしています。

図 *1 : BGP* プレフィックス *SID* の簡単な例



隣接 SID

隣接関係セグメント識別子（SID）は、特定のインターフェイスとそのインターフェイスからの次のホップを指す、ローカル ラベルです。隣接関係 SID を有効にするために必要な特定の設定はありません。アドレス ファミリの BGP を介してセグメント ルーティングが有効になると、BGP が実行されるすべてのインターフェイスに対して、アドレスファミリがそのインターフェイスのすべてのネイバーに対して隣接 SID を自動的に割り当てます。

セグメント ルーティングのための高可用性

インサービス ソフトウェア アップグレード (ISSU) は、BGP グレースフル リスタートで最低限サポートされます。すべての状態（セグメント ルーティング状態を含む）は、BGP ルータのピアから再学習する必要があります。グレースフル リスタート期間中、以前に学習したルートとラベルの状態は保持されます。

ラベル インデックスの構成

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル 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-name*
	例：
	`switch(config)# route-map SRmap`
	`switch(config-route-map)#`
		ルート マップを作成するか、または既存のルート マップに対応するルート マップ コンフィギュレーション モードを開始します。

ステップ 3	[**no**] **set label-index** *index* 例： `switch(config-route-map)# set label-index 10`	**network** コマンドにマッチするルートのラベル インデックスを設定します。範囲は 0 ～ 471788 です。デフォルトでは、ラベル インデックスはルートに追加されません。

ステップ 4	**exit**
	例：
	`switch(config-route-map)# exit`
	`switch(config)#`
		ルートマップ設定モードを終了します。

ステップ 5	**router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp 64496`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに AS 番号を割り当てます。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

ステップ 6	必須: **address-family ipv4 unicast**
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		IPv4 アドレス ファミリに対応するグローバル 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 7	**network** *ip-prefix* [**route-map** *map-name*] 例： `switch(config-router-af)# network 10.10.10.10/32 route-map SRmap`	ネットワークを、この自律システムに対してローカルに設定し、BGP ルーティング テーブルに追加します。
ステップ 8	（任意） **show route-map** [*map-name*] 例： `switch(config-router-af)# show route-map`	ラベル インデックスなど、ルート マップに関する情報を表示します。
ステップ 9	（任意） **copy running-config startup-config** 例： `switch(config-router-af)# copy running-config startup-config`	実行設定を、スタートアップ設定にコピーします。

MPLS ラベル割り当ての構成

IPv4 ユニキャスト アドレス ファミリの MPLS ラベル割り当てを構成できます。

始める前に

MPLS 機能セットは、**install feature-set mpls** および **feature-set mpls** コマンドを使用してインストールし、有効にする必要があります。

MPLS セグメント ルーティング機能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	**configure terminal** 例： `switch# configure terminal switch(config)#`	グローバル 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	[**no**] **router bgp** *autonomous-system-number* 例： `switch(config)# router bgp 64496 switch(config-router)#`	BGP を有効にして、ローカル BGP スピーカに AS 番号を割り当てます。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

	コマンドまたはアクション	目的
		BGP プロセスおよび関連する設定を削除するには、このコマンドで **no** オプションを使用します。

ステップ 3	必須: **address-family ipv4 unicast**
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		IPv4 アドレス ファミリに対応するグローバル 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	[**no**] **allocate-label** {**all** | **route-map** *route-map-name*} 例： `switch(config-router-af)# allocate-label route-map map1`	指定されたルートマップに一致するルート、またはこのアドレス ファミリでアドバタイズされるすべてのルートのローカル ラベル割り当てを構成します。

ステップ 5	必須: **exit**
	例：
	`switch(config-router-af)# exit`
	`switch(config-router)#`
		グローバル 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 6	**neighbor** *ipv4-address* **remote-as** *autonomous-system-number*
	例：
	`switch(config-router)# neighbor 10.1.1.1 remote-as 64497`
	`switch(config-router-neighbor)#`
		リモート BGP ピアの IPv4 アドレスおよび AS 番号を設定します。

ステップ 7	**address-family ipv4 labeled-unicast**
	例：
	`switch(config-router-neighbor)# address-family ipv4 labeled-unicast`
	`switch(config-router-neighbor-af)#`
		RFC 3107 で指定されているように、ラベル付き IPv4 ユニキャスト ルートをアドバタイズします。

ステップ 8	（任意） **show bgp ipv4 labeled-unicast** *prefix* 例： `switch(config-router-neighbor-af)# show bgp ipv4 labeled-unicast 10.10.10.10/32`	指定された IPv4 プレフィックスのアドバタイズされたラベル インデックスおよび選択されたローカル ラベルを表示します。
ステップ 9	（任意） **copy running-config startup-config** 例： `switch(config-router-neighbor-af)# copy running-config startup-config`	実行設定を、スタートアップ設定にコピーします。

BGP プレフィックス SID の構成例

このセクションの例は、2 台のルータ間の一般的な BGP プレフィックス SID 構成を示しています。

この例は、10.10.10.10/32 と 20.20.20.20/32 の BGP スピーカー構成を、それぞれ 10 と 20 のラベル インデックスでアドバタイズする方法を示しています。16000 ～ 23999 のデフォルトのセグメント ルーティング グローバル ブロック （SRGB）範囲を使用します。

```
hostname s1
install feature-set mpls
feature-set mpls

feature telnet
feature bash-shell
feature scp-server
feature bgp
feature mpls segment-routing

segment-routing mpls
vlan 1

route-map label-index-10 permit 10
  set label-index 10
route-map label-index-20 permit 10
  set label-index 20

vrf context management
  ip route 0.0.0.0/0 10.30.108.1

interface Ethernet1/1
  no switchport
  ip address 10.1.1.1/24
  no shutdown

interface mgmt0
  ip address dhcp
  vrf member management

interface loopback1
  ip address 10.10.10.10/32

interface loopback2
  ip address 20.20.20.20/32

line console
line vty

router bgp 1
  address-family ipv4 unicast
    network 10.10.10.10/32 route-map label-index-10
    network 20.20.20.20/32 route-map label-index-20
    allocate-label all
  neighbor 10.1.1.2 remote-as 2
    address-family ipv4 labeled-unicast
```

この例は、BGP スピーカーからの構成を受信する方法を示しています。

```
hostname s2
install feature-set mpls
feature-set mpls
```

```
feature telnet
feature bash-shell
feature scp-server
feature bgp
feature mpls segment-routing

segment-routing mpls
vlan 1

vrf context management
  ip route 0.0.0.0/0 10.30.97.1
  ip route 0.0.0.0/0 10.30.108.1

interface Ethernet1/1
  no switchport
  ip address 10.1.1.2/24
  ipv6 address 10:1:1::2/64
  no shutdown

interface mgmt0
  ip address dhcp
  vrf member management

interface loopback1
  ip address 2.2.2.2/32
line console

line vty

router bgp 2
  address-family ipv4 unicast
    allocate-label all
  neighbor 10.1.1.1 remote-as 1
    address-family ipv4 labeled-unicast
```

BGP リンク ステート アドレス ファミリの設定

対応する SID をアドバタイズするコントローラを持つネイバー セッションに対し、 BGP リンク ステート アドレス ファミリを設定することができます。この機能は、グローバル コンフィギュレーション モードおよびネイバー アドレス ファミリ コンフィギュレーション モードで設定できます。

始める前に

BGP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	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル 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 bgp** <*bgp autonomous number*>	自律ルータ BGP 番号を指定します。

ステップ 3	[**no**] **address-family link-state**
	例：
	`switch(config)# router bgp 64497`
	`switch (config-router af)#`
	`address-family link-state`
		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。
		（注） このコマンドは、ネイバー アドレス ファミリ コンフィギュレーション モードでも設定できます。

ステップ 4	**neighbor** <*IP address*>	ネイバーの IP アドレスを設定します。

ステップ 5	[**no**] **address-family link-state**
	例：
	`switch(config)#router bgp 1`
	`switch(config-router)#address-family link-state`
	`switch(config-router)#neighbor 20.20.20.20`
	`switch(config-router)#address-family link-state`
		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。
		（注） このコマンドは、ネイバー アドレス ファミリ コンフィギュレーション モードでも設定できます。

セグメント ルーティングの設定の確認

スタティック ルーティングの設定を表示するには、次のいずれかの作業を行います。

コマンド	目的
show bgp ipv4 labeled-unicast *prefix*	指定された IPv4 プレフィックスのアドバタイズされたラベル インデックスおよび選択されたローカル ラベルを表示します。
show bgp paths	アドバタイズされたラベル インデックスを含む BGP パス情報を表示します。
show mpls label range	構成されたラベルの SRGB 範囲を表示します。
show route-map [*map-name*]	ラベル インデックスなど、ルート マップに関する情報を表示します。
show running-config | **inc 'feature segment-routing'**	MPLS セグメント ルーティング機能のステータスを表示します。
show ip ospf neighbors detail	OSPFv2 ネイバー、および割り当てられた隣接関係 SID のリストを、対応するフラグとともに表示します。

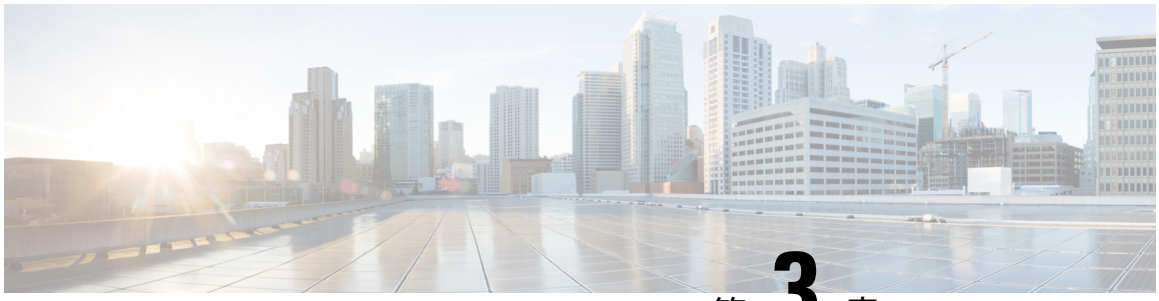
コマンド	目的
show ip ospf database opaque-area	隣接 SID の LSA を表示します。
show ip ospf segment-routing adj-sid-database	ローカルに割り当てられた隣接 SID をすべて表示します。
show running-config segment-routing	セグメント ルーティング機能のステータスを表示します。
show srte policy	許可されたポリシーのみを表示します。
show srte policy [**all**]	SR-TE で使用可能なすべてのポリシーのリストを表示します。
show srte policy [**detail**]	要求されたすべてのポリシーの詳細ビューを表示します。
show srte policy <*name*>	SR-TE ポリシーを名前でフィルタリングし、SR-TE でその名前で使用できるすべてのポリシーのリストを表示します。 （注） このコマンドには、ポリシー名のオートコンプリート機能があります。この機能を使用するには、疑問符を追加するか、TAB キーを押します。
show srte policy color <*color*> **endpoint** <*endpoint*>	カラーとエンドポイントの SR-TE ポリシーを表示します。 （注） このコマンドには、カラーとエンドポイントのオートコンプリート機能があります。この機能を使用するには、疑問符を追加するか、TAB キーを押します。
show srte policy fh	最初のホップのセットを表示します。
show segment-routing mpls clients	SR-APP に登録されているクライアントを表示します。
show segment-routing mpls details	詳細情報を表示します。
show segment-routing ipv4	IPv4 アドレス ファミリの BGP 情報を表示します。
show segment-routing mpls	セグメント ルーティング MPLS 情報を表示します

コマンド	目的
show segment-routing ipv4 connected-prefix-sid	SRGB の MPLS ラベル範囲を表示します。 （注） このコマンドは、Cisco NX-OS リリース 9.3(1) でのみ使用できます。
show ip ospf プロセス	OSPF モードを表示します。
show ip ospf プロセス **segment-routing sid-database**	セグメント ルーティング データベースの詳細を表示します。
show ip ospf プロセス **segment-routing global block**	セグメント ルーティング グローバル ブロック情報を表示します。

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
BGP	*[Cisco Nexus 3600* シリーズ ユニキャスト ルーティング構成ガイド（*Cisco Nexus 3600 Series Unicast Routing Configuration Guide*）*]*



第 3 章

『Configuring MPLS Layer 3 VPNs』

この章では、Cisco Nexus 3600 シリーズ スイッチでマルチプロトコル ラベル スイッチング（MPLS）レイヤ 3 仮想プライベート ネットワーク（VPN）を構成する方法について説明します。

MPLS レイヤ 3 VPNs の概要

MPLS レイヤ 3 VPN は、MPLS プロバイダー コア ネットワークにより相互接続されている一連のサイトから構成されます。各カスタマーサイトでは、1つ以上のカスタマーエッジ（CE）ルータまたはレイヤ2スイッチが、1つ以上のプロバイダーエッジ（PE）ルータに接続されます。ここでは次の項目について説明します。

- MPLS レイヤ 3 VPN の定義
- MPLS レイヤ 3 VPN の動作方法
- MPLS レイヤ 3 VPN のコンポーネント
- ハブ アンド スポーク トポロジ
- MPLS VPN のための OSPF 模造リンクのサポート

MPLS レイヤ 3 VPN の定義

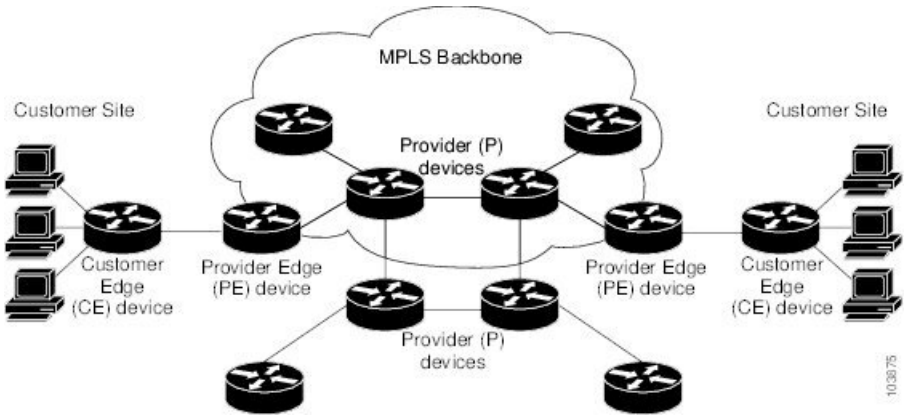
MPLS レイヤ 3 VPN はピア モデルに基づいており、これにより、サービス プロバイダーおよびカスタマーは、レイヤ3のルーティング情報を交換できます。プロバイダーは、カスタマーサイト間でデータをリレーします。このとき、カスタマーが直接何かを行う必要はありません。

新しいサイトが MPLS VPN に追加された場合、更新する必要があるのは、カスタマー サイトにサービスを提供するサービス プロバイダーのエッジ ルータだけです。

MPLS レイヤー 3 VPN には、以下のコンポーネントが含まれています。

- プロバイダー（P）ルータ：プロバイダー ネットワークのコア内のルータ。P ルータは MPLS スイッチングを実行しますが、ルーティングされるパケットに VPN ラベル（PE ルータによって割り当てられた、各ルート内の MPLS ラベル）を付加しません。P ルーターは、Label Distribution Protocol（LDP）に基づいてパケットを転送します。
- プロバイダー エッジ（PE）ルータ：着信パケットが受信されるインターフェイスまたはサブインターフェイスに基づいて、着信パケットに VPN ラベルを付加するルータ。PE ルータは、CE ルータに直接接続します。
- カスタマーエッジ（CE）ルータ：ネットワーク上の PE ルータに接続するプロバイダーのネットワーク上のエッジルータ。CE ルータは、PE ルータとインターフェイスする必要があります。

図 *2 : MPLS* レイヤ *3 VPN* の基本用語



MPLS レイヤ 3 VPN の動作方法

MPLS レイヤ 3 VPN 機能は、MPLS ネットワークのエッジで有効になっています。PE ルータは、次のタスクを実行します。

- CE ルータとルーティング アップデートを交換する。
- CE ルーティング情報を VPN ルートに変換する。
- マルチプロトコル ボーダー ゲートウェイ プロトコル（MP-BGP）を介して、他の PE ルータとレイヤ 3 VPN ルートを交換する。

MPLS レイヤ 3 VPN のコンポーネント

MPLS ベースの VPN 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。

1. VPN ルート ターゲット コミュニティ：VPN ルート ターゲット コミュニティは、レイヤ 3 VPN コミュニティのすべてのメンバのリストです。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。
2. 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：マルチプロトコル BGP は、VPN コミュニティのすべてのメンバに VRF の到達可能情報を伝播します。VPN コミュニティ内のすべての PE ルータにマルチプロトコル BGP ピアリングを設定する必要があります。
3. MPLS 転送：MPLS は、VPN エンタープライズまたはサービス プロバイダー ネットワーク上のすべての 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。

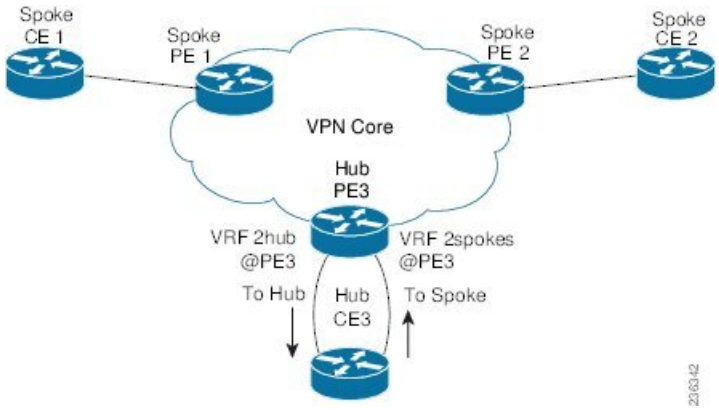
1 対 1 の関係は、カスタマー サイトと VPNs 間に必ずしも存在する必要はありません。1 つのサイトを複数の VPNs のメンバにできます。ただし、サイトは、1 つの VRF とだけ関連付けることができます。カスタマー サイトの VRF には、そのサイトがメンバとなっている VPNs からサイトへの、利用できるすべてのルートが含まれています。

ハブ アンド スポーク トポロジ

ハブ アンド スポーク トポロジは、スポーク プロバイダー エッジ（PE）ルータでの加入者間のローカル接続を禁止し、加入者がハブ サイトに常に接続されるようにします。同じ PE ルータに接続しているすべてのサイトは、ハブ サイトを使用して、サイト間のトラフィックを転送する必要があります。このトポロジより、スポーク サイトでのルーティングは、常にアクセス側インターフェイスからネットワーク側インターフェイスに対して、またはネットワーク側インターフェイスからアクセス側インターフェイスに対して実行されます。アクセス側インターフェイスからアクセス側インターフェイスへのルーティングは発生しません。ハブ アンド スポーク トポロジにより、サイト間のアクセス制限を維持できます。

ハブ アンド スポーク トポロジを使用すると、PE ルータが、トラフィックをハブ サイトを介して渡さずに、スポークをローカルに切り替えるという状況が回避されます。このトポロジにより、加入者が互いに直接接続することがなくなります。ハブアンドスポーク トポロジでは、スポークごとに 1 つの VRF は必要ありません。

図 *3:* ハブ アンド スポーク トポロジ



図に示すように、ハブ アンド スポーク トポロジは通常、2 つの VRF で設定されたハブ PE で設定されます。

- 専用リンクが設定された VRF 2hub がハブのカスタマー エッジ（CE）に接続されます。
- VRF 2spoke は、ハブ CE に接続された別の専用リンクを使用します。

内部ゲートウェイ プロトコル（IGP）または外部 BGP（eBGP）セッションは、通常、ハブ PE-CE リンクを介してセットアップされます。VRF 2hub は、すべてのスポーク PE からエクスポートされたすべてのルート ターゲットをインポートします。ハブ CE はスポーク サイトからのすべてのルートを学習し、それらをハブ PE の VRF 2spoke に再アドバタイズして戻します。VRF 2spoke は、これらすべてのルートをスポーク PE にエクスポートします。

ハブ PE とハブ CE の間の eBGP を使用する場合は、通常は禁止されているパスで自律システム（AS）番号を複製できるようにする必要があります。ハブ PE の VRF 2spoke のネイバー、およびすべてのスポーク PE の VPN アドレス ファミリ ネイバーでこの重複 AS 番号を許可するようにルータを設定できます。さらに、ハブ PE の VRF 2spoke でネイバーにルートを配布する場合は、ハブ CE でピア AS 番号チェックを無効にする必要があります。

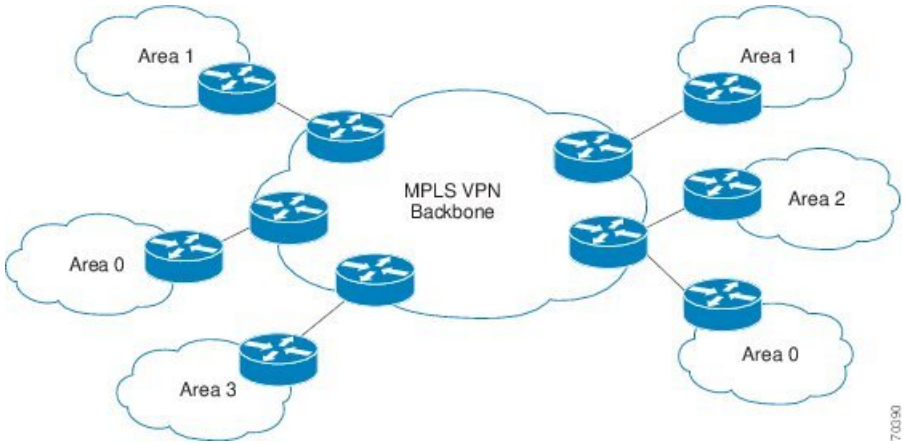
MPLS VPN のための OSPF 模造リンクのサポート

マルチプロトコル ラベル スイッチング（MPLS）VPN 構成では、Open Shortest Path First（OSPF）プロトコルを使用して、VPN バックボーン内のカスタマー エッジ（CE）デバイスをサービス プロバイダーエッジ（PE）デバイスに接続できます。多くのカスタマーは、OSPF をサイト内ルーティング プロトコルとして実行し、VPN サービスにサブスクライブし、MPLS VPN バックボーンで OSPF を（移行時または常時）使用してサイト間でルーティング情報を交換することを望んでいます。

MPLS VPN の OSPF 模造リンク サポートの利点は次のとおりです。

- MPLS VPN バックボーン全体でのクライアント サイトの接続：模造リンクによって、バックドア リンクを共有する OSPF クライアント サイトが、MPLS VPN バックボーンを介して通信を行い、VPN サービスに参加するようになります。
- MPLS VPN 設定での柔軟なルーティング：MPLS VPN 設定で模造リンクに対して設定する OSPF コストを使用して、OSPF クライアント サイトのトラフィックを、バックドア リンク経由にするか、または VPN バックボーン経由にするかを指定できます。

下の図に、OSPF を実行する各 VPN クライアント サイトを、MPLS VPN バックボーンで接続する例を示します。



OSPF を使用して PE デバイスと CE デバイスを接続するには、VPN サイトから学習したすべてのルーティング情報を、着信インターフェイスに関連付けられた VPN ルーティングおよび転送（VRF）インスタンスに格納します。VPN に接続された PE デバイス間では、ボーダーゲートウェイ プロトコル（BGP）を使用して、VPN ルートが交換されます。CE デバイスはこの VPN 内の他のサイトへのルートを、自分が接続された PE デバイスとのピアリングによって学習します。MPLS VPN スーパーバックボーンは、OSPF を実行する各 VPN サイトを内部接続するための追加のルーティング階層レベルを提供します。

OSPF ルートが MPLS VPN バックボーン全体に伝播されると、プレフィックスに関する追加情報が、BGP 拡張コミュニティ形式（ルート タイプ、ドメイン ID 拡張コミュニティ）で BGP アップデートに付加されます。このコミュニティ情報を使用して、受信した PE デバイスは、BGP ルートを OSPF PE-CE プロセスに再配布するときに生成するリンクステート アドバタイズメント（LSA）のタイプを決定します。このようにして、同じ VPN に属し、VPN バックボーン全体にアドバタイズされる内部 OSPF ルートが、リモート サイト上でエリア内ルートとして認識されます。

MPLS レイヤ 3 VPNs の前提条件

MPLS レイヤ 3 VPNs には次の前提条件があります。

- ネットワークに MPLS およびラベル配布プロトコル（LDP）を設定する必要があります。PE ルータを含む、コア内のすべてのルータは、MPLS 転送をサポートできる必要があります。
- MPLS の正しいライセンスおよび MPLS で使用する他の機能をインストールすることが必要です。

MPLS レイヤ 3 VPNs に関する注意事項と制限事項

MPLS レイヤ 3 VPNs 設定時の注意事項と制限事項は次のとおりです。

- Cisco Nexus 3600-R プラットフォーム スイッチおよび N9K-X9636C-RX、N9K-X9636C-R、N9K-X96136YC-R、および N9K-X9636Q-R ライン カードを搭載した および Cisco Nexus 9504 および 9508 プラットフォーム スイッチで、MPLS レイヤ 3 VPN（LDP）を設定できます。
- 着信パケットのラベルに基づいて転送の決定が行われるインターフェイスでは、MPLS IP 転送を有効にする必要があります。VPN ラベルがプレフィックス モードごとに割り当てられている場合は、PE と CE 間のリンクで MPLS IP 転送を有効にする必要があります。
- MPLS の明示的 NULL のパケットは、デフォルトのラインカードプロファイルでは正しく解析されない場合があります。
- MPLS レイヤ 3 VPN は、次の CE-PE ルーティング プロトコルをサポートします。
  - BGP（IPv4 および IPv6）
  - 拡張内部ゲートウェイ プロトコル（EIGRP）（IPv4）
  - Open Shortest Path First（OSPFv2）
  - ルーティング情報プロトコル（RIPv2）

