Cisco Nexus 9000 シリーズ NX-OS ラベル スイッチング構成ガイド、リリース 10.4(x)

初版：2023 年 8 月 18 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。

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、ユーザが値を指定する引数です。
[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、角かっこで囲み、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ないでください。引用符を使用すると、その引用符も含めて string と見なされます。

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 screen 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 9000 シリーズ スイッチの関連資料

Cisco Nexus 9000 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。

http://www.cisco.com/en/US/products/ps13386/tsd_products_support_series_home.html

マニュアルに関するフィードバック

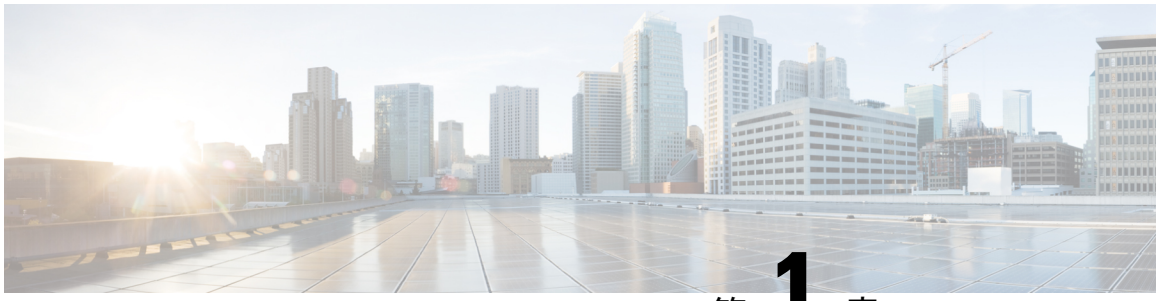
このマニュアルに関する技術的なフィードバック、または誤りや記載もれなどお気づきの点がございましたら、HTML ドキュメント内のフィードバック フォームよりご連絡ください。ご協力をよろしくお願いいたします。

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.



第 1 章

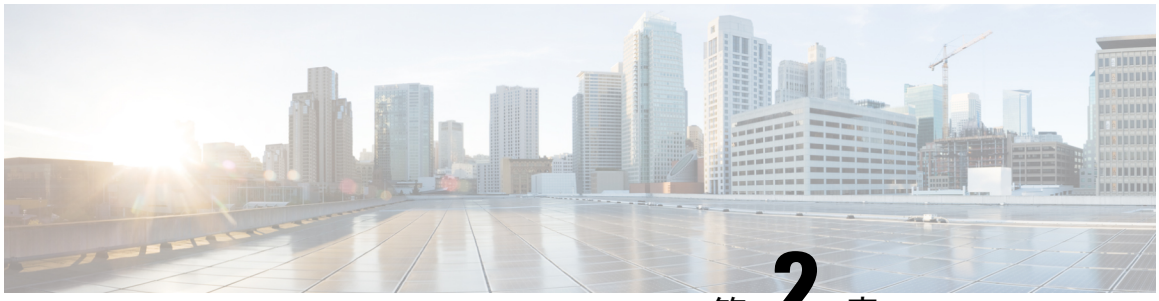
新機能と更新情報

新機能と更新情報

表 *1* : 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
MPLS タグ付きトラフィックのポートチャネル ロードバランス コマンド	Cisco Nexus 9300 -EX/FX/FX2/ FX3/GX/GX2 TOR および EOR プラットフォーム スイッチにサポートを追加しました。	10.4(1)F	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 （137 ページ）

新機能と更新情報



第 2 章

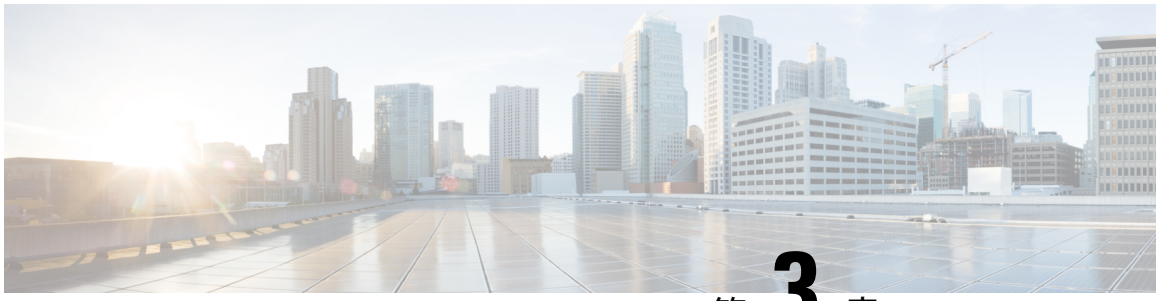
概要

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* ライセンス ガイド』および『*Cisco NX-OS* ライセンス オプション ガイド』を参照してください。