  インポート ルート マップの set ステートメントは無視されます。
- すべての iBGP および eBGP セッションの BGP 最小ルート アドバタイズメント インターバル（MRAI）値はゼロであり、設定できません。
- EIGRP に多数の BGP ルートが再配布されるハイ スケールなセットアップでは、EIGRP のコンバージェンス時間が BGP のコンバージェンス時間よりも長くなるように EIGRP シグナル タイマーの設定を変更する必要があります。このプロセスにより、EIGRP シグナルのコンバージェンス前にすべての BGP ルートを EIGRP に再配布することができます。
- PE と CE デバイス間のプロトコルとして OSPF を使用する場合、VPN バックボーン全体にルートがアドバタイズされる際、OSPF メトリックは保持されます。このメトリックは、リモート PE デバイスで適切なルートを選択するために使用されます。OSPF から BGP への再配布、および、BGP から OSPF への再配布において、メトリック値を変更しないでください。メトリック値を変更すると、ルーティング ループが発生する可能性があります。

MPLS レイヤ 3 VPNs のデフォルト設定

表 *2:* デフォルトの *MPLS* レイヤ *3 VPN* パラメータ

パラメータ	デフォルト
L3VPN 機能	無効
L3VPN SNMP 通知	無効

パラメータ	デフォルト
allowas-in（ハブアンドスポーク トポロジの場合）	0
disable-peer-as-check（ハブアンドスポーク トポロジの場合）	ディセーブル

『Configuring MPLS Layer 3 VPNs』

コア ネットワークの設定

MPLS レイヤ 3 VPN カスタマーのニーズの評価

MPLS レイヤ 3 VPN のカスタマーに最善のサービスを提供できるように、コア ネットワーク トポロジを識別することができます。

- ネットワークのサイズを識別します。
  - 必要となるルータとポートの数を決定するために、次の内容を識別します。
  - サポートする必要があるカスタマーの数
  - カスタマーごとに必要となる VPN の数
  - 各 VPN に存在する、仮想ルーティングおよび転送インスタンスの数
- コア ネットワークで必要なルーティング プロトコルを決定します。
- MPLS VPN ハイ アベイラビリティのサポートが必要であるかどうかを判断します。

（注） MPLS VPN ノンストップ フォワーディングおよびグレースフル リスタートは、選択ルータおよび Cisco NX-OS リリースでサポートされています。BGP および LDP のグレースフル リスタートが有効であることを確認する必要があります。

- コア ネットワークのルーティング プロトコルを設定します。
- MPLS レイヤ 3 VPN コアで BGP 負荷共有および冗長パスが必要であるかどうかを決定します。

コアにおける MPLS の設定

コアのすべてのルータでMPLSをイネーブルにするには、ラベル配布プロトコルを設定する必要があります。次のいずれかをラベル配布プロトコルとして使用できます。

• MPLS ラベル配布プロトコル（LDP）。

PE ルータおよびルート リフレクタでのマルチプロトコル BGP の設定

PE ルータおよびルート リフレクタでマルチプロトコル BGP 接続を設定できます。

始める前に

BGP および LDP のすべてのルータでグレースフル リスタートがイネーブルになっ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	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル 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	**feature bgp**
	例：
	`switch(config)# feature bgp`
	`switch(config)#`
		BGP 機能をイネーブルにします。

ステップ 3	**install feature-set mpls**
	例：
	`switch(config)# install feature-set mpls`
	`switch(config)#`
		MPLS 機能セットを有効化します。

ステップ 4	**feature-set mpls**
	例：
	`switch(config)# feature-set mpls`
	`switch(config)#`
		MPLS フィーチャ セットをイネーブルにします。

ステップ 5	**feature-set mpls l3vpn**
	例：
	`switch(config)# feature-set mpls l3vpn`
	`switch(config)#`
		MPLS レイヤ 3 VPN 機能をイネーブルにします。

ステップ 6	**router bgp** *as - number* 例： `switch(config)# router bgp 1.1`	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。as-number 引数は、ルータを他の BGP ルータに対して識別し、ルーティング情報にタグを設定する自律システムの番号を示します。AS 番号は 16 ビット整数または 32 ビット整数

	コマンドまたはアクション	目的
		にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。
ステップ 7	**router-id** *ip-address* 例： `switch(config-router)# router-id 192.0.2.255`	（任意）BGP ルータ ID を設定します。この IP アドレスによって、この BGP スピーカを特定します。このコマンドによって、BGP ネイバーセッションの自動通知およびセッション リセットが開始されます。。

ステップ 8	**neighbor** *ip-address***remote-as** *as-number*
	例：
	`switch(config-router)# neighbor 209.165.201.1 remote-as 1.1`
	`switch(config-router-neighbor)#`
		エントリを iBGP ネイバー テーブルに追加します。ip-address 引数には、ドット付き 10 進表記でネイバーの IP アドレスを指定します。

ステップ 9	**address-family { vpnv4 | vpnv6 } unicast**
	例：
	`switch(config-router-neighbor)# address-family vpnv4 unicast`
	`switch(config-router-neighbor-af)#`
		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、標準 VPNv4 または VPNv6 アドレス プレフィックスを使用する、BGP などのルーティング セッションを設定します。

ステップ 10	**send-community extended** 例： `switch(config-router-neighbor-af)# send-community extended`	コミュニティ属性が BGP ネイバーに送信されるように指定します。
ステップ 11	**show bgp { vpnv4 | vpnv6 } unicast neighbors** 例： `switch(config-router-neighbor-af)# show bgp vpnv4 unicast neighbors`	（任意）BGP ネイバーに関する情報を表示します。
ステップ 12	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

MPLS VPN カスタマーの接続

カスタマーの接続を可能にするための、PE ルータでの VRF の定義

カスタマーの接続をイネーブルにするため PE ルータに VRF を作成する必要があります。ルートターゲットを設定し、カスタマーの VPN サイトへの IP プレフィックスのインポート、および BGP ネットワークへの IP プレフィックスのエクスポートを制御します。必要に応じて、インポートまたはエクスポート ルート マップを使用して、カスタマー VPN サイトにインポートされる、または VPN サイトからエクスポートされる IP プレフィックスを、より詳細に制御できます。ルート マップを使用して、ルートのルート ターゲット拡張コミュニティ属性に基づいて、VRF でのインポートまたはエクスポートに適したルートをフィルタリングできます。たとえば、ルート マップにより、インポート ルート ターゲット リスト上のコミュニティから、選択したルートへのアクセスが拒否される場合があります。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル 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	**install feature-set mpls**
	例：
	`switch(config)# install feature-set mpls`
	`switch(config)#`
		MPLS 機能セットを有効化します。

ステップ 3	**feature-set mpls**
	例：
	`switch(config)# feature-set mpls`
	`switch(config)#`
		MPLS フィーチャ セットをイネーブルにします。

ステップ 4	**feature-set mpls l3vpn**
	例：
	`switch(config)# feature-set mpls l3vpn`
	`switch(config)#`
		MPLS レイヤ 3 VPN 機能をイネーブルにします。

ステップ 5	**vrf context** *vrf-name*
	例：
	`switch(config)# vrf context vpn1`
	`switch(config-vrf)#`
		VRF 名を割り当て、VRF コンフィギュレーション モードを開始することにより、VPN ルーティング インスタンスを定義します。vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。

	コマンドまたはアクション	目的

ステップ 6	**rd** *route-distinguisher*
	例：
	`switch(config-vrf)# rd 1.2:1`
	`switch(config-vrf)#`
		ルート識別子を設定します。
		route-distinguisher 引数によって、8 バイトの値が IPv4 プレフィックスに追加され、VPN IPv4 プレフィックスが作成されます。RD は、次のいずれかの形式で入力できます。
		• 16 ビットまたは 32 ビットの AS 番号:32 ビットの番号。1.2:3 など。
		• 32 ビットの IP アドレス:16 ビットの番号。192.0.2.1:1 など。

ステップ 7	**address-family { ipv4 | ipv6 } unicast**
	例：
	`switch(config-vrf)# address-family ipv4 unicast`
	`switch(config-vrf-af-ipv4)#`
		IPv4 アドレス ファミリ タイプを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 8	**route-target** { **import** | **export** } *route-target-ext-community* **}** 例： `switch(config-vrf-af-ipv4)# route-target import 1.0:1`	次のように VRF 用にルート ターゲット拡張コミュニティを指定します。 • import キーワードを使用すると、ターゲット VPN 拡張コミュニティからルーティング情報がインポートされます。 • export キーワードを使用すると、ルーティング情報がターゲット VPN 拡張コミュニティにエクスポートされます。 • route-target-ext-community 引数により、ルート ターゲット拡張コミュニティ属性が、インポート、またはエクスポートのルート ターゲット拡張コミュニティの VRF リストに追加されます。 route-target-ext-community 引数は、次のいずれかの形式で入力できます。 • 16 ビットまたは 32 ビットの AS 番号:32 ビットの番号。1.2:3 など。

	コマンドまたはアクション	目的
		• 32 ビットの IP アドレス:16 ビットの番号。192.0.2.1:1 など。
ステップ 9	**maximum routes** *max-routes* [**threshold** *value*] [**reinstall**] 例： `switch(config-vrf-af-ipv4)# maximum routes 10000`	（任意）VRF ルートテーブルに格納できる最大ルート数を設定します。max-routes の範囲は 1 ～ 4294967295 です。しきい値の値の範囲は 1 ～ 100 です。
ステップ 10	**import** [**vrf default** *max-prefix*] **map** *route-map* 例： `switch(config-vrf-af-ipv4)# import vrf default map vpn1-route-map`	（任意）デフォルト VRF からプレフィックスをインポートするための VRF のインポート ポリシーを次のように設定します。 • max-prefix の範囲は 1 ～ 2147483647 です。デフォルトは 1000 プレフィックスです。 • route-map 引数は VRF のインポート ルートマップとして使用されるルート マップを最大 63 文字の英数字文字列（大文字と小文字を区別）で指定します。
ステップ 11	**show vrf** *vrf-name* 例： `switch(config-vrf-af-ipv4)# show vrf vpn1`	（任意）VRF の情報を表示します。vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。
ステップ 12	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

各 VPN カスタマー用の PE ルータでの VRF インスタンスの設定

PE ルータのインターフェイスまたはサブインターフェイスに仮想ルーティングおよび転送（VRF）インスタンスを関連付けることができます。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**interface** *type number*
	例：
	`switch(config)# interface Ethernet 5/0`
	`switch(config-if)#`
		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 モードを開始する方法は次のとおりです。
		• type 引数で、設定するインターフェイスのタイプを指定します。
		• number 引数には、ポート、コネクタ、またはインターフェイス カード番号を指定します。

ステップ 3	**vrf member** *vrf-name* 例： `switch(config-if)# vrf member vpn1`	指定したインターフェイスまたはサブインターフェイスに VRF を関連付けます。vrf-name 引数は、VRF に割り当てる名前です。
ステップ 4	**show vrf** *vrf-name* **interface** 例： `switch(config-if)# show vrf vpn1 interface`	（任意）VRF に関連付けられるインターフェイスの情報を表示します。vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。
ステップ 5	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

PE ルータと CE ルータ間でのルーティング プロトコルの設定

PE ルータと CE ルータ間でスタティックまたは直接接続されたルートの設定

スタティック ルートを使用する PE-to-CE ルーティング セッション用の PE ルータを設定することができます。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

ステップ 2	**vrf context** *vrf-name*
	例：
	`switch(config)# vrf context vpn1`
	`switch(config-vrf)#`
		VRF 名を割り当て、VRF コンフィギュレーションモードを開始することにより、VPN ルーティング インスタンスを定義します。vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 3	{ **ip ipv6** } **route** *prefix nexthop* 例： `switch(config-vrf)# ip route 192.0.2.1/28 ethernet 2/1`	PE から CE への各セッション用のスタティック ルート パラメータを定義します。prefix および nexthop は次のとおりです。 • IPv4：ドット付き 10 進表記 • IPv6：16 進形式

ステップ 4	**address-family { ipv4 | ipv6 } unicast**
	例：
	`switch(config-vrf)# address-family ipv4 unicast`
	`switch(config-vrf-af)#`
		IPv4 アドレス ファミリ タイプを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 5	**feature bgp** *as - number*
	例：
	`switch(config-vrf-af)# feature bgp`
	`switch(config)#`
		BGP 機能をイネーブルにします。

ステップ 6	**router bgp** *as - number* 例： `switch(config)# router bgp 1.1`	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。as-number 引数は、ルータを他の BGP ルータに対して識別し、ルーティング情報にタグを設定する自律システムの番号を示します。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

	コマンドまたはアクション	目的

ステップ 7	**vrf** *vrf-name*
	例：
	`switch(config-router)# vrf vpn1`
	`switch(config--router-vrf)#`
		BGP プロセスを VRF に関連付けます。
		vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 8	**address-family { ipv4 | ipv6 } unicast**
	例：
	`switch(config-vrf)# address-family ipv4 unicast`
	`switch(config-vrf-af)#`
		IPv4 アドレス ファミリ タイプを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 9	**redistribute static route-map** *map-name* 例： `switch(config-router-vrf-af)# redistribute static route-map StaticMap`	スタティック ルートを BGP に再配布します。 マップ-名には最大 63 文字の英数字を使用できます。大文字と小文字は区別されます。
ステップ 10	**redistribute direct route-map** *map-name* 例： `switch(config-router-vrf-af)# redistribute direct route-map StaticMap`	直接接続されたルートを BGP に再配布します。 マップ-名には最大 63 文字の英数字を使用できます。大文字と小文字は区別されます。
ステップ 11	**show { ipv4 | ipv6 } route vrf** *vrf-name* 例： `switch(config-router-vrf-af)# show ip ipv4 route vrf vpn1`	（任意）ルートに関する情報を表示します。 vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。
ステップ 12	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

BGP を PE ルータと CE ルータ間のルーティング プロトコルに設定

eBGP を使用して PE-to-CE ルーティング セッション用の PE ルータを設定できます。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル 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	**feature bgp**
	例：
	`switch(config)# feature bgp`
	`switch(config)#`
		BGP 機能をイネーブルにします。

ステップ 3	**router bgp** *as - number*
	例：
	`switch(config)# router bgp 1.1`
	`switch(config-router)#`
		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。
		as-number 引数は、ルータを他の BGP ルータに対して識別し、転送するルーティング情報にタグを設定する自律システムの番号を示します。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

ステップ 4	**vrf** *vrf-name*
	例：
	`switch(config-router)# vrf vpn1`
	`switch(config--router-vrf)#`
		BGP プロセスを VRF に関連付けます。
		vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 5	**neighbor** *ip-address***remote-as** *as-number*
	例：
	`switch(config-router)# neighbor 209.165.201.1 remote-as 1.1`
	`switch(config-router-neighbor)#`
		エントリを iBGP ネイバーテーブルに追加します。ip-address 引数には、ドット付き 10 進表記でネイバーの IP アドレスを指定します。as-number 引数には、ネイバーが属している自律システムを指定します。

ステップ 6	**address-family { ipv4 | ipv6 } unicast**
	例：
	`switch(config-vrf)# address-family ipv4 unicast`
	`switch(config-vrf-af)#`
		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、標準 IPv4 または IPv6 アドレス プレフィックスを使用する、BGP などのルーティング セッションを設定します。

	コマンドまたはアクション	目的
ステップ 7	**show bgp { vpnv4 | vpnv6 } unicast neighbors vrf** *vrf-name* 例： `switch(config-router-neighbor-af)# show bgp vpnv4 unicast neighbors`	（任意）BGP ネイバーに関する情報を表示します。vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。
ステップ 8	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

PE ルータと CE ルータ間での RIPv2 の設定

RIP を使用して PE-to-CE ルーティング セッション用の PE ルータを設定できます。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル 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	**feature rip**
	例：
	`switch(config)# feature rip`
	`switch(config)#`
		RIP 機能を有効にします。

ステップ 3	**router rip** *instance-tag* 例： `switch(config)# router rip Test1`	RIP 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。 instance-tag には最大 20 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 4	**vrf** *vrf-name*
	例：
	`switch(config-router)# vrf vpn1`
	`switch(config--router-vrf)#`
		RIP プロセスを VRF に関連付けます。
		vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 5	**address-family ipv4 unicast** 例： `switch(config-router-vrf)# address-family ipv4 unicast`	アドレス ファミリ タイプを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
	`switch(config-router-vrf-af)#`	
ステップ 6	**redistribute** { **bgp** as | **direct** | { **egrip** | **ospf** | **rip** } *instance-tag* | static } **route-map** *map-name vrf-name* 例： `switch(config-router-vrf-af)# show ip rip vrf vpn1`	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。 as 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。instance-tag には最大 20 文字の英数字文字列を指定します。大文字と小文字は区別されます。
ステップ 7	**show ip rip vrf** *vrf-name* 例： `switch(config-router-vrf-af)# show ip rip vrf vpn1`	（任意）RIP に関する情報を表示します。 vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。
ステップ 8	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

PE ルータと CE ルータ間での OSPF の設定

OSPFv2 を使用して PE-to-CE ルーティング セッション用の PE ルータを設定できます。MPLS ネットワークの一部ではない OSPF バックドア リンクがある場合は、オプションで OSPF 模造リンクを作成できます。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル 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	**feature ospf**
	例：
	`switch(config)# feature ospf`
	`switch(config)#`
		OSPF 機能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	**router ospf** *instance-tag* 例： `switch(config)# router ospf Test1`	OSPF 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。 instance-tag には最大 20 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 4	**vrf** *vrf-name*
	例：
	`switch(config-router)# vrf vpn1`
	`switch(config--router-vrf)#`
		ルータ VRF 設定モードを開始します。
		vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 5	**area** area-id **sham-link** *source-address destination-address* 例： `switch(config-router-vrf)# area 1 sham-link 10.2.1.1 10.2.1.2`	（任意）PE インターフェイス上の模造リンクを、指定した OSPF エリア内に設定します。エンドポイントとして各ループバック インターフェイスを IP アドレスで指定します。 PE の両エンドポイントで模造リンクを設定する必要があります。
ステップ 6		

ステップ 7	**address-family { ipv4 | ipv6 } unicast**
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-vrf-af)#`
		アドレス ファミリ タイプを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 8	**redistribute** { **bgp** as | **direct** | { **egrip** | **ospf** | **rip** } *instance-tag* | static } **route-map** *map-name* 例： `switch(config-router-vrf-af)# redistribute bgp 1.0 route-map BGPMap`	BGP を EIGRP に再配布します。 BGP ネットワークの自律システム番号は、このステップで設定されます。BGP を CE サイトの EIGRP に再配布して、EIGRP 情報を伝送する BGP ルートを受け入れるようにする必要があります。また、BGP ネットワークにメトリックを指定する必要があります。 マップ-名には最大 63 文字の英数字を使用できます。大文字と小文字は区別されます。

	コマンドまたはアクション	目的

ステップ 9	**autonomous-system** *as-number*
	例：
	`switch(config-router-vrf-af)#`
	`autonomous-system 1.3`
		（任意）自律システム番号を、カスタマーサイトのこのアドレスファミリに指定します。
		as-number 引数は、ルータを他の BGP ルータに対して識別し、転送するルーティング情報にタグを設定する自律システムの番号を示します。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

ステップ 10		
ステップ 11	**show ip egrip vrf** *vrf-name* 例： `switch(config-router-vrf-af)# show ipv4 eigrp vrf vpn1`	（任意）この VRF の EIGRP に関する情報を表示します。 vrf-name には最大 32 文字の英数字文字列を指定します。大文字と-小文字は区別されます。
ステップ 12	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

PE ルータと CE ルータ間での EIGRP の設定

PE ルータと CE ルータ間で Enhanced Interior Gateway Routing Protocol（EIGRP）を使用して MPLS 対応 BGP コア ネットワーク経由で EIGRP カスタマー ネットワークがトランスペアレントに接続されるように PE ルータを設定できます。これにより、EIGRP ルートが BGP ネットワークの VPN を経由して内部 BGP（iBGP）ルートとして再配布されます。

始める前に

ネットワーク コアで BGP を設定する必要があります。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル 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	**feature egrip**
	例：
	`switch(config)# feature egrip`
	`switch(config)#`
		EGRIP 機能を有効にします。

ステップ 3	**router egrip** *instance-tag* 例： `switch(config)# router egrip Test1`	EIGRP インスタンスを設定し、ルータ コンフィギュレーション モードを開始します。 instance-tag には最大 20 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 4	**vrf** *vrf-name*
	例：
	`switch(config-router)# vrf vpn1`
	`switch(config--router-vrf)#`
		ルータ VRF 設定モードを開始します。
		vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 5	**address-family ipv4 unicast**
	例：
	`switch(config-router-vrf)# address-family ipv4 unicast`
	`switch(config-router-vrf-af)#`
		（任意）標準 IPv4 アドレス プレフィックスを使用するルーティング セッションを設定するために、アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 6	**redistribute** { **bgp** as-number**route-map** *map-name* 例： `switch(config-router-vrf-af)# show ip rip vrf vpn1`	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。 as 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。instance-tag には最大 20 文字の英数字文字列を指定します。大文字と小文字は区別されます。
ステップ 7	**show ip ospf** *instance-tag* **vrf** *vrf-name* 例： `switch(config-router-vrf-af)# show ip rip vrf vpn1`	（任意）OSPF に関する情報を表示します。
ステップ 8	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

MPLS VPN での BGP の PE-CE 再配布の設定

PE-CE プロトコルが BGP ではない場合は、MPLS レイヤ 3 VPN サービスを提供するすべての PE ルータで、PE-CE ルーティング プロトコルが配布されるように BGP を設定する必要があります。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル 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	**feature bgp**
	例：
	`switch(config)# feature bgp`
	`switch(config)#`
		BGP 機能をイネーブルにします。

ステップ 3	**router bgp** *instance-tag*
	例：
	`switch(config)# router bgp 1.1`
	`switch(config-router)#`
		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。as-number 引数は、ルータを他の BGP ルータに対して識別し、転送するルーティング情報にタグを設定する自律システムの番号を示します。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

ステップ 4	**router id** *ip-address*
	例：
	`switch(config-router)# router-id 192.0.2.255 1`
	`switch(config-router)#`
		（任意）BGP ルータ ID を設定します。この IP アドレスによって、この BGP スピーカを特定します。このコマンドによって、BGP ネイバー セッションの自動通知およびセッション リセットが開始されます。

ステップ 5	**router id** *ip-address* **remote-as** *as-number*
	例：
	`switch(config-router)# neighbor 209.165.201.1 remote-as 1.2`
	`switch(config-router-neighbor)#`
		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。ip-address 引数には、ドット付き 10 進表記でネイバーの IP アドレスを指定します。as-number 引数には、ネイバーが属している自律システムを指定します。

	コマンドまたはアクション	目的
ステップ 6	**update-source loopback [0 | 1]** 例： `switch(config-router-neighbor)# update-source loopback 0#`	BGP セッションの送信元アドレスを指定します。

ステップ 7	**address-family { ipv4 | ipv6 } unicast**
	例：
	`switch(config-router-neighbor)# address-family vpnv4`
	`switch(config-router-neighbor-af)#`
		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、標準 VPNv4 または VPNv6 アドレス プレフィックスを使用する、BGP などのルーティング セッションを設定します。unicast キーワード（任意）では、VPNv4 または VPNv6 ユニキャスト アドレス プレフィックスを指定します。

ステップ 8	**send-community extended** 例： `switch(config-router-neighbor-af)# send-community extended`	コミュニティ属性が BGP ネイバーに送信されるように指定します。

ステップ 9	**vrf** *vrf-name*
	例：
	`switch(config-router-neighbor-af)# vrf vpn1`
	`switch(config-router-vrf)#`
		ルータ VRF 設定モードを開始します。
		vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 10	**address-family { ipv4 | ipv6 } unicast**
	例：
	`switch(config-router-vrf)# address-family ipv4 unicast`
	`switch(config-router-vrf-af)#`
		標準 IPv4 または VPNv6 アドレス プレフィックスを使用するルーティング セッションを設定するために、アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 11	**redistribute** { **direct** | { **egrip** | **ospfv3** | **ospfv3** |**rip** } *instance-tag* | static } **route-map** *map-name* 例： `switch(config-router-af-vrf)# redistribute eigrp Test2 route-map EigrpMap`	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。as 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。 instance-tag には最大 20 文字の英数字文字列を指定します。大文字と小文字は区別されます。map-name には最大 63 文字の英数字文字列を指定します。大文字と小文字は区別されます。

	コマンドまたはアクション	目的
ステップ 12	**show bgp { ipv4 | ipv6 } unicast vrf** *vrf-name* 例： `switch(config-router--vrf-af)# show bgp ipv4 unicast vrf vpn1vpn1`	（任意）BGP に関する情報を表示します。vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。
ステップ 13	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

ハブ アンド スポーク トポロジの設定

ハブ PE ルータにおける VRF の設定

ハブ PE ルータ上でハブ アンド スポーク VRF を設定できます。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル 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	**install feature-set mpls**
	例：
	`switch(config)# install feature-set mpls`
	`switch(config)#`
		MPLS 機能セットを有効化します。

ステップ 3	**feature-set mpls**
	例：
	`switch(config)# feature-set mpls`
	`switch(config)#`
		MPLS フィーチャ セットをイネーブルにします。

ステップ 4	**feature-set mpls l3vpn**
	例：
	`switch(config)# feature-set mpls l3vpn`
	`switch(config)#`
		MPLS レイヤ 3 VPN 機能をイネーブルにします。

ステップ 5	**vrf context** *vrf-hub* 例：	VRF 名を割り当て、VRF コンフィギュレーション モードを開始することにより、PE ハブの VPN ルーティング イン

	コマンドまたはアクション	目的

	`switch(config)# vrf context 2hub`
	
	`switch(config-vrf)#`
		スタンスを定義します。vrf-hub 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 6	**rd** *route-distinguisher*
	例：
	`switch(config-vrf)# rd 1.2:1`
	
	`switch(config-vrf)#`
		ルート識別子を設定します。route-distinguisher 引数によって、8 バイトの値が IPv4 プレフィックスに追加され、VPN IPv4 プレフィックスが作成されます。RD は、次のいずれかの形式で入力できます。
		• 16 ビットまたは 32 ビットの AS 番号:32 ビットの番号。1.2:3 など。
		• 32 ビットの IP アドレス:16 ビットの番号。192.0.2.1:1 など。

ステップ 7	**address-family { ipv4 | ipv6 } unicast**
	例：
	`switch(config-vrf)# address-family ipv4 unicast`
	
	`switch(config-vrf-af-ipv4)#`
		IPv4 アドレス ファミリ タイプを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 8	**route-target** { **import** | **export** } route-target-ext-community **}** 例： `switch(config-vrf-af-ipv4)# route-target import 1.0:1`	次のように VRF 用にルート ターゲット拡張コミュニティを指定します。 • **import** キーワードを使用すると、ルーティング情報がターゲット VPN 拡張コミュニティからインポートされます。 • **export** キーワードを使用すると、ルーティング情報がターゲット VPN 拡張コミュニティにエクスポートされます。 • route-target-ext-community 引数により、ルート ターゲット拡張コミュニティ属性が、インポート、またはエクスポートのルート ターゲット拡張コミュニティの VRF リストに追加されます。 route-target-ext-community 引数は、次のいずれかの形式で入力できます。

	コマンドまたはアクション	目的
		• 16 ビットまたは 32 ビットの AS 番号:32 ビットの番号。1.2:3 など。 • 32 ビットの IP アドレス:16 ビットの番号。192.0.2.1:1 など。

ステップ 9	**vrf context** *vrf-spoke*
	例：
	`switch(config-vrf-af-ipv4)# vrf context 2spokes`
	`switch(config-vrf)#`
		VRF 名を割り当て、VRF コンフィギュレーションモードを開始することにより、PE スポークの VPN ルーティングインスタンスを定義します。vrf-spoke 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 10	**address-family { ipv4 | ipv6 } unicast**
	例：
	`switch(config-vrf)# address-family ipv4 unicast`
	`switch(config-vrf-af-ipv4)#`
		IPv4 アドレス ファミリ タイプを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 11	**route-target** { **import** | **export** } *route-target-ext-community* **}** 例： `switch(config-vrf-af-ipv4)# route-target export 1:100`	次のように VRF 用にルート ターゲット拡張コミュニティを指定します。 • VRF 用にルートターゲット拡張コミュニティを作成します。**import** キーワードを使用すると、ルーティング情報がターゲット VPN 拡張コミュニティからインポートされます。**export** キーワードを使用すると、ルーティング情報がターゲット VPN 拡張コミュニティにエクスポートされます。 route-target-ext-community 引数により、ルートターゲット拡張コミュニティ属性が、インポート、またはエクスポートのルートターゲット拡張コミュニティの VRF リストに追加されます。 route-target-ext-community 引数は、次のいずれかの形式で入力できます。

	コマンドまたはアクション	目的
		• 16 ビットまたは 32 ビットの AS 番号:32 ビットの番号。1.2:3 など。 • 32 ビットの IP アドレス:16 ビットの番号。192.0.2.1:1 など。
ステップ 12	**show running-config vrf** *vrf-name* 例： `switch(config-vrf-af-ipv4)# show running-config vrf 2spokes`	（オプション）VRF の実行コンフィギュレーションを表示します。 vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。 。
ステップ 13	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

ハブ PE ルータにおける eBGP の設定

eBGP を使用して PE-to-CE ハブ ルーティング セッションを設定できます。

（注） すべての CE サイトが同じ BGP AS 番号を使用している場合は、次のタスクを実行する必要があります。

- PE（ハブ）で BGP **as-override** コマンドを構成するか、受信 CE ルータで **allowas-in** コマンドを構成します。
- ある ASN から学習した BGP ルートを同じ ASN に戻してアドバタイズするには、ループバックを防止するために、PE ルータで **disable-peer-as-check** コマンドを設定します。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `switch# configure terminal switch(config)#`	グローバル 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	**feature-set mpls** 例： `switch(config)# feature-set mpls`	MPLS フィーチャ セットをイネーブルにします。
ステップ 3	**feature mpls l3vpn** 例： `switch(config)# feature mpls l3vpn`	MPLS レイヤ 3 VPN 機能をイネーブルにします。

ステップ 4	**feature bgp**
	例：
	`switch(config)# feature bgp`
	`switch(config)#`
		BGP 機能をイネーブルにします。

ステップ 5	**router bgp** *as - number*
	例：
	`switch(config)# router bgp 1.1`
	`switch(config-router)#`
		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。
		as-number 引数は、ルータを他の BGP ルータに対して識別し、転送するルーティング情報にタグを設定する自律システムの番号を示します。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

ステップ 6	**neighbor** *ip-address***remote-as** *as-number*
	例：
	`switch(config-router)# neighbor 209.165.201.1 remote-as 1.2`
	`switch(config-router-neighbor)#`
		エントリを iBGP ネイバー テーブルに追加します。
		• ip-address 引数には、ドット付き 10 進表記でネイバーの IP アドレスを指定します。
		• as-number 引数には、ネイバーが属している自律システムを指定します。

ステップ 7	**address-family { ipv4 | ipv6 } unicast**
	例：
	`switch(config-router-vrf-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		IP アドレス ファミリ タイプを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 8	**send-community extended** 例：	（任意）BGP を設定し、拡張コミュニティ リストをアドバタイズします。

	コマンドまたはアクション	目的
	`switch(config-router-neighbor-af)# send-community extended`	

ステップ 9	**vrf** *vrf-hub*
	例：
	`switch(config-router-neighbor-af)# vrf 2hub`
	`switch(config-router-vrf)#`
		VRF 設定モードを開始します。*vrf-hub* 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 10	**neighbor** *ip-address***remote-as** *as-number*
	例：
	`switch(config-router-vrf)# neighbor 33.0.0.33 1 remote-as 150`
	`switch(config-router-vrf-neighbor)#`
		BGP またはマルチプロトコル BGP ネイバー テーブルに、この VRF のためのエントリを追加します。
		• ip-address 引数には、ドット付き 10 進表記でネイバーの IP アドレスを指定します。
		• as-number 引数には、ネイバーが属している自律システムを指定します。

ステップ 11	**address-family { ipv4 | ipv6 } unicast**
	例：
	`switch(config-router-vrf-neighbor)# address-family ipv4 unicast`
	`switch(config-router--vrf-neighbor-af)#`
		IP アドレス ファミリ タイプを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 12	**as-override** 例： `switch(config-router-vrf-neighbor-af)# as-override`	（オプション）更新を送信するときに AS 番号を上書きします。すべての BGP サイトが同じ AS 番号を使用している場合、次のコマンドのいずれか： • PE（ハブ）で BGP as-override コマンドを設定します または • 受信 CE ルータで allowas-in コマンドを設定します。

ステップ 13	**vrf** *vrf-spoke*
	例：
	`switch(config-router-vrf-neighbor-af)# vrf 2spokes`
	`switch(config-router-vrf)#`
		VRF 設定モードを開始します。vrf-spoke 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。

	コマンドまたはアクション	目的

ステップ 14	**neighbor** *ip-address***remote-as** *as-number*
	例：
	`switch(config-router-vrf)# neighbor 33.0.0.33 1 remote-as 150`
	`switch(config-router-vrf-neighbor)#`
		BGP またはマルチプロトコル BGP ネイバー テーブルに、この VRF のためのエントリを追加します。
		• ip-address 引数には、ドット付き 10 進表記でネイバーの IP アドレスを指定します。
		• as-number 引数には、ネイバーが属している自律システムを指定します。

ステップ 15	**address-family { ipv4 | ipv6 } unicast**
	例：
	`switch(config-router-vrf-neighbor)# address-family ipv4 unicast`
	`switch(config-router--vrf-neighbor-af)#`
		IP アドレス ファミリ タイプを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 16	**allowas-in** [*number*] 例： `switch(config-router-vrf-neighbor-af)# allowas-in 3`	(オプション) AS パスでの AS 番号の重複を許可します。 VPN アドレス ファミリ コンフィギュレーション モード（PE スポーク）およびネイバー モード（PE ハブ）で、このパラメータを設定します。
ステップ 17	**show running-config bgp** *vrf-name* 例： `switch(config-router-vrf-neighbor-af)# show running-config bgp`	（任意）BGP の実行コンフィギュレーションを表示します。
ステップ 18	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

ハブ CE ルータにおける eBGP の設定

eBGP を使用して PE-to-CE ハブ ルーティング セッションを設定できます。

（注） すべての CE サイトが同じ BGP AS 番号を使用している場合は、次のタスクを実行する必要があります。

- PE（ハブ）で as-override コマンドを設定するか、受信 CE ルータで allowas-in コマンドを設定します。

- CE ルータで disable-peer-as-check コマンドを設定します。
- ある ASN から学習した BGP ルートを同じ ASN に戻しアドバタイズするには、ループバックを防止するために、PE ルータで disable-peer-as-check コマンドを設定します。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル 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	**feature-set mpls** 例： `switch(config)# feature-set mpls`	MPLS フィーチャ セットをイネーブルにします。
ステップ 3	**feature mpls l3vpn** 例： `switch(config)# feature mpls l3vpn`	MPLS レイヤ 3 VPN 機能をイネーブルにします。

ステップ 4	**feature bgp**
	例：
	`switch(config)# feature bgp`
	`switch(config)#`
		BGP 機能をイネーブルにします。

ステップ 5	**router bgp** *as - number*
	例：
	`switch(config)# router bgp 1.1`
	`switch(config-router)#`
		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。
		as-number 引数は、ルータを他の BGP ルータに対して識別し、転送するルーティング情報にタグを設定する自律システムの番号を示します。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

ステップ 6	**neighbor** *ip-address***remote-as** *as-number*
	例：
	`switch(config-router)# neighbor 209.165.201.1 remote-as 1.2`
	`switch(config-router-neighbor)#`
		エントリを iBGP ネイバー テーブルに追加します。
		• ip-address 引数には、ドット付き 10 進表記でネイバーの IP アドレスを指定します。

	コマンドまたはアクション	目的
		• as-number 引数には、ネイバーが属している自律システムを指定します。

ステップ 7	**address-family { ipv4 | ipv6 } unicast**
	例：
	`switch(config-router-vrf-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		IP アドレス ファミリ タイプを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 8	**send-community extended** 例： `switch(config-router-neighbor-af)# send-community extended`	（任意）BGP を設定し、拡張コミュニティ リストをアドバタイズします。

ステップ 9	**vrf** *vrf-hub*
	例：
	`switch(config-router-neighbor-af)# vrf 2hub`
	`switch(config-router-vrf)#`
		VRF 設定モードを開始します。*vrf-hub* 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 10	**neighbor** *ip-address***remote-as** *as-number*
	例：
	`switch(config-router-vrf)# neighbor 33.0.0.33 1 remote-as 150`
	`switch(config-router-vrf-neighbor)#`
		BGP またはマルチプロトコル BGP ネイバー テーブルに、この VRF のためのエントリを追加します。
		• ip-address 引数には、ドット付き 10 進表記でネイバーの IP アドレスを指定します。
		• as-number 引数には、ネイバーが属している自律システムを指定します。

ステップ 11	**address-family { ipv4 | ipv6 } unicast**
	例：
	`switch(config-router-vrf-neighbor)# address-family ipv4 unicast`
	`switch(config-router--vrf-neighbor-af)#`
		IP アドレス ファミリ タイプを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 12	**as-override** 例： `switch(config-router-vrf-neighbor-af)# as-override`	（オプション）更新を送信するときに AS 番号を上書きします。すべての BGP サイトが同じ AS 番号を使用している場合、次のコマンドのいずれか： • PE（ハブ）で BGP as-override コマンドを設定します

	コマンドまたはアクション	目的
		または • 受信 CE ルータで allowas-in コマンドを設定します。

ステップ 13	**vrf** *vrf-spoke*
	例：
	`switch(config-router-vrf-neighbor-af)# vrf 2spokes`
	`switch(config-router-vrf)#`
		VRF 設定モードを開始します。vrf-spoke 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 14	**neighbor** *ip-address***remote-as** *as-number*
	例：
	`switch(config-router-vrf)# neighbor 33.0.0.33 1 remote-as 150`
	`switch(config-router-vrf-neighbor)#`
		BGP またはマルチプロトコル BGP ネイバー テーブルに、この VRF のためのエントリを追加します。
		• ip-address 引数には、ドット付き 10 進表記でネイバーの IP アドレスを指定します。
		• as-number 引数には、ネイバーが属している自律システムを指定します。

ステップ 15	**address-family { ipv4 | ipv6 } unicast**
	例：
	`switch(config-router-vrf-neighbor)# address-family ipv4 unicast`
	`switch(config-router--vrf-neighbor-af)#`
		IP アドレス ファミリ タイプを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 16	**allowas-in** [*number*] 例： `switch(config-router-vrf-neighbor-af)# allowas-in 3`	(オプション) AS パスでの AS 番号の重複を許可します。 VPN アドレス ファミリ コンフィギュレーション モード（PE スポーク）およびネイバー モード（PE ハブ）で、このパラメータを設定します。
ステップ 17	**show running-config bgp** *vrf-name* 例： `switch(config-router-vrf-neighbor-af)# show running-config bgp`	（任意）BGP の実行コンフィギュレーションを表示します。
ステップ 18	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

スポーク PE ルータにおける VRF の設定

スポーク PE ルータ上でハブ アンド スポーク VRFs を設定できます。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル 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	**install feature-set mpls**
	例：
	`switch(config)# install feature-set mpls`
	`switch(config)#`
		MPLS 機能セットを有効化します。

ステップ 3	**feature-set mpls**
	例：
	`switch(config)# feature-set mpls`
	`switch(config)#`
		MPLS フィーチャ セットをイネーブルにします。

ステップ 4	**feature-set mpls l3vpn**
	例：
	`switch(config)# feature-set mpls l3vpn`
	`switch(config)#`
		MPLS レイヤ 3 VPN 機能をイネーブルにします。

ステップ 5	**vrf context** *vrf-spoke*
	例：
	`switch(config)# vrf context spoke`
	`switch(config-vrf)#`
		VRF 名を割り当て、VRF コンフィギュレーションモードを開始することにより、PE スポークの VPN ルーティング インスタンスを定義します。vrf-spoke 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 6	**rd** *route-distinguisher*
	例：
	`switch(config-vrf)# rd 1.101`
	`switch(config-vrf)#`
		ルート識別子を設定します。
		route-distinguisher 引数によって、8 バイトの値が IPv4 プレフィックスに追加され、VPN IPv4 プレフィックスが作成されます。RD は、次のいずれかの形式で入力できます。
		• 16 ビットまたは 32 ビットの AS 番号:32 ビットの番号。1.2:3 など。
		• 32 ビットの IP アドレス:16 ビットの番号。192.0.2.1:1 など。

	コマンドまたはアクション	目的

ステップ 7	**address-family { ipv4 | ipv6 } unicast**
	例：
	`switch(config-vrf)# address-family ipv4 unicast`
	`switch(config-vrf-af-ipv4)#`
		IPv4 アドレス ファミリ タイプを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 8	**route-target** { **import** | **export** } route-target-ext-community **}** 例： `switch(config-vrf-af-ipv4)# route-target import 1.0:1`	次のように VRF 用にルート ターゲット拡張コミュニティを指定します。 • **import** キーワードを使用すると、ルーティング情報がターゲット VPN 拡張コミュニティからインポートされます。 • **export** キーワードを使用すると、ルーティング情報がターゲット VPN 拡張コミュニティにエクスポートされます。 • route-target-ext-community 引数により、ルート ターゲット拡張コミュニティ属性が、インポート、またはエクスポートのルート ターゲット拡張コミュニティの VRF リストに追加されます。 route-target-ext-community 引数は、次のいずれかの形式で入力できます。 • 16 ビットまたは 32 ビットの AS 番号:32 ビットの番号。1.2:3 など。 • 32 ビットの IP アドレス:16 ビットの番号。192.0.2.1:1 など。
ステップ 9	**show running-config vrf** *vrf-name* 例： `switch(config-vrf-af-ipv4)# show running-config vrf 2spokes`	（オプション）VRF の実行コンフィギュレーションを表示します。 vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。 。

	コマンドまたはアクション	目的
ステップ 10	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

スポーク PE ルータにおける eBGP の設定

eBGP を使用して PEスポーク ルーティング セッションを設定できます。

（注） すべての CE サイトが同じ BGP AS 番号を使用している場合は、次のタスクを実行する必要があります。

- 受信スポーク ルータで allowas-in コマンドを構成します。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル 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	**feature-set mpls** 例： `switch(config)# feature-set mpls`	MPLS フィーチャ セットをイネーブルにします。
ステップ 3	**feature mpls l3vpn** 例： `switch(config)# feature mpls l3vpn`	MPLS レイヤ 3 VPN 機能をイネーブルにします。

ステップ 4	**feature bgp**
	例：
	`switch(config)# feature bgp`
	`switch(config)#`
		BGP 機能をイネーブルにします。

ステップ 5	**router bgp** *as - number*
	例：
	`switch(config)# router bgp 100`
	`switch(config-router)#`
		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。
		as-number 引数は、ルータを他の BGP ルータに対して識別し、転送するルーティング情報にタグを設定する自律シ

	コマンドまたはアクション	目的
		ステムの番号を示します。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

ステップ 6	**neighbor** *ip-address***remote-as** *as-number*
	例：
	`switch(config-router)# neighbor 63.63.0.63 remote-as 100`
	`switch(config-router-neighbor)#`
		エントリを iBGP ネイバー テーブルに追加します。
		• ip-address 引数には、ドット付き 10 進表記でネイバーの IP アドレスを指定します。
		• as-number 引数には、ネイバーが属している自律システムを指定します。

ステップ 7	**address-family { ipv4 | ipv6 } unicast**
	例：
	`switch(config-router-vrf-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		IPv4 または IPv6 アドレス ファミリ タイプを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 8	**allowas-in** *number* 例： `switch(config-router-vrf-neighbor-af)# allowas-in 3`	（任意）指定した回数だけ、PE ASN が設定された AS パスを許可します。 • 値の範囲は 1 ～ 10 です。 • すべての BGP サイトが同じ AS 番号を使用している場合、次のコマンドのいずれか： （注） PE（ハブ）で BGP as-override コマンドを設定するか、受信 CE ルータで allowas-in コマンドを設定します。 *as-number* 引数は、ルータを他の BGP ルータに対して識別し、転送するルーティング情報にタグを設定する自律システムの番号を示します。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

	コマンドまたはアクション	目的
ステップ 9	**send-community extended** 例： `switch(config-router-neighbor)# send-community extended`	（任意）BGP を設定し、拡張コミュニティ リストをアドバタイズします。
ステップ 10	**show running-config bgp** 例： `switch(config-router-vrf-neighbor-af)# show running-config bgp`	（任意）BGP の実行コンフィギュレーションを表示します。
ステップ 11	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

ハードウェア プロファイル コマンドを使用した MPLS の設定

リリース 7.0（3）F3（3）以降、Cisco Nexus 3600 は複数のハードウェア プロファイルをサポートします。スイッチでハードウェア プロファイル コンフィギュレーション コマンドを使用して、MPLS および/または VXLAN を設定できます。ハードウェア プロファイル コンフィギュレーション コマンドは、スイッチで使用可能な適切なコンフィギュレーション ファイルを呼び出します。VXLAN はデフォルトで有効になっています。

始める前に

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

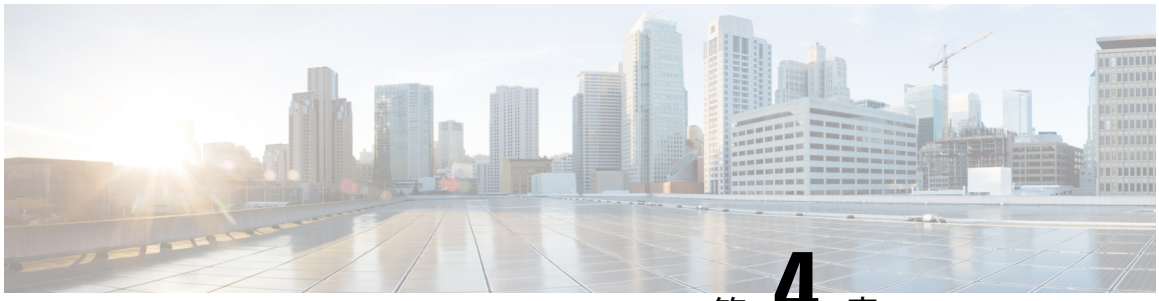
ステップ 2	**feature bgp**
	例：
	`switch(config)# feature bgp`
	`switch(config)#`
		BGP 機能をイネーブルにします。

ステップ 3	**hardware profile** [**vxlan** | **mpls**] **module all** 例： `switch(config)# hardware profile mpls module all`	すべてのスイッチ モジュールで MPLS を有効にします。。

	コマンドまたはアクション	目的

ステップ 4	**show hardware profile module** [**all** | *number*]
	例：
	`switch(config)# show hardware profile module all`
	`switch(config)#`
		すべてのモジュールまたは特定のモジュールのハードウェア プロファイルを表示します。

ステップ 5	**show module internal sw info** | [**i** | **mpls**] 例： `switch(config)# show module internal sw info`	スイッチのソフトウェア情報を表示します。
ステップ 6	**show running configuration** | [**i** | **mpls**] 例： `switch(config)# show module internal sw info`	実行設定を表示します。



第 4 章

MPLS レイヤ 3 VPN ラベル割り当ての設定

この章では、Cisco Nexus 3600シリーズ スイッチでマルチプロトコル ラベル スイッチング（MPLS）レイヤ 3 仮想プライベート ネットワーク（L3VPN）のラベル割り当てを設定する方法について説明します。

MPLS L3VPN ラベル割り当ての概要

MPLS プロバイダー エッジ（PE）ルータには、ローカル ルートとリモート ルートの両方が格納されており、各ルートに対するラベル エントリも含まれています。デフォルトでは、Cisco NX-OS はプレフィックス単位のラベル割り当てを使用します。プレフィックスごとに 1 つのラベルが割り当てられます。分散プラットフォームでは、プレフィックス単位のラベルによりメモリが消費されます。多数の VPN ルーティングおよび転送（VRF）インスタンスおよびルートが存在する場合、プレフィックス単位のラベルにより消費されるメモリ量が問題となります。

VRF 全体でローカル ルートに単一の VPN ラベルがアドバタイズされるように、VRF 単位のラベル割り当てをイネーブルにすることができます。ルータは、VRF デコードおよび IP ベースのルックアップに新しい VPN ラベルを使用して、PE またはカスタマーエッジ（CE）インターフェイスのパケットの転送先を学習します。

ボーダー ゲートウェイ プロトコル（BGP）レイヤ 3 VPN ルートごとに異なるラベル割り当てモードをイネーブルにすることが可能です。これにより、異なる要件を満たし、拡張性とパフォーマンスの間のトレードオフを実現することができます。ラベルはすべてグローバル ラベ

ル スペース内で割り当てられます。Cisco NX-OS は、次のラベル割り当てモードをサポートしています。

- プレフィックス単位：各 VPN プレフィックスに 1 つのラベルが割り当てられます。ラベル転送テーブルに基づき、リモート PE から着信する VPN パケットは接続された CE に直接転送できます。CE にはプレフィックスがアドバタイズされます。しかし、このモードでは多くのラベルが使用されます。このモードが利用可能なのは、PE から CE に送信される VPN パケットがラベル スイッチングされる場合のみです。これがデフォルトのラベル割り当てモードになります。
- VRF 単位：VRF のローカル VPN ルートすべてに単一のラベルが割り当てられます。このモードでは、VPN ラベルが出力 PE で削除されると、VRF の転送テーブルで IPv4 ルックアップまたは IPv6 ルックアップが必要になります。このモードは、ラベル スペースと BGP アドバタイズメントに関して最も効率的であり、ルックアップによってパフォーマンスが低下することはありません。Cisco NX-OS では、IPv4 プレフィックスおよび IPv6 プレフィックスの両方で同じ VRF 単位のラベルを使用します。

（注） EIBGP ロード バランシングでは、VRF 単位のラベル モードを使用する VRF はサポートされません。

- 集約ラベル：BGP は、集約プレフィックスのローカル ラベルを割り当てたり、アドバタイズしたりできます。転送時には、VRF 単位の場合と同じように IPv4 ルックアップまたは IPv6 ルックアップが必要になります。単一の VRF 単位のラベルは、ルックアップが必要なすべてのプレフィックスに割り当てられ、使用されます。
- VRF 接続されたルート：直接接続されたルートが再配布およびエクスポートされるときに、各ルートに集約ラベルが割り当られます。コアから送信されるパケットは非カプセル化され、VRF の IPv4 テーブルまたは IPv6 テーブルで、ローカル ルータへのパケットか、別のルータまたは直接接続されたホストへのパケットかを判断するためにルックアップが行われます。単一の VRF 単位のラベルは、これらすべてのルートに割り当てられます。
- ラベルの抑制：ローカル ラベルがこれ以上プレフィックスに関連付けられないときは、他の PE に送信されるアップデートの時間を確保するために、ローカル ラベルがすぐに解放されない場合があります。ラベルごとに 10 分の抑制タイマーが作動します。この間、ラベルをプレフィックスに対して再利用することができます。タイマーが切れると、BGP はラベルを解放します。

IPv6 ラベルの割り当て

IPv6 プレフィックスは、割り当てられたラベルとともに、ラベル付きユニキャスト アドレス ファミリがイネーブルになっている iBGP ピアにアドバタイズされます。着信した eBGP ネクスト ホップはこのピアに伝播されず、代わりにローカル IPv4 セッションのアドレスが IPv4 射影 IPv6 ネクスト ホップとして送信されます。リモート ピアは、コア ネットワーク内の 1 つまたは複数の IPv4 MPLS LSP を介してこのネクスト ホップを解決します。