サポートされるプラットフォーム

Cisco NX-OS リリース 7.0(3)I7(1) 以降、「Nexus スイッチ プラットフォーム サポート マトリクス」を使用して、選択した機能をサポートするさまざまな Cisco Nexus 9000 および 3000 スイッチのリリース元である Cisco NX-OS を知ることができます。



第 3 章

静的 MPLS の設定

この章では、静的なマルチプロトコル ラベル スイッチング（MPLS）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* ライセンス ガイド』および『*Cisco NX-OS* ライセンス オプション ガイド』を参照してください。

スタティック MPLS について

通常、ラベル スイッチング ルータ（LSR）は、パケットのラベル スイッチングに使用する必要があるラベルを、ラベル配布プロトコルを使用してダイナミックに学習します。そのようなプロトコルの例には、次のものがあります。

- ラベルをネットワークアドレスにバインドするために使用されるインターネットエンジニアリング タスク フォース（IETF）標準であるラベル配布プロトコル（LDP）
- トラフィック エンジニアリング（TE）のラベル配布に使用されるリソース予約プロトコル（RSVP）

- MPLS 仮想プライベートネットワーク（VPN）のラベル配布に使用される境界ゲートウェイ プロトコル（BGP）

学習したラベルをパケットのラベル スイッチングに使用するために、LSR はそのラベルをラベル転送情報ベース（LFIB）にインストールします。

静的 MPLS 機能を使用すると、以下を静的に設定できます。

- ラベルと IPv4 または IPv6 プレフィックス間のバインディング
- ラベルと IPv4 または IPv6 プレフィックスとの間のバインディングに対応するアクション（ラベル スワップまたはポップ）
- LFIB 相互接続エントリの内容

ラベルの入れ替えとポップ

ラベル付きパケットが MPLS ドメインを通過すると、ラベル スタックの最も外側のラベルが各ホップで検査されます。ラベルの内容により、スワップまたはポップ（ディスポーズ）のいずれかの操作がラベル スタックに対して実行されます。転送の決定は、パケット ヘッダー内のラベルの MPLS テーブル検索によって行われます。ネットワークを介したパケットの送信中にパケット ヘッダーを再評価する必要はありません。ラベルは構造化されていない固定長の値であるため、MPLS 転送テーブル検索プロセスは簡単かつ高速です。