ルート リフレクタを使用して、PE 間のラベル付き 6PE プレフィックスをアドバタイズできます。このとき、ルート リフレクタとこれらすべてのピアの間で、ラベル付きユニキャスト アドレス ファミリをイネーブルにする必要があります。ルート リフレクタは転送パスにある必要はなく、受信したネクスト ホップをそのまま iBGP ピアおよびルート リフレクタ クライアントに伝播します。

（注） 6PE は、6VPE と同様に、プレフィックス単位および VRF 単位のラベル割り当てモードの両方をサポートします。

VRF 単位のラベル割り当てモード

VRF 単位のラベル割り当てを設定する場合、次の条件が適用されます。

- VRF は、すべてのローカル ルートに対して 1 つのラベルを使用します。
- VRF 単位のラベル割り当てをイネーブルにした場合、すべての既存の VRF 単位の集約ラベルが使用されます。VRF 単位の集約ラベルが存在しない場合は、ソフトウェアによって新規の VRF 単位のラベルが作成されます。

  VRF 単位のラベルの割り当てをディセーブルにした場合、デフォルトのプレフィックス単位のラベリング設定に戻るため、CE がデータを失うことはありません。
- VRF 単位ラベルのフォワーディング エントリは、VRF、BGP、またはアドレス ファミリ設定が削除された場合にのみ、削除されます。

ラベル付きユニキャスト パスとラベルなしユニキャスト パスについて

後続アドレス ファミリ識別子（SAFI）は、BGP ルートの指標です。例 1 はラベルなしルート、4 はラベル付きルートです。

- IPv4 のラベルなしユニキャスト（U）は SAFI 1 です。
- IPv4 のラベル付きユニキャスト（LU）は SAFI 4 です。
- IPv6 のラベルなしユニキャスト（U）は、AFI 2 および SAFI 1 です。
- IPv6 のラベル付きユニキャスト（LU）は、AFI 2 および SAFI 4 です。

Cisco NX-OS リリース 9.2(2) は、1 つの BGP セッションで、IPv4 と IPv6 のラベルなしおよびラベル付きユニキャストの両方をサポートします。この動作は、同じセッションで SAFI-1 と SAFI-4 の一方または両方が有効になっているかどうかに関係なく同じです。

この動作は、すべての eBGP、iBGP、および再配布パスと、eBGP および iBGP ネイバーに適用されます。

MPLS L3VPN ラベル割り当ての前提条件

L3VPN ラベル割り当てには、次の前提条件があります。

- ネットワークに MPLS、および LDP を設定する必要があります。PE ルータを含む、コア内のすべてのルータは、MPLS 転送をサポートできる必要があります。
- MPLS の正しいライセンスおよび MPLS で使用する他の機能をインストールすることが必要です。
- VRF 単位のラベル割り当てモードを設定する前に、外部/内部ボーダー ゲートウェイ プロトコル（BGP）マルチパス機能がイネーブルになっている場合は、ディセーブルにします。
- VRF ラベル単位での 6VPE を設定する前に、IPv6 アドレス ファミリをその VRF で設定する必要があります。

MPLS L3VPN ラベル割り当てに関する注意事項と制限事項

L3VPN ラベル割り当て設定時の注意事項と制限事項は次のとおりです。

- レイヤ 3 VPN ラベル割り当ては、Cisco Nexus 3600 プラットフォーム スイッチでもサポートされています。
- VRF 単位のラベル割り当てをイネーブルにすると、BGP 再コンバージェンスが発生します。これにより、MPLS VPN コアから発信されるトラフィックでのデータ損失につながる場合があります。

（注） スケジュールされた MPLS メンテナンスの時間帯に VRF 単位のラベル割り当てをイネーブルにすることにより、ネットワークの中断を最小限に抑えることができます。また、可能であれば、現在アクティブなルータでこの機能をイネーブルにすることは避けてください。

- 集約ラベルと VRF ごとのラベルは、仮想デバイス コンテキスト（VDC）全体でグローバルであり、独立した専用のラベル範囲を持っています。
- プレフィックス単位のラベル割り当てのための集約プレフィックスは、特定の VRF で同じラベルを共有します。

MPLS L3VPN ラベル割り当てのデフォルト設定

表 *3:* デフォルトの *L3VPN* ラベル割り当てパラメータ

パラメータ	デフォルト
L3VPN 機能	無効
ラベル割り当てモード	プレフィックス単位

MPLS L3VPN ラベル割り当ての設定

VRF 単位での L3VPN ラベル割り当てモードの設定

レイヤ 3 VPN の VRF 単位での L3VPN ラベル割り当てモードを設定できます。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル 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	**feature bgp**
	例：
	`switch(config)# feature bgp`
	`switch(config)#`
		BGP 機能をイネーブルにします。

ステップ 3	**feature-set mpls**
	例：
	`switch(config)# feature-set mpls`
	`switch(config)#`
		MPLS フィーチャ セットをイネーブルにします。

ステップ 4	**feature-set mpls l3vpn**
	例：
	`switch(config)# feature-set mpls l3vpn`
	`switch(config)#`
		MPLS レイヤ 3 VPN 機能をイネーブルにします。

ステップ 5	**router bgp** *as - number* 例： `switch(config)# router bgp 1.1`	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。as-number 引数は、ルータを他の BGP ルータに対して識別し、

	コマンドまたはアクション	目的
		ルーティング情報にタグを設定する自律システムの番号を示します。AS番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。
ステップ 6	**vrf** *vrf-name* 例： `switch(config-router)# vrf vpn1`	ルータ VRF 設定モードを開始します。vrf-name には最大 32 文字の英数字文字列を指定します。大文字と-小文字は区別されます。
ステップ 7	**address-family { ipv4 | ipv6 } unicast | multicast }** 例： `switch(config-router-vrf)# address-family ipv6 unicast`	IP アドレス ファミリ タイプを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。
ステップ 8	**label-allocation-mode per-vrf** 例： `switch(config-router-vrf-af)# label-allocation-mode per-vrf`	VRF 単位でラベルを割り当てます。
ステップ 9	**show bgp l3vpn detail vrf** *vrf-name* 例： `switch(config-router-vrf-af)# show bgp l3vpn detail vrf vpn1`	（任意）この VRF の BGP でのレイヤ 3 VPN の設定に関する情報を表示します。vrf-name には最大 32 文字の英数字文字列を指定します。大文字と-小文字は区別されます。
ステップ 10	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

デフォルト VRF での IPv6 プレフィックスへのラベル割り当て

IPv4 MPLS 上で IPv6 を実行している場合、デフォルト VRF で IPv6 プレフィックスにラベルを割り当てることができます。

（注） デフォルトでは、デフォルト VRF で IPv6 プレフィックスにラベルは割り当てられません。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル 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	**feature bgp**
	例：
	`switch(config)# feature bgp`
	`switch(config)#`
		BGP 機能をイネーブルにします。

ステップ 3	**feature-set mpls**
	例：
	`switch(config)# feature-set mpls`
	`switch(config)#`
		MPLS フィーチャ セットをイネーブルにします。

ステップ 4	**feature-set mpls l3vpn**
	例：
	`switch(config)# feature-set mpls l3vpn`
	`switch(config)#`
		MPLS レイヤ 3 VPN 機能をイネーブルにします。

ステップ 5	**router bgp** *as - number* 例： `switch(config)# router bgp 1.1`	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。as-number 引数は、ルータを他の BGP ルータに対して識別し、ルーティング情報にタグを設定する自律システムの番号を示します。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

ステップ 6	**address-family { ipv4 | ipv6 } unicast | multicast }**
	例：
	`switch(config-router-vrf)#`
	`address-family ipv6 unicast`
		IP アドレス ファミリ タイプを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 7	**allocate-label { all | route-map** *route-map* **}**
	例：
	`switch(config-router-af)#`
	`allocate-label all`
		デフォルト VRF で IPv6 プレフィックスにラベルを割り当てます。
		• **all** キーワードを使用すると、すべての IPv6 プレフィックスにラベルが割り当てられます。

	コマンドまたはアクション	目的
		• **route-map** キーワードを使用すると、特定のルート マップで、マッチする IPv6 プレフィックスにラベルが割り当てられます。route-map には最大 63 文字の英数字文字列を指定します。大文字と小文字は区別されます。
ステップ 8	**show running-config bgp** 例： `switch(config-router-af)# show running-config bgp`	（任意）BGP の設定に関する情報を表示します。
ステップ 9	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

iBGP ネイバーへの IPv4 MPLS コア ネットワーク（6PE）を介した IPv6 内の MPLS ラベル送信の有効化

6PE は、ラベル付きユニキャスト アドレスファミリがイネーブルになっている iBGP ピアへの割り当てラベルを持つ iPv4 ベース MPLS ネットワーク上のグローバル VRF 内で、IPv6 プレフィックスをアドバタイズします。6PE では、コアに面したインターフェイスで LDP が有効になっていて、IPv4 ベースの MPLS ネットワーク経由で IPv6 トラフィックが転送され、BGP の下で「address-family ipv6 labeled-unicast」により PE 間で IPv6 プレフィックスのラベルを交換される必要があります。

（注） **address-family ipv6 labeled-unicast** コマンドは iBGP ネイバーでのみサポートされます。このコマンドを **address-family ipv6 unicast** コマンドとともに使用することはできません。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `switch# configure terminal switch(config)#`	グローバル 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	**feature bgp**
	例：
	`switch(config)# feature bgp`
	`switch(config)#`
		BGP 機能をイネーブルにします。

ステップ 3	**feature-set mpls**
	例：
	`switch(config)# feature-set mpls`
	`switch(config)#`
		MPLS フィーチャ セットをイネーブルにします。

ステップ 4	**feature-set mpls l3vpn**
	例：
	`switch(config)# feature-set mpls l3vpn`
	`switch(config)#`
		MPLS レイヤ 3 VPN 機能をイネーブルにします。

ステップ 5	**router bgp** *as - number* 例： `switch(config)# router bgp 1.1`	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。as-number 引数は、ルータを他の BGP ルータに対して識別し、ルーティング情報にタグを設定する自律システムの番号を示します。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

ステップ 6	**neighbor** *ip-address*
	例：
	`switch(config-router)# neighbor 209.165.201.1`
	`switch(config-router-neighbor)#`
		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。ip-address 引数には、ドット付き 10 進表記でネイバーの IP アドレスを指定します。

ステップ 7	**address-family ipv6 labeled-unicast**
	例：
	`switch(config-router-neighbor)# address-family ipv6 labeled-unicast`
	`switch(config-router-neighbor-af)#`
		IPv6 ラベル付きユニキャスト アドレス プレフィックスを指定します。このコマンドは、iBGP ネイバーによってのみ受け入れられます。

ステップ 8	**show running-config bgp** 例： `switch(config-router-af)# show running-config bgp`	（任意）BGP の設定に関する情報を表示します。

	コマンドまたはアクション	目的
ステップ 9	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

例

次のタスク

-

アドバタイズと撤回のルール

次の表は、さまざまなシナリオでのアドバタイズと撤回の動作を示しています。

表 *4:* アドバタイズと撤回のルール

大文字/小文字	Bestpath/ Addpath のタイプ	ローカル ラベルが存在しますか?	NHS または NHU	Update-group SAFI	アドバタイズまたは撤回？	コメント
1	ラベルのないパス。たとえば、RX ラベルがない。	はい	NHS	SAFI-1	デフォルトでアトバタイズ。	現在のデフォルトの動作は、アドバタイズです。理想的なデフォルトの動作は、下位互換性を維持するために撤回である必要があります。ネイバーに SAFI 1 と SAFI 4 の両方が設定されている場合、**advertise local-labeled-route** CLI コマンドは、ピアへの SAFI 4 パスのみをアドバタイズする決定論的な方法を提供します。この機能は、ラベル付きパスの優先順位を強制する方法を提供します。

大文字/小文字	Bestpath/ Addpath のタイプ	ローカル ラベルが存在しますか?	NHS または NHU	Update-group SAFI	アドバタイズまたは撤回？	コメント
2				SAFI-4	アドバタイズ	IPv4/IPv6 再配布ルートと 6PE: 常に暗黙の NHS。
3			NHU	SAFI-1	アドバタイズ	
4				SAFI-4	出金	IPv4/IPv6 再配布ルートと 6PE: NHU は無視されます。常に暗黙の NHS。現在、NXOS BGP は暗黙の null でアドバタイズしています。
5		いいえ	NHS	SAFI-1	アドバタイズ	
6				SAFI-4	出金	
7			NHU	SAFI-1	アドバタイズ	
8				SAFI-4	出金	

大文字/小文字	Bestpath/Addpath のタイプ	ローカル ラベルが存在しますか?	NHS または NHU	Update-group SAFI	アドバタイズまたは撤回？	コメント
9	ラベル付きのパス。たとえば、RX ラベルがある。	はい	NHS	SAFI-1	デフォルトでアトバタイズ。 NbrKnob で撤回。	現在のデフォルトの動作は、アドバタイズです。理想的なデフォルトの動作は、下位互換性を維持するために撤回である必要があります。
10				SAFI-4	アドバタイズ	
11			NHU	SAFI-1	出金	next-hop-self 値を持つ IBGP-IBGP リフレクトルートについては、現在、期待どおりに撤回しています。 next-hop-unchanged 値を持つ IBGP-EBGP ルートの場合、NXOS BGP は現在、ラベルなしでアドバタイズしています。
12				SAFI-4	アドバタイズ	

大文字/小文字	Bestpath/ Addpath のタイプ	ローカル ラベルが存在しますか?	NHS または NHU	Update-group SAFI	アドバタイズまたは撤回？	コメント
13		いいえ	NHS	SAFI-1	アドバタイズ	
14				SAFI-4	出金	
15			NHU	SAFI-1	出金	IBGP-IBGP リフレクト ルートについては、撤回します。IBGP-EBGP ルートについては、アドバタイズしています。
				SAFI-4	アドバタイズ	IBGP-IBGP リフレクト ルートについては、撤回します。IBGP-EBGP ルートについては、アドバタイズしています。

ローカル ラベル割り当ての有効化

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル 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	**feature bgp**
	例：
	`switch(config)# feature bgp`
	`switch(config)#`
		BGP 機能をイネーブルにします。

ステップ 3	**feature-set mpls**
	例：
	`switch(config)# feature-set mpls`
	`switch(config)#`
		MPLS フィーチャ セットをイネーブルにします。

ステップ 4	**router bgp** *as - number* 例： `switch(config)# router bgp 1.1`	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。as-number 引数は、ルータを他の BGP ルータに対して識別し、ルーティング情報にタグを設定する自律システムの番号を示します。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

ステップ 5	**address-family { ipv4 | ipv6 } unicast | multicast }**
	例：
	`switch(config-router-vrf)#`
	`address-family ipv4 unicast`
		IP アドレス ファミリ タイプを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 6	**allocate-label { all | route-map** *route-map* **}**
	例：
	`switch(config-router-af)#`
	`allocate-label all`
		デフォルト VRF で IPv6 プレフィックスにラベルを割り当てます。
		• **all** キーワードを使用すると、すべての IPv6 プレフィックスにラベルが割り当てられます。
		• **route-map** キーワードを使用すると、特定のルートマップで、マッチする IPv6 プレフィックスにラベルが割り当てられます。route-map には最大 63 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 7	**neighbor** *ip-address* 例：	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。ip-address 引数に

	コマンドまたはアクション	目的

	`switch(config-router)# neighbor 209.165.201.1`
	
	`switch(config-router-neighbor)#`
		は、ドット付き 10 進表記でネイバーの IP アドレスを指定します。

ステップ 8	**[no] advertise local-labeled-route** 例： `switch(config-router-neighbor)# advertise local-labeled-route`	IPv4 または IPv6 ユニキャスト SAFI（SAFI-1）を介して、BGP ネイバーに、ローカル ラベルを持つ IPv4 または IPv6 ルートをアドバタイズするかどうかを示します。デフォルトは有効になっているため、BGP ネイバーにアドバタイズできます。
ステップ 9	**address-family { ipv4 | ipv6 } unicast | multicast }** 例： `switch(config-router-vrf)# address-family ipv6 unicast`	IP アドレス ファミリ タイプを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。
ステップ 10	**[no] advertise local-labeled-route** 例： `switch(config-router-neighbor)# advertise local-labeled-route`	IPv4 または IPv6 ユニキャスト SAFI（SAFI-1）を介して、BGP ネイバーに、ローカル ラベルを持つ IPv4 または IPv6 ルートをアドバタイズするかどうかを示します。デフォルトは有効になっているため、BGP ネイバーにアドバタイズできます。
ステップ 11	**route-map label_routemap permit 10** 例： `switch(config-router-vrf)# route-map label_routemap permit 10`	
ステップ 12	**show running-config bgp** 例： `switch(config-router-af)# show running-config bgp`	（任意）BGP の設定に関する情報を表示します。
ステップ 13	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

MPLS L3VPN ラベル割り当ての設定の確認

L3VPN ラベル割り当ての設定を表示するには、次のいずれかの作業を行います。

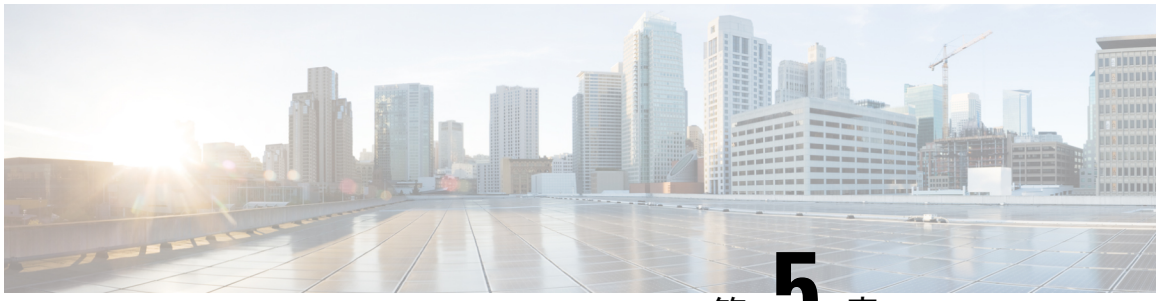
表 *5 : MPLS L3VPN* ラベル割り当ての設定の確認

コマンド	目的
show bgp l3vpn [**detail**] [**vrf** *v rf-name*]	VRF での BGP のレイヤ 3 VPN 情報を表示します。
show bgp vpnv4 unicast labels [**vrf** *v rf-name*]	BGP のラベル情報を表示します。
show ip route [**vrf** *v rf-name*]	ルートのラベル情報を表示します。

MPLS L3VPN ラベル割り当ての設定例

次に、IPv4 MPLS ネットワークの VRF 単位のラベル割り当てを設定する例を示します。

```
PE1
-----
vrf context vpn1
rd 100:1
address-family ipv4 unicast
route-target export 200:1
router bgp 100
neighbor 10.1.1.2 remote-as 100
address-family vpnv4 unicast
send-community extended
update-source loopback10
vrf vpn1
address-family ipv4 unicast
label-allocation-mode per-vrf
neighbor 36.0.0.2 remote-as 300
address-family ipv4 unicast
```



第 5 章

MPLS レイヤ 3 VPN ロード バランシングの設定

この章では、Cisco NX-OS デバイスでマルチプロトコル ラベル スイッチング（MPLS）レイヤ 3 バーチャル プライベート ネットワーク（VPN）のロード バランシングを設定する方法について説明します。

MPLS レイヤ 3 VPN ロード バランシングに関する情報

ロードバランシングは、個々のルーターに過度の負荷がかからないようにトラフィックを分散します。IMPLS レイヤ 3 ネットワークでは、ボーダー ゲートウェイ プロトコル（BGP）を使用することにより、ロードバランシングを実現します。ルーティング テーブルに複数の iBGP パスがインストールされている場合、ルート リフレクタは 1 つのパス（ネクスト ホップ）だけをアドバタイズします。ルータがルート リフレクタの背後にある場合、マルチホーム サイトに接続されているすべてのルートは、別のルート識別子が仮想ルーティングおよび転送インスタンス（VRF）ごとに設定されていない限り、アドバタイズされません。（ルート リフレクタは学習したルートをネイバーに渡すことで、すべての iBGP ピアをフルメッシュにしなくてもすむようにします）。

iBGP ロード バランシング

ローカル ポリシーが設定されていない BGP 対応ルーターが、同じ宛先の内部 BGP (iBGP) から複数のネットワーク層到達可能性情報 (NLRI) を受信すると、ルーターは 1 つの iBGP パスを最適パスとして選択し、その IP ルーティング テーブルに最適パスをインストールします。

iBGP ロード バランシングにより、BGP 対応ルータは、宛先への最適パスとして複数の iBGP パスを選択し、IP ルーティング テーブルに複数の最適パスをインストールできます。

eBGP ロード バランシング

ルータは、1 つのプレフィックスに対し、ネイバー自律システムから 2 つの同一 eBGP パスを学習した場合、ルート ID が小さいパスを最良パスとして選択します。この最良パスが IP ルーティング テーブルにインストールされます。eBGP ロード バランシングをイネーブルにすると、ネイバー自律システムから複数の eBGP パスを学習したときに、最良パスを 1 つ選択するのではなく、複数のパスを IP ルーティング テーブルにインストールします。

パケット スイッチング中には、スイッチング モードに応じて、複数のパス間でパケット単位または宛先単位のロード バランシングが実行されます。

Layer 3 VPN ロード バランシング

eBGP および iBGP の両方に対するレイヤ 3 VPN ロード バランシング機能を使用すると、マルチホーム自律システムおよびプロバイダー エッジ（PE）ルータで、外部 eBGP（eBGP）および iBGP マルチパスの両方にわたってトラフィックを配信するように設定できます。

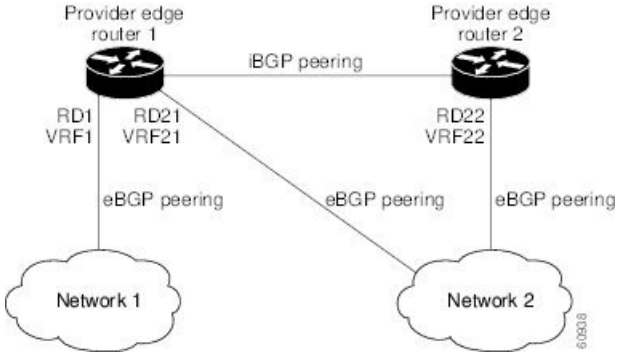
レイヤ 3 VPN ロード バランシングは、PE ルーターと VPN で IPv4 と IPv6 をサポートします。

BGP は、許可される最大数のマルチパスまでインストールします。BGP は、最良パス アルゴリズムを使用して、最良パスとして 1 つのパスを選択し、その最良パスをルーティング情報ベース（RIB）に挿入し、最良パスを BGP ピアにアドバタイズします。ルータは他のパスを RIB に挿入できますが、1 つのパスだけを最適なパスとして選択します。

レイヤ 3 VPN は、パケットごと、または送信元または宛先のペアごとにロード バランシングを行います。ロードバランシングを有効にするには、eBGP パスと iBGP パスの両方をインポートする VPN ルーティングおよび転送インスタンス（VRF）を含むレイヤ 3 VPN でルータを構成します。VRF ごとに個別にパスの数を設定できます。

次の図は、BGP を使用する MPLS プロバイダー ネットワークを示しています。この図では、2 つのリモート ネットワークが PE1 と PE2 に接続されており、どちらも VPN ユニキャスト iBGP ピアリング用に設定されています。ネットワーク 2 は、PE1 および PE2 に接続されているマルチホーム ネットワークです。またネットワーク 2 は、ネットワーク 1 とのエクストラネット VPN サービスが設定されています。ネットワーク 1 とネットワーク 2 は両方とも、PE ルータを使用した eBGP ピアリングが設定されています。

図 *4 : BGP* を使用したプロバイダー *MPLS* ネットワーク



PE1 を設定して、iBGP パスと eBGP パスの両方をマルチパスとして選択し、これらのパスをネットワーク 1 の VPN ルーティングおよび転送インスタンス（VRF）にインポートして、ロード バランシングを実行できます。

トラフィックは次のように分散されます。

- ネットワーク 2 から PE1 および PE2 に送信される IP トラフィックは、IP トラフィックとして eBGP パスを経由して送信されます。
- PE1 から PE2 に送信される IP トラフィックは、MPLS トラフィックとして iBGP パスを介して送信されます。
- eBGP パスを介して送信されるトラフィックは、IP トラフィックとして送信されます。

ネットワーク 2 からアドバタイズされているすべてのプレフィックスは、ルート識別子（RD）21 と RD22 を経由し、PE1 によって受信されます。

- RD21 を経由するアドバタイズメントは IP パケットに伝送されます。
- RD22 を経由するアドバタイズメントは MPLS パケットに伝送されます。

ルータは両方のパスを VRF1 のマルチパスとして選択でき、これらのパスを VRF1 RIB にインストールできます。

ルート リフレクタを使用したレイヤ 3 VPN ロード バランシング

ルート リフレクタは、PE ルータでのセッション数を減らし、レイヤ 3 VPN ネットワークの拡張性を向上させます。ルート リフレクタは、PE ルータとピアリングするために、受信したすべての VPN ルートを保持します。異なる PE では、異なるルート ターゲット タグ付き VPNv4 および VPNv6 ルートが必要になる場合があります。ルート リフレクタはまた、VRF 設定が変更されたときに特定のルート ターゲットのリフレッシュを PE に送信する必要がある場合があります。すべてのルートを保存すると、ルート リフレクタのスケーラビリティ要件が増大します。ルート リフレクタはルート ターゲット コミュニティの定義済みのセットを持つルートだけを保持するように設定できます。