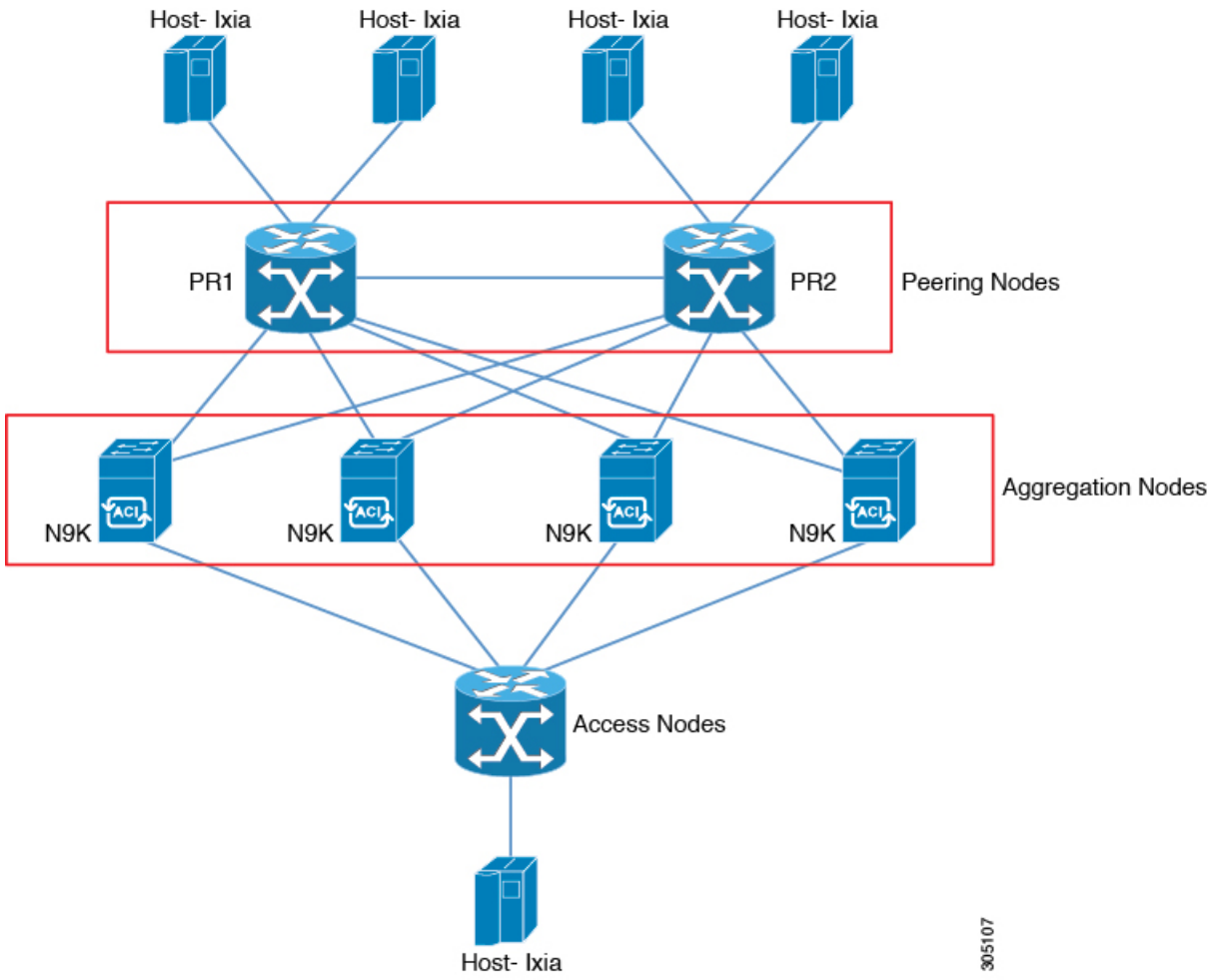
スワップ操作では、ラベルが新しいラベルと交換され、パケットは着信ラベルによって決定される次のホップに転送されます。

ポップ操作では、ラベルがパケットから削除され、下に内部ラベルが表示される場合があります。ポップされたラベルがラベル スタックの最後のラベルである場合、パケットは MPLS ドメインの外部へ転送されます。通常、このプロセスは出力 LSR で行われます。アグリゲータのプライマリ リンクに障害が発生すると、MPLS トラフィックがバックアップ リンクに再ルーティングされ、スワップ操作が発生します。

スタティック MPLS トポロジ

この図は、スタティック MPLS ソース ルーティング トポロジを示しています。アクセス ノードはスワップ操作を実行し、集約ノードはプライマリ パスのポップ操作とバックアップ パスのスワップ操作を実行します。

図 *1*: スタティック *MPLS* トポロジ



スタティック MPLS の利点

- ラベルと IPv4 または IPv6 プレフィックス間のスタティック バインディングは、LDP ラベル配布を実装しないネイバー ルータを通る MPLS ホップバイホップ転送をサポートするよう設定できます。
- スタティック相互接続は、ネイバー ルータが LDP または RSVP ラベル配布のいずれも実装していないものの、MPLS転送パスを実装している場合に、MPLS ラベル スイッチド パス（LSP）ミッドポイントをサポートするよう設定できます。

スタティック MPLS のためのハイ アベイラビリティ

Cisco Nexus 9500 シリーズ スイッチは、スタティック MPLS のステートフル スイッチオーバー（SSO）をサポートします。SSO の後、スタティック MPLS は以前の状態に戻ります。

スタティック MPLS は、SSO 中のゼロ トラフィック損失をサポートします。MPLS のスタティック再起動はサポートされていません。

（注） Cisco Nexus 9300 シリーズ スイッチは、SSO をサポートしていません。

スタティック MPLS の前提条件

スタティック MPLS には、次の前提条件があります。

- Cisco Nexus 9300 および 9500 シリーズ スイッチ、および Cisco Nexus 3164Q、31128PQ、3232C、および3264Q スイッチの場合、MPLS の ACL TCAM リージョンサイズを設定し、設定を保存して、スイッチをリロードする必要があります。（詳細については、Cisco Nexus 9000 Series NX-OS Security Configuration Guide の「Using Templates to Configure ACL TCAM Region Sizes」および「Configuring ACL TCAM Region Sizes」のセクションを参照してください）。Cisco Nexus 9200 シリーズ スイッチでは、静的 MPLS の TCAM カービングは必要ありません。

（注） デフォルトでは、mpls の領域サイズはゼロです。静的 MPLS をサポートするには、この領域を 256 に設定する必要があります。

スタティック MPLS の注意事項および制限事項

スタティック MPLS に関する注意事項と制限事項は次のとおりです。

- スタティック MPLS は、9400、9500、9600、および 9700-EX ラインカードを備えた Cisco Nexus 3100、3200、9200、9300、9300-EX、FX、FX2、および 9500 スイッチでサポートされています。
- Cisco NX-OS リリース 9.3(3) 以降、スタティック MPLS は Cisco Nexus 9364C-GX、Cisco Nexus 9316D-GX、および Cisco Nexus 93600CD-GX スイッチでサポートされています。
- 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。
- 等コスト マルチパス ルーティング（ECMP）は、ラベル ポップでサポートされていません。
- ラベルのポップ操作とスワップ操作はサポートされていますが、ラベルのプッシュ操作はサポートされていません。
- MPLS パケットは、入力ラベルが設定されたラベルとマッチし、設定された FEC（プレフィックス）がルーティング テーブルにある限り、転送されます。

- このデバイスは、通常、ラベル スイッチング ルータ（LSR）として機能します。パケットが隣接するラベル エッジ ルーター（LER）に渡される前に、LSR によってラベル FIB (LFIB) の出力ラベルとして明示的なヌル ラベルをインストールすると、デバイスは最後から 2 番目のホップ ポップの LER として動作します。つまり、ラベル スイッチング ルータ（LSR）は 1 つ以上のラベルで機能します。

（注） LSR で暗黙的ヌル CLI を意図的に使用する場合、LER に送信される出力パケットには、明示的ヌルと内部ラベルが含まれます。

- スタティック MPLS は、最大 128 のラベルをサポートします。
- バックアップ パスは、単一の隣接でのみサポートされ、ECMP ではサポートされません。
- Cisco Nexus 9300 シリーズ スイッチはバックアップ パス高速再ルート（FRR）サブセカンド コンバージェンスをサポートしますが、Cisco Nexus 9500 シリーズ スイッチは限定的なバックアップ パス FRR コンバージェンスをサポートします。
- ほとんどの MPLS コマンドの出力は、XML または JSON で生成できます。例については、静的 MPLS 設定の確認 （15 ページ） を参照してください。
- VRF、vPC、FEX、および VXLAN は、スタティック MPLS ではサポートされていません。
- サブインターフェイスを使用してリモート vpnv4 ネイバーに接続する場合、親インターフェイスで「mpls ip forwarding」コマンドを有効にする必要があります。
- コマンド「mpls ip forwarding」は、サブインターフェイスでは設定できません。
- サブインターフェイスは、スタティック MPLS ではサポートされていません。
- 転送等価クラス（FEC）は、ルーティング テーブル内のルートとマッチしている必要があります。
- X9536PQ、X9564PX、および X9564TX ライン カードと M12PQ 汎用拡張モジュール（GEM）では、スタティック MPLS が有効になっており、無効にすることはできません。
- 高速再ルート（バックアップ）を構成する場合、バックアップ構成のネクスト ホップ プレフィックスとして、接続されているネクスト ホップ（再帰ネクスト ホップではない）のみを指定できます。
- 複数の FEC がバックアップ（同じネクストホップとインターフェイス）を共有している場合、バックアップ構成を変更するには、バックアップ構成を共有している他のすべての FEC を再構成する必要があります。
- バックアップ パスがアクティブな場合、**show mpls switching labels** コマンドは、出力ラベル/出力インターフェイス/ネクスト ホップおよび関連する統計情報を表示しません。統計情報は、**show forwarding mpls label***label***stats platform**コマンドを使用して確認できます。
- トラフィックがデフォルト以外のユニット（デフォルトのユニットは unit0）で入出力される場合、対応する ULIB 統計情報は、**show mpls switching labels** *low-label-value*