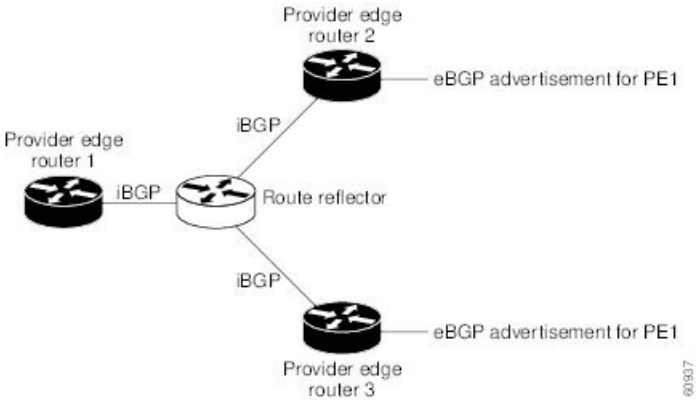
さまざまな VPN セットにサービスを提供するようにルート リフレクタを設定し、PE で設定された VRF にサービスを提供するすべてのルート リフレクタとピアリングするように PE を設

定できます。PE が、まだルートを保持していないルート ターゲットを使用して、新しい VRF を設定すると、この PE はルート リフレクタに対してルート更新要求を発行し、関連する VPN ルートを取得します。

下の図に、3 つの PE ルータと 1 つのルート リフレクタを含むトポロジを示します。これらすべてには、iBGP ピアリングが設定されています。PE 2 と PE 3 はそれぞれ、PE 1 への等プリファレンス eBGP パスをアドバタイズします。デフォルトでは、ルート リフレクタは 1 つのパスだけを選択し、PE 1 にアドバタイズします。

（注） ルート リフレクタは転送パスに存在する必要はありませんが、マルチホームの VPN サイトに固有のルート識別子（RD）を設定する必要があります。

図 *5:* ルート リフレクタを配置したトポロジ



PE1 への等価プリファレンス パスのすべてがルート リフレクタを経由してアドバタイズされるためには、異なる RD を使用して各 VRF を設定する必要があります。ルート リフレクタによって受信されたプレフィックスは別々に認識され、PE 1 にアドバタイズされます。

レイヤ 2 ロード バランシングの併用

レイヤ 2 VPN で必要とされるロード バランシング方式は、レイヤ 3 VPN で使用される方式とは異なります。レイヤ 3 VPN およびレイヤ 2 VPN の転送は、2 つの異なるタイプの隣接関係を使用して個別に実行されます。レイヤ 2 VPN で別のロード バランシング方式を使用しても、転送は影響を受けません。

（注） レイヤ 2 VPN の場合、入力 PE ではロード バランシングがサポートされません。

BGP VPNv4 マルチパス

BGP VPNv4 マルチパス機能は、自律システム ボーダー ルーター（ASBR）からマルチプロトコル ラベル スイッチング（MPLS）クラウド ネットワーク内のプロバイダー エッジ（PE）デ

バイスに向かって流れるトラフィックの等コストマルチパス（ECMP）を実現するのに役立ちます。プレフィックスとMPLSラベルの数が少なくなります。この機能は、eBGPパスとiBGPパスの両方にマルチパスの最大数を設定します。この機能は、MPLS トポロジの PE デバイスおよびルート リフレクタで設定できます。

デュアルホームのカスタマー エッジ (CE) デバイスが 2 つの PE デバイスに接続されており、ASBR-2 から CE デバイスへのトラフィック フローで両方の PE デバイスを利用する必要があるシナリオを考えてみます。

現在、次の図に示すように、各PEの仮想ルーティングおよび転送（VRF）機能は、個別のルート識別子（RD）を使用して構成されています。CEデバイスは、BGP IPv4プレフィックスを生成します。PE デバイスは 2 つの個別の RD で構成され、CE デバイスによって送信される BGP IPv4 プレフィックスに対して 2 つの異なる VPN-IPv4 プレフィックスを生成します。ASBR-1 は両方の VPN-IPv4 プレフィックスを受信し、ルーティング テーブルに追加します。ASBR-1 は、Inter-AS オプション B ラベル、Inlabel L1 および Inlabel L2 を両方の VPN ルートに割り当て、両方の VPN ルートを ASBR-2 にアドバタイズします。両方の PE デバイスを使用してトラフィック フローを維持するには、ASBR-1 で 2 つの Inter-AS オプション B ラベルと 2 つのプレフィックスを利用する必要があります。これにより、サポートできるスケールは制限されます。

図 *6* : 個別のルート識別子を使用して構成された各 *PE* での仮想ルーティングおよび転送（*VRF*）

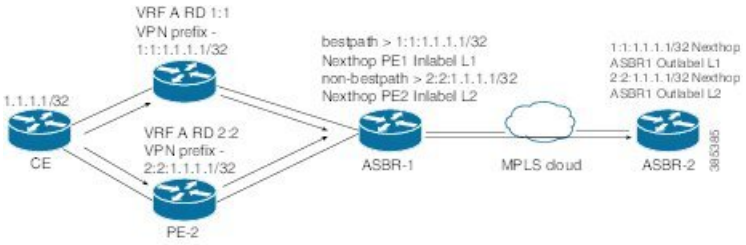
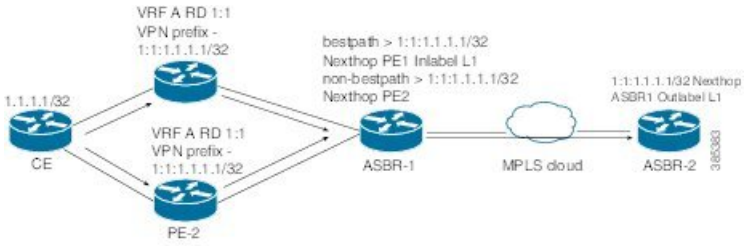


図 22-4 に示すように、BGP VPN マルチパス機能を使用すると、両方の PE デバイスの VRF が同じ RD を使用できるようになります。このようなシナリオでは、ASBR-1 は両方の PE デバイスから同じプレフィックスを受信します。ASBR-1 は、受信したプレフィックスに 1 つの Inter-AS オプション B ラベル、Inlabel L1 のみを割り当て、VPN ルートを ASBR-2 にアドバタイズします。この場合、両方の PE デバイスを使用するトラフィック フローが ASBR-1 の 1 つのプレフィックスとラベルだけで確立されるため、スケール性が強化されます。

図 *7:* 両方の *PE* デバイスで *VRF* が同じ *RD* を使用できるようにする



BGP コスト コミュニティ

BGP コスト コミュニティは非推移的な拡張コミュニティ属性で、iBGP およびコンフェデレーション ピアには渡されますが、eBGP ピアには渡されません。（コンフェデレーションは、同じ自律システム番号を使用して外部ネットワークと通信する、iBGP ピアからなるグループです）。BGP コスト コミュニティ属性には、コスト コミュニティ ID とコスト値が含まれます。BGP コスト コミュニティ属性を設定することにより、ローカルの自律システムまたはコンフェデレーションにおける BGP ベスト パス選択プロセスをカスタマイズできます。コミュニティ ID とコスト値を使用して、ルート マップにコスト コミュニティ属性を設定します。BGP は、コミュニティ ID が最小のパスを優先します。コミュニティ ID が同一の場合には、BGP コスト コミュニティ属性のコスト値が最小のパスを優先します。

同一の宛先に向かう複数のパスが使用可能な場合、BGP はベスト パス選択プロセスを使用して、どのパスがベストであるかを決定します。複数の等コスト パスが使用可能な場合、ユーザーは、特定のパスが優先されるよう設定することができます。

iBGP のアドミニストレーティブ ディスタンスは、ほとんどの内部ゲートウェイ プロトコル（IGP）のディスタンスよりも悪いため、ユニキャスト ルーティング情報ベース（RIB）は、プロトコルまたはルートの通常のディスタンスまたはメトリック比較を使用する前に、同じ BGP コスト コミュニティ比較アルゴリズムを適用する場合があります。。iBGP を介して学習された VPN ルートは、ローカルで学習された IGP ルートよりも優先されます。

コスト拡張コミュニティ リンク属性は、拡張コミュニティ交換が有効な場合、iBGP ピアに伝播します。

BGP コストコミュニティによるベスト パス選択プロセスへの影響

BGP ベスト パス選択プロセスは、挿入ポイント（POI）においてコスト コミュニティ属性の影響を受けます。POI は内部ゲートウェイ プロトコル（IGP）メトリック比較に準拠します。同一の宛先に向かう複数のパスを受信したとき、BGP はベストパス選択プロセスを使用して、いずれのパスがベスト パスであるかを決定します。ベスト パスは BGP により自動的に決定され、ルーティング テーブルにインストールされます。複数の等コスト パスが使用可能な場合、POI で個別のパスにプリファレンスを割り当てることができます。ローカルのベストパス選択で POI が有効でない場合は、コスト コミュニティ属性は暗黙的に無視されます。

コスト コミュニティ属性を使用して、同一の POI に対し複数のパスを設定できます。最も低いコスト コミュニティ ID を持つパスが最優先で検討されます。特定の POI に対するすべてのコスト コミュニティ パスは、最も低いコスト コミュニティ ID を持つパスから考慮されて行きます。コスト コミュニティを持たないパス（POI でコミュニティ ID が評価されるもの）には、デフォルトのコミュニティ コスト値が割り当てられます。

POI でコスト コミュニティ属性を適用することで、ローカルの自律システムまたはコンフェデレーションにおける任意の部分にあるピアを起点とするか、このピアで学習したパスに、値を割り当てることができるようになります。ルータは、コスト コミュニティを、最適パス選択プロセス中の「タイ ブレーカー」として使用できます。同一の自律システムまたはコンフェデレーション内部の個別の等コスト パスに対し、コスト コミュニティのインスタンスを複数設定できます。たとえば、複数の等コスト出口ポイントを持つネットワーク内の特定の出口パスに低コストのコミュニティ値を適用することができます。BGP 最良パス選択プロセスでは、その特定の出口パスを優先します。

コスト コミュニティおよび EIGRP PE-CE とバックドア リンク

バックドア リンクが最初に学習された場合、BGP は拡張内部ゲートウェイ プロトコル（EIGRP）レイヤ 3 VPN トポロジのバックドア リンクを優先します。バックドア リンクまたはルートは、遠隔地とメイン サイト間のレイヤ 3 VPN の外で設定される接続です。

BGP コスト コミュニティの「準最適パス」挿入ポイント（POI）は、VPN およびバックドア リンクが混在する EIGRP レイヤ 3 VPN ネットワーク トポロジをサポートします。この POI は BGP に再配布される EIGRP ルートに自動的に適用されます。準最適パス POI は、EIGRP のルート タイプおよびメトリックを伝送します。この POI は、BGP がその他のあらゆる比較ステップの前にこの POI を考慮するように設定することで、ベスト パス計算プロセスに影響を及ぼします。

MPLS レイヤ 3 VPN ロード バランシングの前提条件

MPLS レイヤ 3 VPN ロード バランシングには、次の前提条件があります。

- MPLS と L3VPN 機能をイネーブルにする必要があります。
- MPLS の正しいライセンスをインストールする必要があります。

MPLS レイヤ 3 VPN ロード バランシングに関する注意事項と制限事項

MPLS レイヤ 3 VPN ロード バランシング設定時の注意事項と制限事項は次のとおりです。

- MPLS レイヤ 3 VPN ロード バランシングは、Cisco Nexus 3600 プラットフォーム スイッチでサポートされています。

- ルータがルート リフレクタの背後にあり、マルチホーム サイトに接続されている場合、VRF ごとに異なる RD を持つ別個のVRF が設定されない限り、アドバタイズされません。
- 複数の iBGP パスがある BGP プレフィックス用の各 IP ルーティング テーブル エントリは、追加メモリを使用します。ルータの使用可能なメモリ量が小さい場合や、ルータがフル インターネット ルーティング テーブルを伝送している場合は、この機能の使用はお勧めしません。
- バックドア リンクが存在し、EIGRP が PE-CE ルーティング プロトコルである場合は、BGP コスト コミュニティを無視しないでください。
- N3K-C3636C-R および N3K-C36180YC-R ラインカードを備えた Cisco Nexus 3600 プラットフォーム スイッチでは、最大 16K の VPN プレフィックスがサポートされます。
- 4K VRF がサポートされます。

MPLS レイヤ 3 VPN ロード バランシングのデフォルト設定

次の表に、MPLS レイヤ 3 VPN ロードバランシング パラメータのデフォルト設定を示します。

表 *6:* デフォルトの *MPLS* レイヤ *3 VPN* ロード バランシング パラメータ

パラメータ	デフォルト
レイヤ 3 VPN 機能	無効
BGP コスト コミュニティ ID	128
BGP コスト コミュニティ コスト	2147483647
最大マルチパス	1
BGP VPNv4 マルチパス	ディセーブル

MPLS レイヤ 3 VPN ロード バランシングの設定

eBGP および iBGP の BGP ロード バランシングの設定

eBGP ネットワークまたは iBGP ネットワークのレイヤ 3 VPN ロード バランシングを設定できます。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル 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	**feature-set mpls** 例： `switch(config)# feature-set mpls`	MPLS フィーチャ セットをイネーブルにします。
ステップ 3	**feature mpls l3vpn** 例： `switch(config)# feature mpls l3vpn`	MPLS レイヤ 3 VPN 機能をイネーブルにします。

ステップ 4	**feature bgp**
	例：
	`switch(config)# feature bgp`
	`switch(config)#`
		BGP 機能をイネーブルにします。

ステップ 5	**router bgp** *as - number*
	例：
	`switch(config)# router bgp 1.1`
	`switch(config-router)#`
		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。
		as-number 引数は、ルータを他の BGP ルータに対して識別し、転送するルーティング情報にタグを設定する自律システムの番号を示します。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

ステップ 6	**bestpath cost-community ignore remote-as** *as-number* 例： `switch(config-router)# bestpath cost-community ignore#`	（オプション）BGP ベストパス計算のコスト コミュニティを無視します。

ステップ 7	**address-family { ipv4 | ipv6 } unicast**
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		IP ルーティング セッションを設定するために、アドレス ファミリ コンフィギュレーション モードに入ります。

	コマンドまたはアクション	目的
ステップ 8	**maximum-paths** [**bgp**] *number-of-paths* 例： `switch(config-router-af)# maximum-paths 4`	許可されるマルチパスの最大数を設定します。ibgp キーワードを使用して、**iBGP** ロード バランシングを設定します。指定できる範囲は 1 ～ 16 です。
ステップ 9	**show running-config bgp** 例： `switch(config-router-vrf-neighbor-af)# show running-config bgp`	（任意）BGP の実行コンフィギュレーションを表示します。
ステップ 10	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

BGPv4 マルチパスの設定

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル 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	**feature bgp** 例： `switch(config)# feature bgp`	BGP 機能をイネーブルにします。

ステップ 3	**router bgp** *as - number*
	例：
	`switch(config)# router bgp 2`
	`switch(config-router)#`
		ルータに割り当てる自律システム（AS）番号を入力し、ルータ BGP 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 vpnv4 unicast**
	例：
	`switch(config-router)# address-family vpnv4 unicast`
	`switch(config-router-af)#`
		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、標準 VPNv4 アドレス プレフィックスを使用する、BGP などのルーティング セッションを設定します。

ステップ 5	**maximum-paths eibgp** *parallel-paths* 例：	eBGP パスと iBGP パスの両方のための BGP VPNv4 マルチパスの最大数を指定

	コマンドまたはアクション	目的
	`switch(config-router-af)# maximum-paths eibgp 3`	します。指定できる範囲は 1 ～ 32 です。

MPLS レイヤ 3 VPN ロード バランシングの設定例

例：MPLS レイヤ 3 VPN ロード バランシング

次に、iBGP ロード バランシングを設定する例を示します。

```
 configure terminal
feature-set mpls
feature mpls l3vpn
feature bgp
router bgp 1.1
bestpath cost-community ignore
address-family ipv6 unicast
maximum-paths ibgp 4
```

例：BGP VPNv4 マルチパス

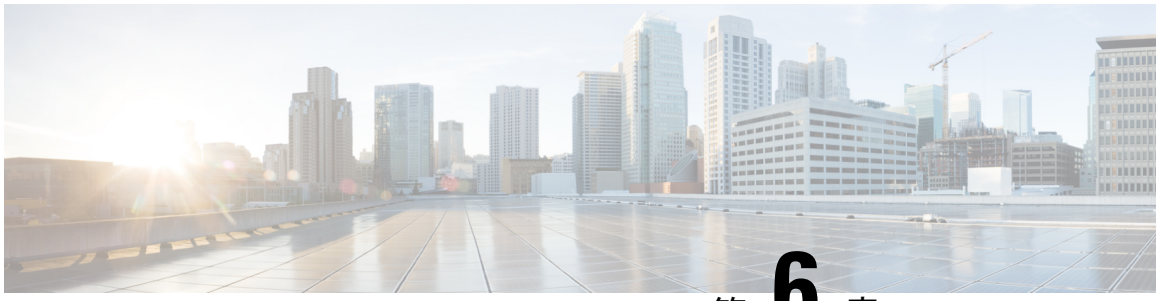
次の例は、最大 3 つの BGP VPNv4 マルチパスを設定する方法を示しています。

```
 configure terminal
router bgp 100
address-family vpnv4 unicast
maximum-paths eibgp 3
```

例：MPLS レイヤ 3 VPN コスト コミュニティ

次の例は、BGP コスト コミュニティを設定する方法を示しています。

```
 configure terminal
feature-set mpls
feature mpls l3vpn
feature bgp
route-map CostMap permit
set extcommunity cost 1 100
router bgp 1.1
router-id 192.0.2.255
neighbor 192.0.2.1 remote-as 1.1
address-family vpnv4 unicast
send-community extended
route-map CostMap in
```



第 6 章

MPLS QoS の設定

この章では、マルチプロトコル ラベル スイッチング（MPLS）レイヤ 3 仮想プライベート ネットワーク（VPN）のサービス品質を設定する方法について説明します。

MPLS Quality of Service（QoS）について

MPLS QoS を使用すると、差別化したサービス タイプを MPLS ネットワーク上で提供できます。差別化したサービス タイプを使用して、各パケットで指定されたサービスを提供することで、さまざまな要件を満たすことができます。QoS では、ネットワーク トラフィックの分類、トラフィック フローのポリシングとプライオリティ設定、および輻輳回避が可能です。

このセクションは、次のトピックで構成されています。

MPLS QoS 用語

ここでは、MPLS QoS 用語を定義します。

- 分類とはマーキングするトラフィックを選択するプロセスです。分類では、選択基準とのマッチングにより、トラフィックを複数の優先レベルまたはサービス クラスに分割します。トラフィック分類は、クラス ベースの QoS プロビジョニングのプライマリ コンポーネントです。スイッチは、受信した MPLS パケット（ポリシーのインストール後）の最上位ラベルの EXP ビットに基づき、分類を行います。
- Diffserv コード ポイント（DSCP）

- IP ヘッダーの ToS バイトの最初の 6 ビット。
- IP パケットだけに存在します。
- IPv4 または IPv6 パケットに存在できます。
- IPv6 ヘッダーの 8 ビット トラフィック クラス オクテットの最初の 6 ビットです。

- E-LSP：ラベル スイッチド パス（LSP）の 1 つであり、ノードはここで MPLS ヘッダーの実験（EXP）ビットから排他的に MPLS パケットの QoS 処理を判断します。QoS 処理が EXP（クラスおよびドロップ優先順位の両方）から判断されるため、いくつかのクラスのトラフィックを 1 つの LSP に多重化することができます（同じラベルを使用）。EXP フィールドは 3 ビット フィールドであるため 1 つの LSP は最大 8 つのトラフィックのクラスをサポートすることができます。
- EXP ビット：ノードがパケットに与える QoS 処理（Per Hop Behavior）を定義します。これは、IP ネットワークの DiffServ コード ポイント（DSCP）に相当します。DSCP は、クラスとドロップ優先順位を定義します。EXP ビットは、一般に IP DSCP でエンコードされた情報をすべて伝送するのに用いられます。ただし、ドロップ優先順位をエンコードするために EXP ビットが排他的に用いられる場合もあります。
- マーキング：パケットのレイヤ 3 DSCP 値を設定するプロセスです。マーキングはまた、MPLS EXP フィールドで異なった値を選択してパケットにマーキングし、輻輳時にパケットが必要なプライオリティを持つようにするプロセスでもあります。
- MPLS 実験フィールド：MPLS 実験（EXP）フィールド値を設定すると、自己のネットワークで伝送される IP パケット内で IP precedence フィールドの値が変更されることを望まないという、オペレータの要件を満たすことができます。MPLS EXP フィールドで異なった値を選択することにより、輻輳時にパケットが必要なプライオリティを持つようパケットをマーキングすることができます。デフォルトでは、インポジション中に、DSCP の最上位 3 ビットが MPLS EXP フィールドにコピーされます。MPLS QoS ポリシーで MPLS EXP ビットをマークできます。

MPLS QoS の機能

QoS により、ネットワークは選択されたネットワーク トラフィックに提供するサービスを向上させることができます。ここでは、次の MPLS QoS 機能について説明します。これらは MPLS ネットワークでサポートされます。

MPLS 実験フィールド

MPLS EXP（実験）フィールド値を設定すると、サービス プロバイダーが自己のネットワークで伝送された IP パケット内で変更された IP precedence フィールドの値を望まない場合に、サービス プロバイダーの要件を満たすことができます。

MPLS EXP フィールドで異なった値を選択することにより、輻輳時にパケットが必要なプライオリティを持つようパケットをマーキングすることができます。

デフォルトでは、インポジション中に、IP precedence 値が MPLS EXP フィールドにコピーされます。MPLS QoS ポリシーで MPLS EXP ビットをマークできます。

信頼

受信レイヤ 3 MPLS パケットに対し、PFC は、通常、受信最上位ラベルの EXP 値を信頼します。MPLS パケットは、次のいずれの影響も受けません。

- インターフェイスの信頼状態
- ポートの CoS 値
- policy-map trust コマンド

受信レイヤ 2 MPLS パケットの場合、PFC は、CoS および出力キュー処理の目的で、受信最上位ラベルの EXP 値を信頼するか、ポートまたはポリシーの信頼を MPLS パケットに適用できます。

分類

分類とはマーキングするトラフィックを選択するプロセスです。分類は、トラフィックを複数の優先順位レベル、つまり、サービス クラスに分割することによりこのプロセスを実施します。トラフィック分類は、クラス ベースの QoS プロビジョニングのプライマリ コンポーネントです。

ポリシングおよびマーキング

ポリシングを行うと、設定レートを超えたトラフィックは廃棄されるか、またはより高いドロップ優先順位にマークダウンされます。マーキングは、パケット フローを識別して、これらを区別する手法です。パケット マーキングを利用すれば、ネットワークを複数の優先プライオリティ レベルまたはサービス クラスに分割することができます。

実装可能な MPLS QoS ポリシングおよびマーキング機能は、受信したトラフィック タイプ、およびトラフィックに適用される転送処理によって決まります。

MPLS スイッチングに関する注意事項と制限事項

MPLS Quality of Service（QoS）設定時の注意事項と制限事項は次のとおりです。

- QoS ポリシーを設定する場合、**topmost**（**set mpls 実験的インポジション** CLI のキーワード）はサポートされません。
- MPLS QoS は、インポジション用に MAX 4 ラベル スタックをサポートします。
- MPLS QoS は、ポリシングに基づく備考をサポートしていません。
- L3 EVPN 出力ノード - ポリシングは、システム レベルの mpls-in-policy ではサポートされていません。

- MPLS EXP に基づく出力 QoS 分類はサポートされていません。
- EXP ラベルは、新しくプッシュまたはスワップされたラベルに対してのみ設定されます。内部ラベルの EXP は変更されません。
- ラベル エッジ ルータ（LER）では、EXP でのポリシーのマッチングはサポートされていません。内部 DSCP を使用してパケットをマッチングさせることはできます。
- インターフェイス ポリシーを使用して、出力ラベル エッジ ルータ（LER）上の MPLS L3 EVPNパケットを分類することはできません。トラフィックの分類には、システム レベルの MPLS-Default ポリシーが使用されます。
- 明示的輻輳通知（ECN）マーキングは、ラベル スイッチング ルータ トランジット ノードではサポートされていません。
- Cisco NX-OS リリース 9.3(1) の MPLS ハンドオフでは、デフォルトの QoS サービス テンプレートのみがサポートされています。MPLS に EXP ラベルを設定することはできません。

MPLS QoS の設定

（注） この機能の Cisco NX-OS コマンドは、Cisco IOS のコマンドとは異なる場合があるので注意してください。

MPLS 入力ラベル スイッチド ルータの設定

MPLS 入力ラベル スイッチド ルータを設定するには、次の手順を実行します。

MPLS 入力 LSR の分類

Differentiated Services Code Point（DSCP）の値にマッチさせるには、QoS ポリシーマップ クラス コンフィギュレーション モードで **match dscp** コマンドを使用します。設定をディセーブルにするには、コマンドの **no** 形式を使用します。

（注） デフォルトのエントリは、入力 QoS ポリシーが設定されていない場合に DSCP でマッチし、EXP をマークするようにプログラムされています (encap での均一モードの動作)。

始める前に

- MPLS 設定を有効にする必要があります。

- 正しい VDC を使用していることを確認します（または switch to 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	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル 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	[**no**] **class-map type qos** *class-map-name*
	例：
	`switch(config)# class-map type qos Class1`
	`switch(config-cmap-qos)#`
		クラス マップを定義し、クラスマップ コンフィギュレーション モードを開始します。

ステップ 3	[**no**] **match** [**not**] **dscp** *dscp-list*
	例：
	`switch(config)#`
	`switch(config-cmap-qos)# match dscp 2-4`
		DSCP 値のリストです。次のように、MPLS ヘッダーの DSCP ラベルにパケットがマッチする（またはしない）必要があることを指定します。
		• **dscp-list**：リストには値と範囲を含めることができます。値の範囲は 0 ～ 63 です。

MPLS 入力ポリシングおよびマーキングの設定

ポリシーマップの値を構成し、すべてのインポーズ ラベル エントリで EXP 値を設定するには、QoS ポリシー マップ クラス コンフィギュレーション モードで **set mpls experimental imposition** コマンドを使用します。設定をディセーブルにするには、コマンドの **no** 形式を使用します。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル 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	[**no**] **policy-map type qos** *policy-map-name* 例：	ポリシーマップを定義し、ポリシーマップ コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
	`switch(config)# policy-map type qos pmap1` `switch(config-pmap-qos)#`	
ステップ 3	**class** *class-name* 例： `switch(config-pmap-qos)# class Class1`	クラスマップに名前を付けます。

ステップ 4	**set mpls experimental imposition** *exp_imposition_name*
	例：
	`switch(config)#`
	`switch(config-pmap-qos)# set mpls experimental imposition 2`
		MPLS の実験（EXP）値です。範囲は 0 ～ 7 です。

ステップ 5	**set qos-group** *group-number* 例： `switch(config-cmap-qos)# set qos-group 1`	qos-group 番号を識別します。
ステップ 6	**police cir** *burst-in-msec* **bc** *conform-burst-in-msec* **conform-action** *conform-action* **violate-action** *violate-action* 例： `switch(config-pmap-qos)# police cir 100 mbps bc 200 ms conform transmit violate drop`	ポリシーマップ クラス ポリシング コンフィギュレーション モードで、分類するトラフィック用のポリサーを定義します。

ステップ 7	**interface** *type slot/port*
	例：
	`switch(config)# interface ethernet 2/2`
	`switch(config-if)#`
		指定した入力インターフェイス、出力インターフェイス、仮想回線（VC）、またはインターフェイスや VC のサービス ポリシーとして使用される VC のためのインターフェイス コンフィギュレーション モードに入ります。

ステップ 8	**service-policy type qos input** *policy-map-name*
	例：
	`switch(config-if)# service-policy type qos input pmap1`
	`switch(config-if)#`
		ポリシー マップを入力インターフェイス、仮想回線（VC）、出力インターフェイス、またはインターフェイスまたは VC のサービス ポリシーとして使用される VC にアタッチします。

MPLS トランジット ラベル スイッチング ルータの設定

MPLS トランジット ラベル スイッチング ルータを設定するには、次の手順を実行します。

MPLS Transit LSR 分類

MPLS EXP フィールドの値をすべてのインポーズされたラベル エントリにマッピングするには、QoS ポリシーマップ クラス コンフィギュレーション モードで **set mpls experimental topmost** コマンドを使用します。設定をディセーブルにするには、コマンドの **no** 形式を使用します。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル 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	[**no**] **class-map type qos** *class-map-name*
	例：
	`switch(config)# class-map type qos Class1`
	`switch(config-cmap-qos)#`
		クラス マップを定義し、クラスマップ コンフィギュレーション モードを開始します。

ステップ 3	[**no**] **match** [**not**] **mpls experimental topmost** *exp-list*
	例：
	`switch(config)#`
	`switch(config-cmap-qos)# match mpls experimental topmost 2, 4-7`
		MPLS 実験（EXP）値のリスト。次のように、MPLS ヘッダーの最も外側の（最上位の）MPLS ラベルにある 3 ビットの EXP フィールドに、パケットがマッチする（またはしない）必要があることを指定します。
		• **exp-list**：リストには値と範囲を含めることができます。指定できる範囲は 0 ～ 7 です。

MPLS トランジット ポリシングおよびマーキングの設定

ポリシー マップ値を構成し、インポーズされたすべてのラベル エントリに EXP 値を設定するには、インターフェイス構成モードで **service-policy type qos input pmap1** コマンドを使用します。設定をディセーブルにするには、コマンドの **no** 形式を使用します。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル 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	[**no**] **policy-map type qos** *policy-map-name*
	例：
	`switch(config)# policy-map type qos Class1`
	`switch(config-pmap-qos)#`
		ポリシーマップを定義し、ポリシーマップ コンフィギュレーション モードを開始します。

ステップ 3	**class** *class-name* 例： `switch(config-pmap-qos)# class Class1`	クラスマップに名前を付けます。

ステップ 4	**set mpls experimental imposition** *exp_imposition_name*
	例：
	`switch(config)#`
	`switch(config-pmap-qos)# set mpls experimental imposition 2`
		MPLS の実験（EXP）値です。範囲は 0 ～ 7 です。

ステップ 5	**set qos-group** *group-number* 例： `switch(config-pmap-qos)# set qos-group 1`	qos-group 番号を識別します。
ステップ 6	**police cir** *burst-in-msec* **bc** *conform-burst-in-msec* **conform-action** *conform-action* **violate-action** *violate-action* 例： `switch(config-pmap-qos)# police cir 100 mbps bc 200 ms conform transmit violate drop`	ポリシーマップ クラス ポリシング コンフィギュレーション モードで、分類するトラフィック用のポリサーを定義します。 • 違反アクション：トランジット LSR でサポートされているキーワードは **drop** だけです

ステップ 7	**interface** *type slot/port*
	例：
	`switch(config)# interface ethernet 2/2`
	`switch(config-if)#`
		指定した入力インターフェイス、出力インターフェイス、仮想回線（VC）、またはインターフェイスや VC のサービス ポリシーとして使用される VC のためのインターフェイス コンフィギュレーション モードに入ります。

ステップ 8	**service-policy type qos input** *policy-map-name*
	例：
	`switch(config-if)# service-policy type qos input pmap1`
	`switch(config-if)#`
		ポリシー マップを入力インターフェイス、仮想回線（VC）、出力インターフェイス、またはインターフェイスまたは VC のサービス ポリシーとして使用される VC にアタッチします。

MPLS 出力ラベル スイッチング ルータの設定

MPLS 出力ラベル スイッチド ルータを設定するには、次の手順を実行します。

MPLS 出力 LSR の分類

出力キューへの着信 SR MPLS トラフィックを分類するには、Differentiated Services Code Point（DSCP）フィールドの一致を使用します。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル 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	[**no**] **class-map type qos** *class-map-name*
	例：
	`switch(config)# class-map type qos Class1`
	`switch(config-cmap-qos)#`
		クラス マップを定義し、クラスマップ コンフィギュレーション モードを開始します。

ステップ 3	[**no**] **match** [**not**] **dscp** *dscp-list*
	例：
	`switch(config)#`
	`switch(config-cmap-qos)# match dscp 2-4`
		DSCP 値のリストです。次のように、MPLS ヘッダーの DSCP ラベルにパケットがマッチする（またはしない）必要があることを指定します。
		• **dscp-list**：リストには値と範囲を含めることができます。値の範囲は 0 ～ 63 です。

MPLS 出力 LSR 分類 - デフォルト ポリシー テンプレート

EVPN トンネルの出力キューへの着信トラフィックを分類するには、システム レベルでデフォルトの **default-mpls-in-policy** コマンドを使用します。設定をディセーブルにするには、コマンドの **no** 形式を使用します。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル 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	**[no] system qos**
	例：
	`switch(config)# system qos`
	`switch(config-sys-qos)#`
		システム QoS コ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] service-policy type qos input default-mpls-in-policy** 例： `switch(config-sys-qos)# service-policy type qos input default-mpls-in-policy`	着信 SR L3 EVPN MPLS トラフィックで照合するには、システム レベルで「default-mpls-in-policy」を指定します。

次に、**service-policy type qos input default-mpls-in-policy** コマンドで設定されたポリシー テンプレートのデフォルトの MPLS を示します。

```
policy-map type qos default-mpls-in-policy
    class  c-dflt-mpls-qosgrp1
      set qos-group 1
    class  c-dflt-mpls-qosgrp2
      set qos-group 2
    class  c-dflt-mpls-qosgrp3
      set qos-group 3
    class  c-dflt-mpls-qosgrp4
      set qos-group 4
    class  c-dflt-mpls-qosgrp5
      set qos-group 5
    class  c-dflt-mpls-qosgrp6
      set qos-group 6
    class  c-dflt-mpls-qosgrp7
      set qos-group 7
    class  class-default
      set qos-group 0

class-map type qos match-any c-dflt-mpls-qosgrp1
  Description: This is an ingress default qos class-map that classify traffic with prec
 1
  match precedence 1

class-map type qos match-any c-dflt-mpls-qosgrp2
  Description: This is an ingress default qos class-map that classify traffic with prec
 2
  match precedence 2

class-map type qos match-any c-dflt-mpls-qosgrp3
  Description: This is an ingress default qos class-map that classify traffic with prec
 3
  match precedence 3

class-map type qos match-any c-dflt-mpls-qosgrp4
  Description: This is an ingress default qos class-map that classify traffic with prec
 4
  match precedence 4

class-map type qos match-any c-dflt-mpls-qosgrp5
  Description: This is an ingress default qos class-map that classify traffic with prec
 5
  match precedence 5
```

```
class-map type qos match-any c-dflt-mpls-qosgrp6
  Description: This is an ingress default qos class-map that classify traffic with prec
 6
  match precedence 6

class-map type qos match-any c-dflt-mpls-qosgrp7
  Description: This is an ingress default qos class-map that classify traffic with prec
 7
  match precedence 7
```

カスタム MPLS-in-Policy マッピング

提供されたテンプレートのローカルコピーを編集することにより、着信トラフィックのキューマッピングをオーバーライドできます。システムマッチングは常に優先順位に基づいており、「mpls-in-policy」文字列がポリシー名の一部であることが必要です。QoS によるマーキングがサポートされています。セットは、qos-group、vlan-cos、またはその両方です。

```
class-map type qos match-all prec-1
      match precedence 1
    class-map type qos match-all prec-2
      match precedence 2

policy-map type qos test-mpls-in-policy
    class  prec-1
      set qos-group 3
    class  prec-2
      set qos-group 4
system qos
  service-policy type qos input test-mpls-in-policy
```

（注） 優先順位に基づく分類のみがサポートされ、マーキングはシステム レベルの mpls-in-policy ではサポートされません。

MPLS 出力 LSR の設定：ポリシングおよびマーキング

ポリサー設定でポリシーマップを設定して適用するには、インターフェイス コンフィギュレーション モードで **service-policy type qos input pmap1** コマンドを使用します。設定をディセーブルにするには、コマンドの **no** 形式を使用します。

（注） ポリシングは SR L3 EVPN MPLS トラフィックではサポートされていません。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル 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	[no] **policy-map type qos** *class-map-name*
	例：
	`switch(config)# policy-map type qos Class1`
	`switch(config-pmap-qos)#`
		クラス マップを定義し、クラスマップ コンフィギュレーション モードを開始します。

ステップ 3	**policy** *policy-name* 例： `switch(config-pmap-qos)# class Class1`	クラスマップに名前を付けます。
ステップ 4	**set dscp** *dscp-value* 例： `switch(config-pmap-qos)# set dscp 4`	dscp 値を識別します。
ステップ 5	**set qos-group** *group-number* 例： `switch(config-pmap-qos)# set qos-group 1`	qos-group 番号を識別します。
ステップ 6	[no] **police cir** *burst-in-msec* **bc** *conform-burst-in-msec* **conform-action** *conform-action* **violate-action** *violate-action* 例： `switch(config-pmap-qos)# police cir 100 mbps bc 200 ms conform transmit violate drop`	ポリシーマップ クラス ポリシング コンフィギュレーション モードで、分類するトラフィック用のポリサーを定義します。

ステップ 7	**interface** *type slot/port*
	例：
	`switch(config)# interface ethernet 2/2`
	`switch(config-if)#`
		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 8	[no] **service-policy type qos input** *policy-map-name*
	例：
	`switch(config-if)# service-policy type qos input pmap1`
	`switch(config-if)#`
		ポリシー マップを入力インターフェイス、仮想回線（VC）、出力インターフェイス、またはインターフェイスまたは VC のサービス ポリシーとして使用される VC にアタッチします。

トラフィック キューイングについて

トラフィックのキューイングとは、パケットの順序を設定して、データの入力と出力の両方に適用することです。デバイスモジュールでは複数のキューをサポートできます。これらのキューを使用することで、さまざまなトラフィック クラスでのパケットのシーケンスを制御できます。また、重み付けランダム早期検出（WRED）およびテール ドロップしきい値を設定することもできます。デバイスでは、設定したしきい値を超えた場合にだけパケットがドロップされます。

QoS トラフィック キューイングの設定

出力キューを設定するには、ポリシー マップ コンフィギュレーション モードで **set qos-group** コマンドを使用します。設定をディセーブルにするには、コマンドの **no** 形式を使用します。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル 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	[**no**] **policy-map type qos** *class-map-name*
	例：
	`switch(config)# class-map type qos Class1`
	`switch(config-cmap-qos)#`
		クラス マップを定義し、クラスマップ コンフィギュレーション モードを開始します。

ステップ 3	**class** *class-name* **例：** `switch(config-cmap-qos)# class Class1`	クラスマップに名前を付けます。
ステップ 4	**set qos-group** *qos_group_number* **例：** `switch(config-pmap-c-qos)# set qos-group`	ポリシー マップの名前付き QoS グループのキューイング パラメータを適用します。範囲は 0 ～ 7 です。

MPLS QoS の確認

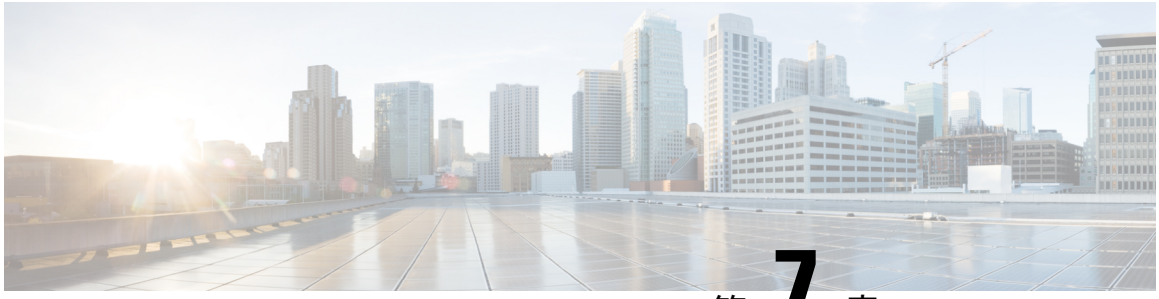
MPLS QoS 設定を表示するには、次の作業を実行します。

コマンド	説明
show hardware internal forwarding table utilization	MAX ラベル エントリと Used ラベル エントリに関する情報を表示します。
show class-map	インターフェイス クラスマッピングの統計情報を表示します。
show policy-map system type qos input	すべてのインターフェイスのすべてのクラスに一致したパケットを示す累積統計を表示します（EVPN トンネルの場合のみ）。詳細については、この表に続く出力例を参照してください。
show policy-map type qos interface interface	指定方向の対象インターフェイスにある各クラスに一致するパケットを表示する統計情報を表示します。
show policy-map type qos <pmap name>	インターフェイス上で設定されたサービス ポリシー マップを表示します。
show queuing interface	インターフェイスのキューイング情報を表示します。

次の例は、すべてのインターフェイスのすべてのクラスに一致したパケットを示す累積統計を表示します（EVPN トンネルの場合のみ）。

```
switch# show policy-map system type qos input


  Service-policy (qos) input:   default-mpls-in-policy

    Class-map (qos):   c-dflt-mpls-qosgrp1 (match-any)

     Slot 3
        2775483 packets
     Aggregate forwarded :
        2775483 packets
      Match: precedence 1
      set qos-group 1

    Class-map (qos):   c-dflt-mpls-qosgrp2 (match-any)

     Slot 3
        2775549 packets
     Aggregate forwarded :
        2775549 packets
      Match: precedence 2
      set qos-group 2

    Class-map (qos):   c-dflt-mpls-qosgrp3 (match-any)

     Slot 2
        2777189 packets
     Aggregate forwarded :
        2777189 packets
```

```
  Match: precedence 3
  set qos-group 3

Class-map (qos):   c-dflt-mpls-qosgrp4 (match-any)

 Slot 3
    2775688 packets
 Aggregate forwarded :
    2775688 packets
  Match: precedence 4
  set qos-group 4

Class-map (qos):   c-dflt-mpls-qosgrp5 (match-any)

 Slot 3
    2775756 packets
 Aggregate forwarded :
    2775756 packets
  Match: precedence 5
  set qos-group 5

Class-map (qos):   c-dflt-mpls-qosgrp6 (match-any)

 Slot 3
    2775824 packets
 Aggregate forwarded :
    2775824 packets
  Match: precedence 6
  set qos-group 6

Class-map (qos):   c-dflt-mpls-qosgrp7 (match-any)

 Slot 3
    2775892 packets
 Aggregate forwarded :
    2775892 packets
  Match: precedence 7
  set qos-group 7

Class-map (qos):   class-default (match-any)

 Slot 3
    2775962 packets
 Aggregate forwarded :
    2775962 packets
  set qos-group 0
```



第 7 章

MVPN の設定

この章には、マルチキャスト仮想プライベート ネットワーク（MVPN）の構成方法に関する情報が含まれています。

MVPN について

マルチキャスト仮想プライベート ネットワーク（MVPN）機能を使用すると、レイヤー 3 VPN を介したマルチキャスト接続をサポートできます。IP マルチキャストは、ビデオ、音声、およびデータを VPN ネットワーク コアにストリーミングするために使用します。

従来、ポイントツーポイント トンネルはエンタープライズまたはサービス プロバイダー ネットワークに接続する唯一の方法でした。このようなトンネル ネットワークは、スケーラビリティの問題が発生しますが、IP マルチキャスト トラフィックを仮想プライベート ネットワーク（VPN）に通過させる唯一の方法でした。レイヤ 3 VPN はユニキャスト トラフィック接続のみをサポートするため、レイヤ 3 VPN を展開することによって、オペレーターは、レイヤ 3 VPN のカスタマーにユニキャスト接続とマルチキャスト接続の両方を提供できます。

MVPN を使用すると、MPLS 環境でマルチキャスト トラフィックを設定し、サポートできます。MVPN は、仮想ルーティングおよび転送（VRF）インスタンスごとにマルチキャスト パケットのルーティングと転送をサポートし、また、エンタープライズまたはサービス プロバイダーのバックボーン全体にわたって VPN マルチキャスト パケットを転送するためのメカニズムも提供します。IP マルチキャストは、ビデオ、音声、およびデータを VPN ネットワーク コアにストリーミングするために使用します。

VPNは、インターネットサービスプロバイダー（ISP）のような共有インフラストラクチャにネットワークの接続性を提供します。この機能により、低い所有コストでプライベートネットワークと同じポリシーとパフォーマンスを提供します。

MVPN により、企業はネットワーク バックボーン全体でプライベート ネットワークをトランスペアレントに相互接続することができます。MVPNs を使用して企業ネットワークを相互接続しても、企業ネットワークの管理方法や、企業の全体的な接続性は変更されません。

MPLS MVPN のルーティング、転送、マルチキャスト ドメイン

MVPNsは、VPNルーティングおよび転送テーブルにマルチキャストルーティング情報を導入します。プロバイダーエッジ（PE）ルータがカスタマーエッジ（CE）ルータからマルチキャスト データまたはコントロール パケットを受信する場合は、ルータが VPN ルーティング/転送（MVRF）の情報に基づいてデータまたはコントロール パケットを転送します。

マルチキャスト トラフィックを相互に送信できる MVRF のセットは、マルチキャスト ドメインの構成要素です。たとえば、特定タイプのマルチキャストトラフィックをすべてのグローバルな従業員に送信するカスタマーのマルチキャストドメインは、そのエンタープライズと関連するすべての CE ルータから構成されます。

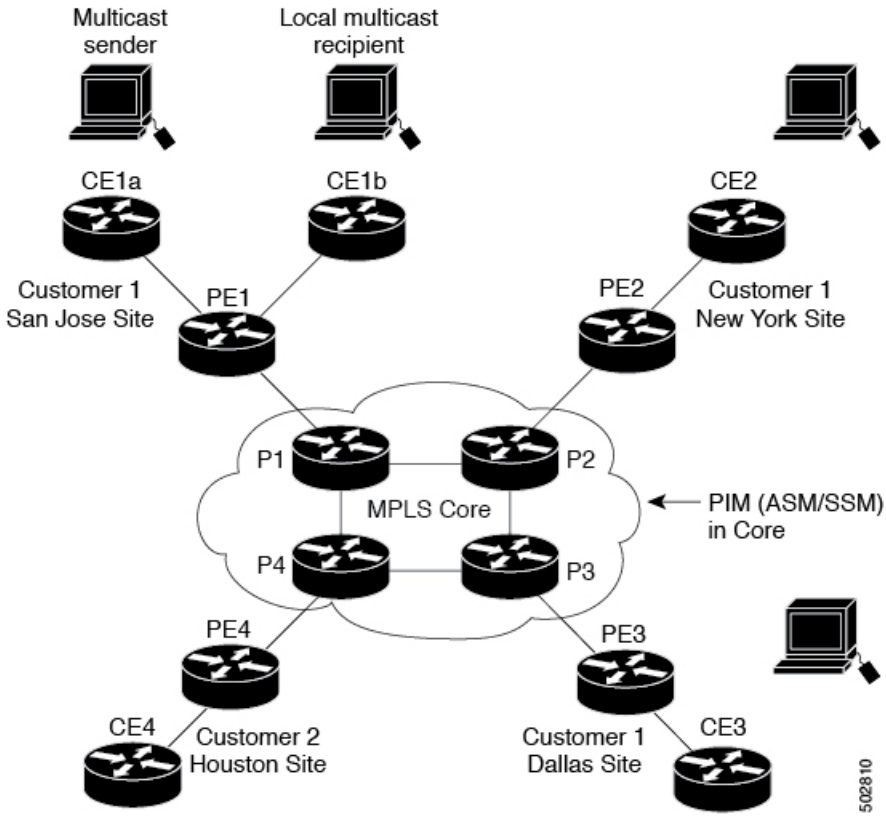
マルチキャスト分散ツリー

MVPN は、各マルチキャスト ドメインにスタティック デフォルト マルチキャスト配信ツリー（MDT）を確立します。デフォルトMDTは、PEルータが使用するパスを定義し、マルチキャスト ドメインにある他のすべての PE ルータに、マルチキャスト データとコントロール メッセージを送信します。

また、MVPN は、高帯域幅伝送用の MDT のダイナミックな作成もサポートします。データMDTは、VPN内のフルモーション ビデオなどの高帯域幅の送信元向けであり、VPNコアの最適なトラフィック転送を確保することを目的としています。

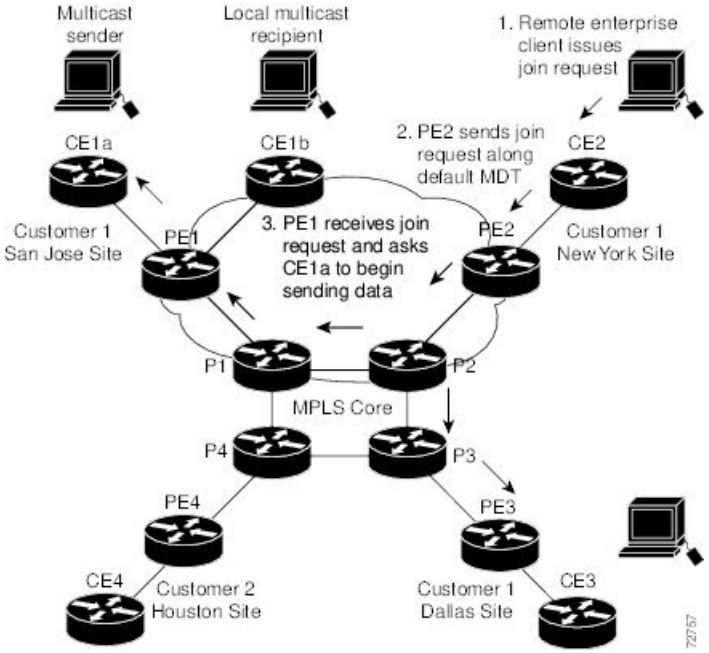
次の例のサービス プロバイダには、San Jose、New York、Dallas にオフィスがあるマルチキャストカスタマーがいます。San Joseでは、一方向のマルチキャストプレゼンテーションが行われています。サービス プロバイダー ネットワークでは、このカスタマーと関連する 3 つすべてのサイト、および別のエンタープライズカスタマーのHoustonサイトがサポートされます。エンタープライズ カスタマーのデフォルト MDT は、プロバイダのルータ P1、P2、P3、およびその関連 PE ルータから構成されています。PE4 は別のカスタマーに関連付けられているため、デフォルト MDT の一部ではありません。次の図からは、San Jose 外では誰もマルチキャストに加入していないため、データがデフォルト MDT に沿って転送されていないことがわかります。

図 *8 :* デフォルト マルチキャスト配信ツリーの概要



New York の従業員がマルチキャスト セッションに加入します。New York のサイトに関連付けられている PE ルータは、カスタマーのマルチキャスト ドメインのデフォルト MDT を介して転送される加入要求を送信します。PE1 は、マルチキャスト セッションの送信元に関連付けられている PE ルータであり、この要求を受信します。次の図は、PE ルータが、マルチキャスト送信元（CE1a）と関連付けられた CE ルータに要求を転送することを示しています。

図 *9:* データ *MDT* の初期化



CE ルータ（CE1a）が関連する PE ルータ（PE1）へマルチキャスト データの送信を開始すると、PE ルータ（PE1）は、デフォルト MDT に沿ってマルチキャスト データを送信します。PE1 はデータ MDT を作成し、データ MDT に関する情報を含むデフォルト MDT を使用して、すべてのルータにメッセージを送信し、3 秒後、データ MDT を使用して、その特定のストリームのマルチキャスト データを送信し始めます。この送信元に関係する受信先は PE2 だけにあるので、PE2 だけがデータ MDT に加入し、データ MDT でトラフィックを受信します。（データ MDT が設定されず、デフォルト MDT のみが設定されている場合、すべてのカスタマー サイトが不要なトラフィックを受信することになります）。PE ルータは、デフォルト MDT を介して他の PE ルータと PIM 関係を維持するとともに、直接接続された P ルータとの PIM 関係をも維持します。