静的 MPLS の設定

[*high-label-value*] **detail** コマンドの出力に表示されません。統計情報は、**show forwarding mpls label***label***stats platform**コマンドを使用して確認できます。

- バックアップ パスとプライマリ パスが同じインターフェイスを指している場合、バックアップ アクションのスワップが優先されます。
- 物理（イーサネット）およびポートチャネルは、バックアップの場合にのみサポートされます。
- 次のガイドラインと制約事項は、Cisco Nexus 9200 シリーズ スイッチに適用されます。
  - ECMP ハッシュは、内部フィールドでのみサポートされます。
  - MTU チェックは、MPLS ヘッダーを持つパケットではサポートされていません。

静的 MPLS の設定

スタティック MPLS の有効化

MPLS スタティック ラベルを設定するには、MPLS 機能セットをインストールして有効にしてから MPLS のスタティック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)#`
		グローバル設定モードを開始します。

ステップ 2	[**no**] **install feature-set mpls** 例： `switch(config)# install feature-set mpls`	MPLS機能セットを有効化します。このコマンドの **no** 形式は、MPLS 機能セットをアンインストールします。
ステップ 3	[**no**] **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。このコマンドの **no** 形式は、MPLS 機能セットを無効化します。
ステップ 4	[**no**] **feature mpls static** 例： `switch(config)# feature mpls static`	MPLS機能セットを有効にします。このコマンドの **no** 形式は、MPLS 機能セットを無効化します。

	コマンドまたはアクション	目的

ステップ 5	（任意） **show feature-set**
	例：
	`switch(config)# show feature-set`
	`Feature Set Name ID State`
	`-------------------- -------- --------`
	`mpls 4 enabled`
		MPLS機能セットのステータスを表示します。

ステップ 6	（任意） **show feature | inc mpls_static**
	例：
	`switch(config)# show feature | inc mpls_static`
	`mpls_static 1 enabled`
		スタティック MPLSのステータスを表示します。

スタティックな割り当てのために予約されたラベル

ダイナミックに割り当てられないようにスタティックに割り当てるラベルを予約します。

始める前に

スタティック MPLS 機能が有効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	[**no**] **mpls label range** *min-value max-value* [**static** *min-static-value max-static-value*] 例： `switch(config)# mpls label range 17 99 static 100 10000`	スタティック ラベル割り当てに使用する一連のラベルを予約します。 最小値と最大値の範囲は 16 ～ 471804 です。
ステップ 3	（任意） **show mpls label range** 例： `switch(config)# show mpls label range`	スタティック MPLS に設定されているラベル範囲を表示します。
ステップ 4	（任意） **copy running-config startup-config** 例：	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。

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

スワップ操作とポップ操作を使用したスタティックラベルとプレフィックス バインディングの設定

トップオブラック構成では、外側のラベルが指定された新しいラベルとスワップされます。パケットはネクストホップ アドレスに転送され、新しいラベルによって自動解決されます。

アグリゲータ構成では、外部ラベルがポップされ、残りのラベルを持つパケットがネクストホップ アドレスに転送されます。ポップ操作はプライマリ パスで実行され、スワップ操作はバックアップ パスで実行されます。

始める前に

静的 MPLS 機能が有効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	**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	**mpls static configuration**
	例：
	`switch(config-if)# mpls static configuration`
	`switch(config-mpls-static)#`
		MPLS 静的グローバル 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	**address-family** {**ipv4** | **ipv6**} **unicast** 例：	指定された IPv4 または iPv6 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
	`switch(config-mpls-static)# address-family ipv4 unicast switch(config-mpls-static-af)#`	
ステップ 6	**local-label** *local-label-value* **prefix** *destination-prefix destination-prefix-mask* 例： `switch(config-mpls-static-af)# local-label 2000 prefix 1.255.200.0 255.255.255.25 switch(config-mpls-static-af-lbl)#`	IPv4 または IPv6 プレフィックスに対する入力ラベルの静的バインディングを指定します。*local-label-value* は、**mpls label range** コマンドで定義された静的 MPLS ラベルの範囲です。
ステップ 7	**next-hop** {**auto-resolve** | *destination-ip-next-hop* **out-label implicit-null** | **backup** *local-egress-interface destination-ip-next-hop* **out-label** *output-label-value*} 例： `switch(config-mpls-static-af-lbl)# next-hop auto-resolve`	ネクスト ホップを指定します。次のオプションを使用できます。 • **next-hop auto-resolve**：このオプションは、ラベル スワップ操作に使用します。 • **next-hop** *destination-ip-next-hop* **out-label implicit-null**：ラベル ポップ操作のプライマリ パスにはこのオプションを使用します。 • **next-hop backup** *local-egress-interface destination-ip-next-hop* **out-label** *output-label-value*：ラベル ポップ操作のバックアップ パスにはこのオプションを使用します。
ステップ 8	（任意） **copy running-config startup-config** 例： `switch(config-mpls-static-af-lbl)# copy running-config startup-config`	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。

セグメント ルーティング隣接関係統計の設定

デフォルトでは、統計情報収集モードは、特定の隣接関係から出力されるパケット数を累積します。Cisco NX-OS リリース 9.3(1) 以降では、隣接関係のバイト数を累積するように統計情報収集モードを設定できます。

このモードは、MPLSセグメントルーティング機能を有効にすると使用できますが、バイトを累積するように収集モード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)#`
		グローバル設定モードを開始します。

ステップ 2	[**no**] **install feature-set mpls** 例： `switch(config)# install feature-set mpls`	MPLS機能セットを有効化します。このコマンドの **no** 形式は、MPLS 機能セットをアンインストールします。
ステップ 3	[**no**] **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。このコマンドの **no** 形式は、MPLS 機能セットを無効化します。
ステップ 4	[**no**] **feature mpls segment-routing** 例： `switch(config)# feature mpls segment-routing`	MPLS セグメント ルーティング機能を有効化します。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 5	[**no**] **hardware profile mpls adjacency-stats bytes** 例： `switch(config)# hardware profile mpls adjacency-stats bytes`	特定の隣接関係のバイト数を累積するように、出力統計の統計収集モードを設定します。このコマンドの **no** 形式を使用すると、収集モードがリセットされ、パケット数が累積されます。

ステップ 6	（任意） **show running-config | grep adjacency stats**
	例：
	`witch(config)# show running-config | grep adjacency-stats`
	`hardware profile mpls adjacency-stats bytes`
	`switch(config)#`
		ノブの設定を表示します。

ステップ 7	（任意） **show feature-set**
	例：
	`switch(config)# show feature-set`
	`Feature Set Name ID State`
	`-------------------- --------`
	`--------`
	`mpls 4 enabled`
		MPLS機能セットのステータスを表示します。

	コマンドまたはアクション	目的

ステップ 8	（任意） **show feature | grep segment-routing**
	例：
	`switch(config)# show feature | grep segment-routing`
	`segment-routing 1`
	`enabled`
		MPLSセグメントルーティングのステータスを表示します。

ステップ 9	**show forwarding mpls** [**label** *label*] **stats**
	例：
	`switch(config)# show forwarding mpls label 22 stats`
	
	`slot 1`
	`=======`
	
	`------+--------+-----------+----------+--------+-----`
	`Local |Prefix |FEC |Next-Hop |Interface |Out`
	`Label |Table Id |(Prefix/Tunnel id) | | |Label`
	`------+--------+-----------+----------+--------+-----`
	`22 |0x1 |182.1.1.7/32 |30.1.8.1 |Po11 |0 SWAP`
	
	`Input Pkts : 488482 Input Bytes : 250102784`
	`SWAP Output Pkts: 0 SWAP Output Bytes: 84215808`
	`TUNNEL Output Pkts: 0 TUNNEL Output Bytes: 0`
	`switch(config)#`
		隣接関係の統計情報を表示します。