マルチキャスト トンネル インターフェイス

マルチキャスト ドメインごとに作成される VPN ルーティング/転送（MVRF）では、ルータは、すべての MVRF トラフィックが発信されるトンネル インターフェイスを作成する必要があります。マルチキャスト トンネル インターフェイスは、MVRF がマルチキャスト ドメインにアクセスするために使用するインターフェイスです。インターフェイスは、MVRF とグローバル MVRF を接続するコンジットです。MVRF ごとに 1 つのトンネル インターフェイスが作成されます。

MPLS MVPN の利点

MVPNs の利点は、次のとおりです。

- 複数の場所に情報を動的に送信するスケーラブルなメソッドを提供します。

- 高速な情報伝送を提供します。
- 共有インフラストラクチャを介して接続性を提供します。

BGP アドバタイズメント方式 - MVPN サポート

PIM-SM 環境ではなく PIM Source Specific Multicast（PIM-SSM）環境でデフォルト MDT を設定する場合は、受信側 PE は送信元 PE とデフォルト MDT に関する情報を必要とします。この情報は、送信元 PE に（S,G）join を送信し、送信元 PE からの配信ツリーを構築するために使用されます。ランデブーポイント（RP）は必要ありません。送信元のプロバイダーエッジ（PE）アドレスとデフォルト MDT のアドレスは、ボーダー ゲートウェイ プロトコル（BGP）を使用して送信されます。

BGP MDT SAFI

BGP MDT SAFI は、MVPNs に使用される BGP アドバタイズメント メソッドです。現在のリリースでは、IPv4 のみがサポートされています。MDT SAFI の設定は次のとおりです。

- AFI = 1
- SAFI = 66

Cisco NX-OS では、BGP MDT SAFI のアップデートを使用して送信元 PE アドレスと MDT アドレスが PIM に渡されます。ルート記述子（RD）は RD type 0 に変更されており、BGP は PIM に情報を渡す前に、MDT アップデートのための最良パスを決定します。

address-family ipv4 mdt コマンドを使用して、BGP ネイバーの MDT SAFI アドレス ファミリを設定する必要があります。また、ローカル BGP の設定で MDT SAFI をサポートしていないネイバーをイネーブルにする必要があります。MDT SAFI が導入される前、VPNv4 ユニキャスト設定からの追加の BGP 設定は、MVPNs をサポートするために必要ではありませんでした。

前提条件

MVPN の設定には、次の前提条件があります。

- ネットワークに MPLS およびラベル配布プロトコル（LDP）を設定する必要があります。PE ルータを含む、コア内のすべてのルータは、MPLS 転送をサポートできる必要があります。PE 送信元アドレスにラベル付きパスが存在しない場合、VPNv4 ルートは BGP によってインストールされません。
- MPLS の正しいライセンスおよび MPLS で使用する他の機能をインストールすることが必要です。

MVPN に関する注意事項と制限事項

MVPN の設定に関する注意事項と制約事項は次のとおりです。

- MVPN は、Cisco NX-OS リリース 9.3(3) 以降でサポートされます。
- Cisco NX-OS リリース 9.3（3）では、MVPN は Cisco Nexus 3600 （N3K-C36180YC-R、N3K-C3636C-R）プラットフォーム スイッチでのみサポートされます。
- 双方向フォワーディング検出（BFD）は、マルチキャスト トンネル インターフェイス（MTI）ではサポートされていません。
- デフォルトでは、BGP アップデートのソースは、MVPN トンネルのソースとして使用されます。ただし、mdt source を使用して BGP アップデートのソースを上書きし、マルチキャスト トンネルに異なる送信元を提供することができます。
- MVPN は、最大 16 の MDT 送信元インターフェイスをサポートします。
- MVPN 操作に参加するすべてのルータで MDT SAFI を設定する必要があります。
- コネクタ属性を伝送する VPNv4 内部 BGP（iBGP）セッションには、拡張コミュニティが必要です。
- MDT の MTU 設定はサポートされていません。MVPN 経由で送信できる最大カスタマー マルチキャスト パケット サイズは、コア インターフェイスの MTU によって制限されます。例：
  - MTU 1500 – カスタマー IP パケット サイズ = 1476
  - MTU 9216 – カスタマー IP パケット サイズ = 9192
- 一部の MVPN マルチキャスト制御パケットは、copp-system-p-class-l2-default CoPP ポリシーに分類されます。違反数が増加した場合は、CoPP ポリシーを変更して、このクラスのポリサー レートを増やすことをお勧めします。
- MDT 双方向有効化はサポートされていません。
- vPC は MVPN ではサポートされていません。
- トランジット PE ルータにレシーバがなく、RP である CE に接続されている場合、データ MDT エントリはキャッシュされません。データ MDT エントリは、ローカル レシーバがこの PE ルータに接続されている場合にのみキャッシュされます。ただし、エントリが事前にダウンロードされないため、切り替えに遅延が発生します。
- 日付 MDT の場合、「即時切り替え」モードのみがサポートされます。しきい値ベースのスイッチングはサポートされていません。
- PE デバイスと P /PE デバイス間のサブインターフェイスおよび SVI サポートは利用できません。
- MVPN 整合性チェッカーは、Cisco Nexus リリース 9.3(3) ではサポートされていません。

- MTI インターフェイスの統計は、Cisco Nexus リリース 9.3(3) ではサポートされていません。
- Cisco Nexus リリース 9.3(3) では、ASIC ごとに最大 40G のマルチキャスト トラフィックがサポートされます。

MVPN のデフォルト設定

表 *7:* デフォルトの *MVPN* パラメータ

パラメータ	デフォルト
mdt default *address*	デフォルトなし
mdt enforce-bgp-mdt-safi	有効
mdt source	デフォルトなし
mdt ip pim hello-interval *interval*	30000 ミリ秒
mdt ip pim jp-interval *interval*	60000 ミリ秒
mdt default asm-use-shared-tree	ディセーブル

MVPN の設定

この章では、Cisco NX-OS デバイスでマルチキャスト仮想プライベート ネットワーク（MVPN）を設定する方法について説明します。

（注） MVPN の場合、新しい TCAM 領域「ing-mvpn」が使用されます（デフォルト サイズは 10）。この領域は自動的に分割されるため、分割する必要はありません。この TCAM 領域が分割されているかどうかを確認するには、次のコマンドを使用します。

```
switch# show hardware access-list tcam region | i ing-mvpn
Ingress mVPN [ing-mvpn] size = 10
switch#
```

なんらかの理由で領域が分割されていない場合（サイズが 0 と示される）、次のコマンドを使用して TCAM 領域をサイズ 10 に分割し、デバイスをリロードできます。TCAM はサイズ 10 に分割されているものと予期されています。

```
switch (config)# hardware access-list tcam region ing-mvpn 10
WARNING: On module 2,
WARNING: On module 4,
Warning: Please reload all linecards for the configuration to take effect
switch (config)#
```

MVPN の有効化

Cisco NX-OS リリース 9.3（3）以降、Cisco Nexus 3600 プラットフォーム スイッチで MVPN を設定できます。

始める前に

install feature-set mpls コマンドと **feature-set mpls** コマンドを使用して、MPLS 機能セットをインストールして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	**configure terminal**
	例：
	`switch#configure terminal`
	`switch(config)#`
		グローバル 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	**feature bgp** 例： `switch(config)#feature bgp`	BGP 機能と構成を有効にします。
ステップ 3	**feature pim** 例： `switch(config)#feature pim`	PIM 機能をイネーブルにします。
ステップ 4	**feature mvpn** 例： `switch(config)#feature mvpn`	MVPN 機能をイネーブルにします。
ステップ 5	**feature mpls l3vpn** 例： `switch(config)#feature mpls l3vpn`	MPLS レイヤ 3 VPN 機能をイネーブルにします。これにより、サイト間のユニキャスト ルートが決定されます。
ステップ 6	**feature mpls ldp** 例： `switch(config)#feature mpls ldp`	MPLS ラベル配布プロトコル（LDP）をイネーブルにします。

インターフェイスでの PIM のイネーブル化

IP マルチキャストに使用されるすべてのインターフェイスのプロトコル独立マルチキャスト（PIM）を設定することができます。バックボーンに接続されるプロバイダー エッジ（PE）ルータのすべての物理インターフェイスで PIM スパース モードに設定することをお勧めします。また、すべてのループバック インターフェイスについて、それが BGP ピアリングに使用

される場合や、その IP アドレスが PIM の RP アドレスとして使用される場合は、PIM スパース モードに設定することをお勧めします。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch#configure terminal`
	`switch(config)#`
		グローバル 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 pim sparse-mode** 例： `switch(config)#ip pim sparse-mode`	インターフェイスで PIM スパース モードをイネーブルにします。

VRF のデフォルト MDT の設定

VRF のデフォルト MDT を設定できます。

始める前に

デフォルト MDT は、同じ VPN に属するすべてのルータの設定と同じであることが必要です。送信元 IP アドレスは、BGP セッションの送信元を特定するために使用するアドレスです。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch#configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**vrf context** *VRF_NAME* 例： `switch(config)#vrf context vrf1`	VRF を設定します。
ステップ 3	**mdt default** *address* 例： `switch(config)#mdt default 232.0.0.1`	VRF に、データ MDTs のマルチキャスト アドレスの範囲を次のように設定します。 • このコマンドによって、トンネル インターフェイスが作成されます。 • デフォルトでは、トンネルヘッダーの宛先アドレスは address 引数です。

VRF の MDT SAFI の設定

デフォルトでは、VRF の MDT 後続アドレス ファミリ識別子（SAFI）が適用されます。必要に応じて、MDT SAFI をサポートしていないピアと相互運用するように MDT を構成できます。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch#configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**vrf context** *VRF_NAME*
	例：
	`switch(config)#vrf context vrf1`
	`switch(config-vrf)#`
		VRF を設定します。

ステップ 3	**no mdt enforce-bgp-mdt-safi** 例： `switch(config-vrf)#no mdt enforce-bgp-mdt-safi`	MDT SAFI をサポートしていないピアとの相互運用を可能にします。Any Source Multicast（ASM）の範囲内であるときは、初期状態ではデフォルト MDT グループの（*,G）エントリのみが読み込まれます。その後、トラフィックに基づき、（S,G）エントリは、通常の ASM ルートと同じように学習されます。 コマンドから **no** オプションを削除すると、指定された VRF に対して MDT SAFI の使用が強制されます。

MVPNs のために BGP 内の MDT アドレス ファミリ の構成

PE ルータに MDT アドレス ファミリ セッションを設定し、MVPN の MDT ピアリング セッションを確立することができます。

MDT アドレス ファミリ セッションを設定するには、ネイバー モードで **address-family ipv4 mdt** コマンドを使用してください。MDT アドレス ファミリ セッションは、BGP MDT Subaddress Family Identifier（SAFI）のアップデートを使用して PIM に送信元 PE アドレスと MDT アドレスを渡すために使用されます。

始める前に

MVPN ピアリングが MDT アドレス ファミリを介して確立できるようにするには、CE ルータに VPN サービスを提供する PE ルータで BGP ネットワークの MPLS とマルチプロトコル BGP を設定する必要があります。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch#configure terminal`
	`switch(config)#`
		グローバル 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	**feature bgp** *as-number* 例： `switch(config)#feature bgp 65635`	スイッチ 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 3	**vrf context** *VRF_NAME*
	例：
	`switch(config)#vrf context vpn1`
	`switch(config-vrf)#`
		vrf-name で識別される VPN ルーティング インスタンスを定義し、VRF コンフィギュレーション モードを開始します。vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 4	**rd** *route-distinguisher* 例： `switch(config-vrf)#rd 1.2.1`	VRF の vrf-name にルート識別子を割り当てます。route-distinguisher 引数によって、8 バイトの値が IPv4 プレフィックスに追加され、VPN IPv4 プレフィックスが作成されます。RD は、次のいずれかの形式で入力できます。 • 16 ビットまたは 32 ビットの AS 番号:32 ビットの番号。1.2:3 など。 • 32 ビットの IP アドレス:16 ビットの番号。192.0.2.1:1 など。

ステップ 5	**address-family ipv4 unicast**
	例：
	`switch(config-vrf)#address-family ipv4 unicast`
	`switch(config-vrf-af)#`
		IPv4 アドレス ファミリ タイプを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 6	**route-target import** *route-target-ext-community* 例： `switch(config-vrf-af)# route-target import 1.0.1`	VRF 用にルート ターゲット拡張コミュニティを指定します。**import** キーワードを使用すると、ルーティング情報がターゲット VPN 拡張コミュニティからインポートされます。 *route-target-ext-community* 引数により、ルート ターゲット拡張コミュニティ属性が、インポート ルート ターゲット拡張コミュニティの VRF リストに追加さ

	コマンドまたはアクション	目的
		れます。*route-target-ext-community* 引数は、次のいずれかの形式で入力できます。 • 16 ビットまたは 32 ビットの AS 番号:32 ビットの番号。1.2:3 など。 • 32 ビットの IP アドレス：16 ビットの番号。192.0.2.1:1 など。
ステップ 7	**route-target export** *route-target-ext-community* 例： `switch(config-vrf-af)# route-target export 1.0.1`	VRF 用にルート ターゲット拡張コミュニティを指定します。**export** キーワードを使用すると、ルーティング情報がターゲット VPN 拡張コミュニティからインポートされます。 *route-target-ext-community* 引数により、ルート ターゲット拡張コミュニティ属性が、インポート ルート ターゲット拡張コミュニティの VRF リストに追加されます。*route-target-ext-community* 引数は、次のいずれかの形式で入力できます。 • 16 ビットまたは 32 ビットの AS 番号:32 ビットの番号。1.2:3 など。 • 32 ビットの IP アドレス：16 ビットの番号。192.0.2.1:1 など。

ステップ 8	**router bgp** *as-number*
	例：
	`switch(config)#router bgp 1.1`
	`switch(config-router)#`
		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。as-number 引数は、ルータを他の BGP ルータに対して識別し、転送するルーティング情報にタグを設定する自律システムの番号を示します。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

ステップ 9	**address-family ipv4 mdt** 例： `switch(config-router)#address-family ipv4 mdt`	IPv4 MDT 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 10	**address-family** {**vpn4**} [**unicast**]
	例：
	`switch(config-router-af)# address-family vpnv4`
	`switch(config-router-af)#`
		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、標準 VPNv4 または VPNv6 アドレス プレフィックスを使用する、BGP などのルーティング セッションを設定します。**unicast** キーワード（任意）では、VPNv4 または VPNv6 ユニキャスト アドレス プレフィックスを指定します。

ステップ 11	**address-family** {**ipv4**} **unicast**
	例：
	`switch(config-router-af)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		標準 IPv4 または VPNv6 アドレス プレフィックスを使用するルーティング セッションを設定するために、アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 12	**neighbor** *neighbor-address* 例： `switch(config-switch-af)# neighbor 192.168.1.1`	ネイバー コンフィギュレーション モードを開始します。
ステップ 13	**update source** *interface* 例： `switch(config-switch-neighbor)# update-source loopback 1`	アップデート ソースを loopback1 に設定します。
ステップ 14	**address-family ipv4 mdt** 例： `switch(config-router-neighbor)# address-family ipv4 mdt`	アドレス ファミリ コンフィギュレーションを開始し、IP MDT アドレス ファミリ セッションを作成します。
ステップ 15	**send-community extended** 例： `switch(config-router-neighbor-af)#send-community extended`	拡張コミュニティ属性が BGP ネイバーに送信されるように指定します。
ステップ 16	**show bgp** {**ipv4**} **unicast neighbors vrf***VRF_NAME* 例： `switch(config-router-neighbor-af)#show bgp ipv4 unicast neighbors vrf vpn1`	BGP ネイバーに関する情報を表示します。vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。
ステップ 17	**copy running-config startup-config** 例： `switch(config-router-neighbor-af)#copy running-config startup-config`	実行設定を、スタートアップ設定にコピーします。

データ MDT の構成

データ MDT を設定できます。データ MDT の作成に使用されるマルチキャスト グループは、設定済み IP アドレスのプールからダイナミックに選択されます。ストリームの数が PE 単位、VRF 単位の MDT より大きい場合、複数のストリームが同じデータ MDT を共有します。

始める前に

データ MDT を設定する前に、VRF のデフォルト MDT を設定する必要があります。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch#configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**vrf context** *VRF_NAME* 例： `switch#ip vrf vrf1`	VRF コンフィギュレーション モードを開始し、VRF 名を割り当てることにより VPN ルーティング インスタンスを定義します。

ステップ 3	**mdt data** *prefix* [**immediate-switch**] [**route-map** *policy-name*]
	例：
	`switch(config-vrf)# mdt data 225.1.1.1/32 immediate-switch route-map test`
	例：
	`switch(config-vrf)# mdt data 225.1.1.1/32 route-map test`
		次のように値の範囲を指定します。
		• *prefix* は、データ MDT プールで使用されるアドレスの範囲を指定します。
		• *policy-name* は、データ MDT への切り替えで考慮されるカスタマーデータ ストリームを定義するポリシー ファイルを定義します。
		（注） このコマンドは、immediate-switch オプションの有無にかかわらず同じ効果があります。

ステップ 4	**exit** 例： `switch(config)#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。

MVPN 構成の検証

MVPN の設定を表示するには、次のいずれかの作業を行います。

表 *8 : MVPN* の設定の確認

コマンド	目的
show interface	インターフェイスの詳細を表示します。
show ip mroute vrf	マルチキャスト ルートを表示します。
show ip pim event-history mvpn	MVPN のイベント履歴ログの詳細を表示します。
show ip pim mdt	MVPN によって作成された MTI トンネルの詳細を表示します。
show ip pim mdt receive vrf vrf-name	カスタマー ソース、データ MDT 送信元へのカスタマー グループ、および受信側のデータ MDT グループそれぞれのマッピングを表示します。
show ip pim mdt send vrf vrf-name	カスタマー ソース、データ MDT 送信元へのカスタマー グループ、および送信側のデータ MDT グループそれぞれのマッピングを表示します。
show ip pim neighbor	確立されたPIMネイバーの詳細を表示します。
show ip route detail	ユニキャスト ルーティング テーブルの詳細を表示します。
show mvpn bgp mdt-safi	MVPN の BGP MDT SAFI データベースを表示します。
show mvpn mdt encap vrf *vrf*	MVPN のカプセル化テーブルを表示します。このテーブルは、デフォルト vrf で MVPN パケットを送信するときにカプセル化する方法を示しています。
show mvpn mdt route	デフォルトおよびMDTルートの詳細を表示します。このデータは、デフォルト VRF でカスタマー データと制御トラフィックを送信する方法を決定します。
show routing [**ip**] **multicast mdt encap**	MRIB のカプセル化テーブルを表示します。このテーブルは、デフォルト vrf で MVPN パケットを送信するときにカプセル化する方法を示しています。

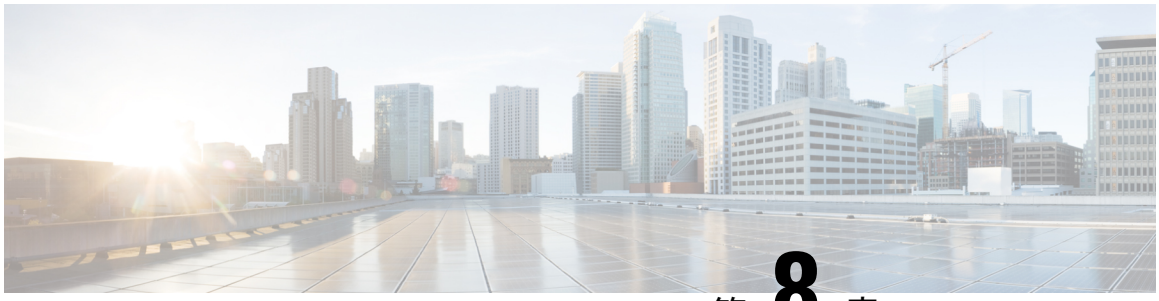
MVPN の構成例

次に、MVPN の設定例を 2 つのコンテキストで示します。

```
vrf context vpn1
  ip pim rp-address 10.10.1.2 -list 224.0.0.0/8
  ip pim ssm range 232.0.0.0/8
  rd auto
  mdt default 232.1.1.1
  mdt source loopback1
  mdt data 225.122.111.0/24 immediate-switch
vrf context vpn4
  ip pim rp-address 10.10.4.2 -list 224.0.0.0/8
  ip pim ssm range 232.0.0.0/8
  mdt default 235.1.1.1
  mdt asm-use-shared-tree
ip pim rp-address 10.11.0.2 -list 224.0.0.0/8
ip pim rp-address 10.11.0.4 -list 235.0.0.0/8
ip pim ssm range 232.0.0.0/8
```

次に、「blue」と名づけられた VRF を VPN ルーティング インスタンスに割り当てる方法の例を示します。VPN VRF の MDT デフォルトは 10.1.1.1、MDT のマルチキャスト アドレスの範囲は 10.1.2.0（ワイルドカード ビットが 0.0.0.3）です。

```
Vrf context blue
mdt data 225.122.111.0/24 immediate-switch
```



第 8 章

InterAS オプション B

この章では、さまざまな InterAS オプション B 構成オプションについて説明します。使用可能なオプションは、InterAS オプション B、InterAS オプション B（RFC 3107 による）、および InterAS オプション B ライトです。InterAS オプション B（RFC 3107 による）の実装により、データセンターと WAN 間の完全な IGP 分離が保証されます。BGP が特定のルートを ASBR にアドバタイズすると、そのルートにマップされたラベルも配布されます。

InterASに関する情報

自律システム（AS）とは、共通のシステム管理グループによって管理され、単一の明確に定義されたプロトコルを使用している単一のネットワークまたはネットワークのグループのことです。多くの場合、仮想プライベート ネットワーック（VPN）は異なる地理的領域の異なる AS に拡張されます。一部の VPN は、複数のサービスプロバイダにまたがって拡張する必要があり、それらはオーバーラッピング VPN と呼ばれます。VPN の複雑さや場所に関係なく、AS 間の接続はお客様に対してシームレスである必要があります。

InterAS と ASBR

異なるサービス プロバイダーの異なる AS は、VPN-IP アドレスの形式で情報を交換することによって通信できます。ASBR は、EBGP を使用してその情報を交換します。IBGP は、各 VPN および各 AS 内の IP プレフィックスのネットワーク層情報を配布します。ルーティング情報は、次のプロトコルを使用して共有されます。

- AS 内では、ルーティング情報は IBGP を使用して共有されます。
- AS 間では、ルーティング情報は EBGP を使用して共有されます。EBGP を使用することで、サービス プロバイダーは、別の AS 間でのルーティング情報のループフリー交換を保証するインタードメイン ルーティング システムをセットアップできます。

EBGP の主な機能は、AS ルートのリストに関する情報を含む、AS 間のネットワーク到達可能性情報を交換することです。AS は、EBGP ボーダー エッジ ルータを使用してラベルスイッチング情報を含むルートを配布します。各ボーダー エッジ ルータでは、ネクスト ホップおよび MPLS ラベルが書き換えられます。

この MPLS VPN における InterAS 設定には、プロバイダー間 VPN を含めることができます。これは、異なるボーダーエッジルータで接続されている 2 つ以上の AS を含む、MPLS VPNです。AS は EBGP を使用してルートを交換します。IBGP やルーティング情報は AS 間では交換されません。

VPN ルーティング情報の交換

AS は、接続を確立するために VPN ルーティング情報（ルートとラベル）を交換します。AS 間の接続を制御するために、PE ルータおよび EBGP ボーダー エッジ ルータはラベル転送情報ベース（LFIB）を保持します。LFIB では、VPN 情報の交換中に PE ルータおよび EBGP ボーダー エッジ ルータが受信するラベルとルートが管理されます。

ASでは、次の注意事項に基づいて VPN ルーティング情報を交換します。

- ルーティング情報に次の内容が含まれています。
  - 接続先ネットワーク。
  - 配布元ルータに関連付けられたネクストホップ フィールド。
  - ローカル MPLS ラベル
- ルート識別子（RD1）は、接続先ネットワーク アドレスの一部として含まれています。ルート識別子によって、VPN-IP ルートは VPN サービス プロバイダー環境内でグローバルに一意となります。

ASBR は、IBGP ネイバーに VPN-IPv4 NLRI を送信する場合に、ネクスト ホップを変更するように設定されています。したがって、ASBR では、IBGP ネイバーに NLRI を転送する場合に新しいラベルを割り当てる必要があります。

InterAS オプション

Nexus 3600 シリーズ スイッチは、次の InterAS オプションをサポートします。

- **InterAS オプション A** - Inter-AS オプション A ネットワークでは、自律システム境界ルータ（ASBR）ピアは複数のサブインターフェイスによって接続され、2 つの自律システムにまたがるインターフェイス VPN が少なくとも 1 つ設定されます。これらの ASBR では、各サブインターフェイスが、VPN ルーティングおよび転送（VRF ）インスタンスおよびラベル付けされていない IP プレフィックスのシグナリング用の BGP セッションに関連付けられます。その結果、バックツーバック VRF 間のトラフィックは IP になります。このシナリオでは、各 VPN は相互に分離されます。また、トラフィックが IP であるため、IP トラフィック上で動作する Quality of Service（QoS）メカニズムを維持できます。この設定の欠点は、サブインターフェイスごとに 1 つの BGP セッションが必要となることです（VPN ごとに少なくとも 1 つのサブインターフェイスも必要となります）。このことは、ネットワークの規模が大きくなるにつれて、スケーラビリティに関する問題が発生する原因となります。

- **InterAS オプション B** - InterAS オプション B ネットワークでは、ASBR ポートは、MPLS トラフィックを受信できる 1 つ以上のインターフェイスによって接続されます。マルチプロトコル ボーダー ゲートウェイ プロトコル（MP-BGP）セッションは、ASBR 間でのラベル付き VPN プレフィックスを配布します。その結果、ASBR の間のトラフィック フローにはラベルが付きます。この設定の欠点は、トラフィックが MPLS であるため、IP トラフィックにのみ適用される QoS メカニズムを伝えることができず、VRF を分離することもできないことです。InterAS オプション B は、ASBR 間のすべての VPN プレフィックスを交換するために 1 つの BGP セッションしか必要としないため、オプション A よりも拡張性に優れています。また、この機能はノンストップ フォワーディング（NSF）とグレースフル リスタートを提供します。このオプションでは、ASBR を直接接続する必要があります。

オプション B のいくつかの機能を以下に示します。

- AS 内の Nexus 3600 シリーズ スイッチ間で IBGP VPNv4/v6 セッションを持つことができ、データセンター エッジ ルータと WAN ルータの間で EBGP VPNv4/v6 セッションを持つことができます。

- ライト バージョンのように、データセンター エッジ ルータ間の VRF IBGP セッションごとの要件はありません。

- – LDP は ASBR 間で IGP ラベルを配布します。

- **InterAS オプション B（BGP-3107 または RFC 3107 実装）**

- AS 内の Nexus 3600 プラットフォーム スイッチ間で IBGP VPNv4/v6 実装を持つことができ、データセンター エッジ ルータと WAN ルータの間で EBGP VPNv4/v6 セッションを持つことができます。

- BGP-3107 により、BGP パケットは ASBR 間で LDP を使用せずにラベル情報を伝送できます。
- 特定の 1 つのルートに対するラベル マッピング情報は、ルート自体の配布に使用される、同じ BGP アップデート メッセージにピギーバッグにより同梱されます。
- 特定のルートへの配布に BGP を使用する場合は、このルートにマッピングされている MPLS ラベルも配布されます。多くの ISP は、データ センター間の完全な IGP 分離が保証されるため、この構成方法を好みます。
- **InterAS オプション B ライト** – InterAS オプション B 機能のサポートは、Cisco NX-OS 6.2(2) リリースでは制限されています。ライト詳細は、「InterAS オプション B（ライト バージョン）の構成」セクションに記載されています。

InterAS オプション B の設定に関する注意事項と制限事項

InterAS オプション B 機能は、BGP コンフェデレーション AS ではサポートされません。ただし、オプション B の実装は Cisco Nexus 3600 プラットフォーム スイッチでサポートされています。

InterAS オプション B のスイッチの構成

スイッチの特定の機能を有効にして、InterAS オプション B を実行します。

始める前に

install feature-set mpls コマンドは、デフォルトの VDC でのみ使用でき、デフォルトの VDC で有効にする必要があります。

DC エッジ スイッチで VRF を次の手順に従って構成します。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル 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	**install feature-set mpls** 例：	デフォルト VDC に MPLS 機能セットをインストールします。

	コマンドまたはアクション	目的
	`switch(config)# install feature-set mpls`	（注） デフォルトの VDC では、MPLS のみをインストールして有効にすることができます。このコマンドの「no」形式を使用して、MPLS 機能セットをアンインストールします。
ステップ 3	**feature mpls ldp** 例： `switch(config)# feature mpls ldp`	デバイスの MPLS LDP 機能を有効にします。 （注） デバイスで MPLS LDP 機能がディセーブルになっていると、LDP コマンドを使用できません。
ステップ 4	**feature mpls l3vpn** 例： `switch(config)# feature mpls l3vpn`	MPLS レイヤ 3 VPN 機能をイネーブルにします。
ステップ 5	**feature bgp** 例： `switch(config)# feature bgp`	BGP 機能をイネーブルにします。
ステップ 6	**vrf-context** *vrf-name* 例： `switch(config)# vrf context VPN1`	VRF 名を割り当て、VRF コンフィギュレーションモードを開始することにより、VPN ルーティング インスタンスを定義します。 vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。
ステップ 7	**rd** *route-target-ext-community* 例： `switch(config-vrf)# rd100:1`	ルート識別子を設定します。 route-distinguisher 引数によって、8 バイトの値が IPv4 プレフィックスに追加され、VPN IPv4 プレフィックスが作成されます。
ステップ 8	**address-family {ipv4 | ipv6} unicast** 例： `switch(config-vrf)# address-family ipv4 unicast`	IPv4 または IPv6 アドレス ファミリ タイプを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 9	**route-target {import | export}** *route-target-ext-community* 例： `switch(config-vrf-af-ip4)# route-target import 1:1`	次のように VRF 用にルート ターゲット拡張コミュニティを指定します。 • import キーワードを使用すると、ターゲット VPN 拡張コミュニティからルーティング情報がインポートされます。 • export キーワードを使用すると、ルーティング情報がターゲット VPN 拡張コミュニティにエクスポートされます。 • route-target-ext-community 引数により、ルート ターゲット拡張コミュニティ属性が、インポート、またはエクスポートのルート ターゲット拡張コミュニティの VRF リストに追加されます。
ステップ 10	**copy running-config startup-config** 例： `switch(config-vrf-af-ip4)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

InterAS オプション B の BGP の設定

次の手順で、IBGP および EBGP VPNv4/v6 を使用して DC エッジ スイッチを構成します。

始める前に

InterAS オプション B の BGP を構成するには、IBGP 側と EBGP 側の両方でこの構成を有効にする必要があります。参考図 1 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	**configure terminal** 例： `switch# configure terminal switch(config)#`	グローバル 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 bgp** *as-number* 例： `switch(config)# router bgp 100`	ルータ BGP コンフィギュレーション モードを開始し、ローカル BGP スピーカ デバイスに自律システム番号（AS）を割り当てます。
ステップ 3	**neighbor** *ip-address* 例： `switch(config-router)# neighbor 10.0.0.2`	BGP 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、ルータ BGP 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	**remote-as** *as-number* 例： `switch(config-router-neighbor)# remote-as 200`	as-number 引数には、ネイバーが属している自律システムを指定します。
ステップ 5	**address-family {vpnv4 | vpnv6} unicast** 例： `switch(config-router-neighbor)# address-family vpnv4 unicast`	IP VPN セッションを設定するために、アドレス ファミリ コンフィギュレーション モードに入ります。
ステップ 6	**send-community {both | extended}** 例： `switch(config-router-neighbor-af)# send-community both`	コミュニティ属性が両方の BGP ネイバーに送信されるように指定します。
ステップ 7	**retain route-target all** 例： `switch(config-router-neighbor-af)# retain route-target all`	(オプション)。VRF 設定なしで ASBR で VPNv4/v6 アドレス設定を保持します。 （注） ASBR に VRF 設定がある場合、このコマンドは必要ありません。
ステップ 8	**vrf** *vrf-name* 例： `switch(config-router-neighbor-af)# vrf VPN1`	BGP プロセスを VRF に関連付けます。
ステップ 9	**address-family {ipv4 | ipv6} unicast** 例： `switch(config-router-vrf)# address-family ipv4 unicast`	IPv4 または 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。
ステップ 10	**exit** 例： `switch(config-vrf-af)# exit`	IPv4 アドレス ファミリを終了します。

	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： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

InterAS オプション B のスイッチの構成（RFC 3107 実装による）

スイッチの特定の機能を有効にして、InterAS オプション B を実行します。

始める前に

DC エッジ スイッチで VRF を次の手順に従って構成します。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル 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	**install feature-set mpls** 例： `switch(config)# install feature-set mpls`	デフォルト VDC に MPLS 機能セットをインストールします。 （注） デフォルトの VDC では、MPLS のみをインストールして有効にすることができます。このコマンドの「no」形式を使用して、MPLS 機能セットをアンインストールします。
ステップ 3	**feature mpls ldp** 例： `switch(config)# feature mpls ldp`	デバイスの MPLS LDP 機能を有効にします。 （注） デバイスで MPLS LDP 機能がディセーブルになっていると、LDP コマンドを使用できません。

	コマンドまたはアクション	目的
ステップ 4	**feature mpls l3vpn** 例： `switch(config)# feature mpls l3vpn`	MPLS レイヤ 3 VPN 機能をイネーブルにします。
ステップ 5	**feature bgp** 例： `switch(config)# feature bgp`	BGP 機能をイネーブルにします。
ステップ 6	**vrf-context** *vrf-name* 例： `switch(config)# vrf context VPN1`	VRF 名を割り当て、VRF コンフィギュレーション モードを開始することにより、VPN ルーティング インスタンスを定義します。 vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。
ステップ 7	**rd** *route-distinguisher* 例： `switch(config-vrf)# rd100:1`	ルート識別子を設定します。 route-distinguisher 引数によって、8 バイトの値が IPv4 プレフィックスに追加され、VPN IPv4 プレフィックスが作成されます。
ステップ 8	**address-family {ipv4 | ipv6} unicast** 例： `switch(config-vrf)# address-family ipv4 unicast`	IPv4 または IPv6 アドレス ファミリ タイプを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。
ステップ 9	**route-target {import | export}** *route-target-ext-community* 例： `switch(config-vrf-af-ip4)# route-target import 1:1`	次のように VRF 用にルート ターゲット拡張コミュニティを指定します。 • import キーワードを使用すると、ターゲット VPN 拡張コミュニティからルーティング情報がインポートされます。 • export キーワードを使用すると、ルーティング情報がターゲット VPN 拡張コミュニティにエクスポートされます。 • route-target-ext-community 引数により、ルート ターゲット拡張コミュニティ属性が、インポート、またはエクスポートのルート ターゲッ

	コマンドまたはアクション	目的
		ト拡張コミュニティの VRF リストに追加されます。
ステップ 10	**copy running-config startup-config** 例： `switch(config-vrf-af-ip4)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

InterAS オプション B の BGP の設定（RFC 3107 実装による）

次の手順で、IBGP および EBGP VPNv4/v6 と BGP ラベル付きユニキャスト ファミリを使用して DC エッジ スイッチを構成します。

始める前に

正しい VDC を開始していること（または **switchto vdc** コマンドを使用済みであること）を確認してください。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**router bgp** *as-number* 例： `switch(config)# router bgp 100`	ルータ BGP コンフィギュレーション モードを開始し、ローカル BGP スピーカ デバイスに自律システム番号（AS）を割り当てます。
ステップ 3	**address-family {vpnv4 | vpnv6} unicast** 例： `switch(config-router-neighbor)# address-family vpnv4 unicast`	IP VPN セッションを設定するために、アドレス ファミリ コンフィギュレーション モードに入ります。
ステップ 4	**redistribute direct route-map** *tag* 例： `switch(config-router-af)# redistribute direct route-map loopback`	ボーダー ゲートウェイ プロトコルを使用して、接続されたルートを直接再配布します。

	コマンドまたはアクション	目的
ステップ 5	**allocate-label all** 例： `switch(config-router-af)# allocate-label all`	接続されたインターフェイスのラベルをアドバタイズするために、BGP ラベル付きユニキャストアドレスファミリを持つ ASBR を設定します。
ステップ 6	**exit** 例： `switch(config-router-af)# exit`	アドレスファミリ ルータ コンフィギュレーションモードを終了して、ルータ BGP 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	**neighbor** *ip-address* 例： `switch(config-router)# neighbor 10.1.1.1`	BGP ネイバーの IP アドレスを構成し、ルータ BGP ネイバー 構成 モードを開始します。
ステップ 8	**remote-as** *as-number* 例： `switch(config-router-neighbor)# remote-as 100`	BGP ネイバーの AS 番号を指定します。
ステップ 9	**address-family {ipv4|ipv6} labeled-unicast** 例： `switch(config-router-neighbor)# address-family ipv4 labeled-unicast`	接続されたインターフェイスのラベルをアドバタイズするために、BGP ラベル付きユニキャストアドレスファミリを持つ ASBR を設定します。 （注） これは、RFC 3107 を実装するコマンドです。
ステップ 10	**retain route-target all** 例： `switch(config-router-neighbor-af)# retain route-target all`	(オプション)。VRF 設定なしで ASBR で VPNv4/v6 アドレス設定を保持します。 （注） ASBR に VRF 設定がある場合、このコマンドは必要ありません。
ステップ 11	**exit** 例： `Switch(config-router-neighbor-af)# exit`	ルータ BGP ネイバー アドレス ファミリ 構成モードを終了し、BGP構成モードに戻ります。
ステップ 12	**neighbor** *ip-address* 例： `switch(config-router)# neighbor 10.1.1.1`	ループバック IP アドレスを構成し、ルータ BGP ネイバー 構成 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 13	**remote-as** *as-number* 例： `switch(config-router-neighbor)# remote-as 100`	BGP ネイバーの AS 番号を指定します。
ステップ 14	**address-family {vpnv4|vpnv6} unicast** 例： `switch(config-router-vrf)# address-family ipv4 unicast`	BGP VPNv4 ユニキャスト アドレス ファミリで ASBR を設定します。
ステップ 15	**exit** 例： `switch(config-vrf-af)# exit`	IPv4 アドレス ファミリを終了します。
ステップ 16	**address-family {vpnv4|vpnv6} unicast** 例： `switch(config-router-vrf)# address-family ipv4 unicast`	BGP VPNv4 ユニキャスト アドレス ファミリで ASBR を設定します。
ステップ 17	**Repeat the process with ASBR2**	オプション B（RFC 3107）設定で ASBR2 を設定し、2 箇所のデータセンター DC1 と DC2 間の完全な IGP 分離を実装します。
ステップ 18	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

ASBR 間の LDP 接続をフィルタ処理するための ACL の作成（RFC 3107 導入）

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `switch# configure terminal switch(config)#`	グローバル構成モードを開始します。

	コマンドまたはアクション	目的
ステップ 2	**ip access-list** *name* 例： `switch(config)# ip access-list LDP`	アクセス リストを作成し、ACL 構成モードを開始します。
ステップ 3	**[** sequence-number **]deny tcp any any eq** *packet-length* 例： `switch(config-acl)# 10 deny tcp any any eq 646`	指定されたシーケンスに従ってACL命令を実行します。
ステップ 4	**[** sequence-number **] deny tcp any eq** *packet-length* **any** 例： `switch(config-acl)# 20 deny tcp any eq 646 any`	指定されたシーケンスに従ってACL命令を実行します。
ステップ 5	**[** sequence-number **] deny udp any any eq** *packet-length* 例： `switch(config-acl)# 30 deny udp any any eq 646`	指定されたシーケンスに従ってACL命令を実行します。
ステップ 6	**[** sequence-number **] deny udp any eq** *packet-length* **any** 例： `switch(config-acl)# 20 deny udp any eq 646 any`	指定されたシーケンスに従ってACL命令を実行します。
ステップ 7	**[** sequence-number **] permit ip any any** 例： `switch(config-acl)# 50 permit ip any any`	指定されたシーケンスに従ってACL命令を実行します。
ステップ 8	**exit** 例： `switch(config-acl)# exit`	ACL構成モードを終了し、グローバル構成モードを開始します。
ステップ 9	**interface** *type number* 例： `switch(config)# interface ethernet 2/20`	インターフェイス構成モードを開始します。
ステップ 10	**mpls ip** 例： `switch(config-if)# mpls ip`	インターフェイスに対して MPLS ホップバイホップ転送を設定します。

	コマンドまたはアクション	目的
ステップ 11	**ip access-group** *name* **in** 例： `switch(config-if)# ip access-group LDP in`	インターフェイスの着信トラフィックにACL（前の手順で作成した名前付き LDP）を適用することを指定します。
ステップ 12	**ip access-group** *name* **out** 例： `switch(config-if)# ip access-group LDP out`	インターフェイスのアウトバウンドトラフィックにACL（前の手順で作成した名前付き LDP）を適用することを指定します。
ステップ 13	**end** 例： `switch(config-if)# end`	インターフェイス構成モードを終了し、特権 EXEC モードに戻ります。

InterAS オプション B（ライト バージョン）の構成

[InterAS オプション B ライトの構成に関する注意事項と制限事項（Guidelines and Limitations for Configuring InterAS Option B lite）]

- アグリゲーション スイッチはローカル VRF のみをサポートし、自律システム（AS）内の Nexus デバイスは VRF 実装を介して接続されます。
- IBGP ピアから学習したルートは EBGP ピアに送信されず、EBGP ピアから学習したルートは IBGP VPNv4/VPNv6 ピアに送信されません。
- EBGP 側の MP-BGP を使用した interAS オプション B は、IBGP 側の MP-BGP では機能しません。1 つのインターフェイスはコアに接続し、もう 1 つのインターフェイスはレイヤ 3 VPN に接続します。
- MP-BGP レイヤ 3 VPN は AS 内では機能しません。

InterAS オプション B（ライト バージョン）のスイッチの構成

スイッチの特定の機能を有効にして、interAS オプション B を実行します。

始める前に

install feature-set mpls コマンドは、デフォルトの VDC でのみ使用でき、デフォルトの VDC で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	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル 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	**install feature-set mpls** 例： `switch(config)# install feature-set mpls`	デフォルト VDC に MPLS 機能セットをインストールします。 （注） デフォルトの VDC では、MPLS のみをインストールして有効にすることができます。このコマンドの no 形式を使用して、MPLS 機能セットをアンインストールします。
ステップ 3	**feature mpls ldp** 例： `switch(config)# feature mpls ldp`	デバイスの MPLS LDP 機能をイネーブルにします。デバイスで MPLS LDP 機能がディセーブルになっていると、LDP コマンドを使用できません。
ステップ 4	**feature mpls l3vpn** 例： `switch(config)# feature mpls l3vpn`	MPLS レイヤ 3 VPN 機能をイネーブルにします。
ステップ 5	**feature bgp** 例： `switch(config)# feature bgp`	BGP 機能をイネーブルにします。
ステップ 6	**vrf-context** *vrf-name* 例： `switch(config)# vrf-context VPN1`	VRF 名を割り当て、VRF コンフィギュレーション モードを開始することにより、VPN ルーティング インスタンスを定義します。vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。
ステップ 7	**rd** *route-distinguisher* 例： `switch(config-vrf)# rd 100:1`	ルート識別子を設定します。route-distinguisher 引数によって、8 バイトの値が IPv4 プレフィックスに追加され、VPN IPv4 プレフィックスが作成されます。

	コマンドまたはアクション	目的
ステップ 8	**address-family {ipv4 | ipv6} unicast** 例： `switch(config-vrf)# address-family ipv4 unicast`	IPv4 または IPv6 アドレス ファミリ タイプを指定し、アドレスファミリ コンフィギュレーションモードを開始します。
ステップ 9	**route-target {import | export}** *route-target-ext-community* 例： `switch(config-vrf-af-ip4)# route-target import 1:1`	次のように VRF 用にルート ターゲット拡張コミュニティを指定します。 （注） • **import** キーワードを使用すると、ルーティング情報がターゲット VPN 拡張コミュニティからインポートされます。 • **export** キーワードを使用すると、ルーティング情報がターゲット VPN 拡張コミュニティにエクスポートされます。 • *route-target-ext-community* 引数により、ルートターゲット拡張コミュニティ属性が、インポート、またはエクスポートのルートターゲット拡張コミュニティの VRF リストに追加されます。
ステップ 10	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

InterAS オプション B（ライト バージョン）のための BGP の構成

DC エッジ スイッチで EBGP VPNv4/v6 を次の手順を使用して構成します。

手順

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

	コマンドまたはアクション	目的
	`switch# configure terminal` `switch(config)#`	
ステップ 2	**router bgp** *as-number* 例： `switch(config)# router bgp 100`	ルータ BGP コンフィギュレーション モードを開始し、ローカル BGP スピーカ デバイスに自律システム番号（AS）を割り当てます。
ステップ 3	**neighbor** *ip-address* 例： `switch(config-router)# neighbor 10.0.0.2`	BGP 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、ルータ BGP 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	**remote-as** *as-number* 例： `switch(config-router-neighbor)# remote-as 200`	as-number 引数には、ネイバーが属している自律システムを指定します。
ステップ 5	**address-family {vpnv4 | vpnv6} unicast** 例： `switch(config-router-neighbor)# address-family vpnv4 unicast`	IP VPN セッションを設定するために、アドレス ファミリ コンフィギュレーション モードに入ります。
ステップ 6	**send-community {both | extended}** 例： `switch(config-router-neighbor-af)# send-community both`	コミュニティ属性が両方の BGP ネイバーに送信されるように指定します。
ステップ 7	**vrf** *vrf-name* 例： `switch(config-router-neighbor-af)# vrf VPN1`	BGP プロセスを VRF に関連付けます。
ステップ 8	**address-family {ipv4 | ipv6} unicast** 例： `switch(config-router-vrf)# address-family ipv4 unicast`	IPv4 または 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。
ステップ 9	**exit** 例： `switch(config-vrf-af)# exit`	IPv4 アドレス ファミリを終了します。
ステップ 10	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

MPLS VPN InterAS オプションの構成の確認

InterAS オプション B の設定情報を確認するには、次のいずれかの作業を行います。

コマンド	目的
show bgp { vpnv4 | vpnv6 } unicast [*ip-prefix/length* [**neighbors** *neighbor*]] {**vrf**{*vrf-name* | **all** } | **rd** *route-distinguisher* }	BGP テーブルからの VPN ルートを表示します。
show bgp ipv6 unicast [**vrf***vrf-name*]	6VPE の VRF での BGP に関する情報を表示します。
show forwarding { ip | ipv6 } route vrf *vrf-name*	VRF に関連付けられた IP 転送テーブルを表示します。ローカル CE ルータとリモート CE ルータのループバック アドレスが、PE ルータのルーティング テーブルに存在することを確認します。
show { ip | ipv6 } bgp[**vrf** *vrf-name*]	VRF での BGP に関する情報を表示します。
show ip route[*ip-address* [mask]] [*protocol*] **vrf** *vrf-name*	ルーティング テーブルの現在の状態を表示します。ip-address 引数を使用して、CE1 に CE2 へのルートが含まれていることを確認します。CE1 から学習したルートを確認します。CE2 へのルートがリストされていることを確認します。
show {**ip** | **ipv6**} **routevrf** *vrf-name*	VRF に関連付けられた IP ルーティング テーブルを表示します。ローカル CE ルータとリモート CE ルータのループバック アドレスが、PE ルータのルーティング テーブルに存在することを確認します。
show running-config bgp	BGP の実行コンフィギュレーションを表示します。
show running-config vrf *vrf-name*	VRF の実行コンフィギュレーションを表示します。
show vrf *vrf-name* **interface** *if-type*	VRF に対して設定されるルート識別子（RD）およびインターフェイスを検証します。
trace *trace destination* **vrf** *vrf-name*	パケットがその宛先に送信されるときに取るルートを検出します。トレース コマンドは、2 つのルータが通信できない場合に問題の箇所を分離するのに役立ちます。

構成 InterAS オプション B の構成例

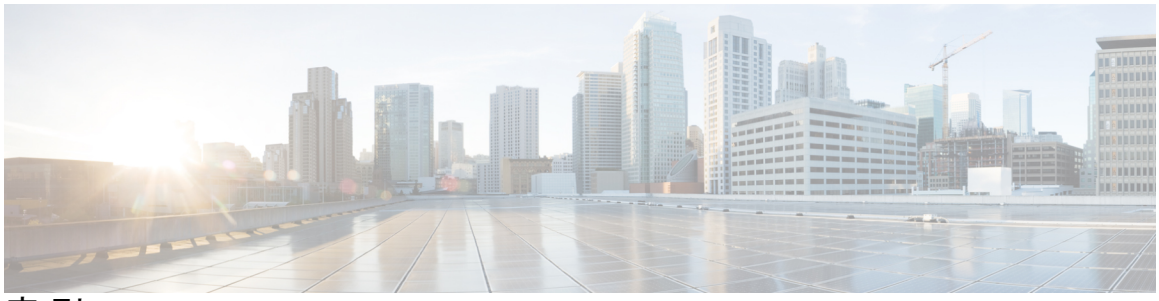
この例は、InterAS オプション B を構成する方法を表示ています

```
!--Configure VRFs on the DC edge switches --!

configure terminal
install feature-set mpls
feature mpls ldp
feature mpls l3vpn
feature bgp
vrf context VPN1
rd 100:1
address-family ipv4 unicast
route-target import 1:1
copy running-config startup-config

!--Configure DC Edge switches with IBGP & EBGP VPNv4/v6 --!

configure terminal
router bgp 100
neighbor 10.0.0.2
remote-as 200
address-family vpnv4 unicast
send-community both
retain route-target all
vrf VPN1
address-family ipv4 unicast
exit
copy running-config startup-config
```

この例は、InterAS オプション B（RFC 3107）を構成する方法を示しています。

```
!--Configure VRFs on the DC edge switches --!

configure terminal
install feature-set mpls
feature mpls ldp
feature mpls l3vpn
feature bgp
vrf context VPN1
rd 100:1
address-family ipv4 unicast
route-target import 1:1
copy running-config startup-config

!--Configure DC Edge switches with IBGP & EBGP VPNv4/v6 --!

configure terminal
router bgp 100
address-family ipv4 unicast
redistribute direct route-map loopback
allocate-label all
exit
neighbor 10.1.1.1
remote-as 100
address-family ipv4 labeled-unicast
retain route-target all
exit
```

```
neighbor 1.1.1.1
remote-as 100
address-family vpnv4 unicast
address-family vpnv6 unicast
!--Repeat the process with ASBR2. --!
copy running-config startup-config

!--Creating an ACL to filter LDP connection between the ASBRs (RFC 3107 implementation)--!


configure terminal
ip access-list LDP
10 deny tcp any any eq 646
20 deny tcp any eq 646 any
30 deny udp any any eq 646
40 deny udp any eq 646 any
50 permit ip any any
exit
interface ethernet 2/20
mpls ip
ip access-group LDP in
ip access-group LDP out
end
```



索 引

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。