静的 MPLS 設定の確認

静的 MPLS 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 feature | inc mpls_static	スタティック MPLSのステータスを表示します。
show feature-set	MPLS機能セットのステータスを表示します。
show ip route	Unicast Route Information Base（RIB）からルートを表示します。
show mpls label range	スタティック MPLS に設定されているラベル範囲を表示します。

コマンド	目的
show mpls static binding {**all** | **ipv4** | **ipv6**}	設定された静的プレフィックスまたはラベルバインディングを表示します。
show mpls switching [**detail**]	MPLS スイッチング情報を表示します。
show mpls switching label [**detail**]	MPLS スイッチング ラベル情報を表示します。
show forwarding mpls [**label** *label*] **stats**	有効になっているラベルに基づいて隣接統計を表示します。
show forwarding adjacency mpls stats	隣接関係の統計情報を表示します。

次の例は、**show mpls static binding all** コマンドの出力例を示しています。

```
1.255.200.0/32: (vrf: default) Incoming label: 2000
  Outgoing labels:
      1.21.1.1 implicit-null
      backup 1.24.1.1 2001

2000:1:255:201::1/128: (vrf: default) Incoming label: 3000
  Outgoing labels:
      2000:1111:2121:1111:1111:1111:1111:1 implicit-null
      backup 2000:1:24:1::1 3001
```

次に、**show mpls switching detail** コマンドの出力例を示します。

```
VRF default

IPv4 FEC
 In-Label                       : 2000
 Out-Label stack                : Pop Label
 FEC                            : 1.255.200.0/32
 Out interface                  : Po21
 Next hop                       : 1.21.1.1
 Input traffic statistics       : 0 packets, 0 bytes
 Output statistics per label    : 0 packets, 0 bytes
IPv6 FEC
 In-Label                       : 3000
 Out-Label stack                : Pop Label
 FEC                            : 2000:1:255:201::1/128
 Out interface                  : port-channel21
 Next hop                       : 2000:1111:2121:1111:1111:1111:1111:1
 Input traffic statistics       : 0 packets, 0 bytes
 Output statistics per label    : 0 packets, 0 bytes
```

この例は、スイッチが静的 IPv4 プレフィックスで構成されている場合の **show mpls switching** コマンドの通常、XML、および JSON のサンプル出力を示しています。

```
switch# show run mpls static | sec 'ipv4 unicast'
address-family ipv4 unicast
local-label 100 prefix 192.168.0.1 255.255.255.255 next-hop auto-resolve out-label 200

switch# show mpls switching
Legend:
(P)=Protected, (F)=FRR active, (*)=more labels in stack.
IPV4:
In-Label   Out-Label  FEC name           Out-Interface      Next-Hop
```

```
VRF default
100        200        192.168.0.1/32     Eth1/23            1.12.23.2
```

```
switch# show mpls switching | xml
<?xml version="1.0" encoding="ISO-8859-1"?> <nf:rpc-reply
xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns="http://w
ww.cisco.com/nxos:1.0:ulib">
 <nf:data>
  <show>
   <mpls>
    <switching>
     <__XML__OPT_Cmd_ulib_show_switching_cmd_labels>
      <__XML__OPT_Cmd_ulib_show_switching_cmd_detail>
       <__XML__OPT_Cmd_ulib_show_switching_cmd_vrf>
        <__XML__OPT_Cmd_ulib_show_switching_cmd___readonly__>
         <__readonly__>
          <TABLE_vrf>
           <ROW_vrf>
            <vrf_name>default</vrf_name>
            <TABLE_inlabel>
             <ROW_inlabel>
              <in_label>100</in_label>
              <out_label_stack>200</out_label_stack>
              <ipv4_prefix>192.168.0.1/32</ipv4_prefix>
              <out_interface>Eth1/23</out_interface>
              <ipv4_next_hop>1.12.23.2</ipv4_next_hop>
              <nhlfe_p2p_flag> </nhlfe_p2p_flag>
             </ROW_inlabel>
            </TABLE_inlabel>
           </ROW_vrf>
          </TABLE_vrf>
         </__readonly__>
        </__XML__OPT_Cmd_ulib_show_switching_cmd___readonly__>
       </__XML__OPT_Cmd_ulib_show_switching_cmd_vrf>
      </__XML__OPT_Cmd_ulib_show_switching_cmd_detail>
     </__XML__OPT_Cmd_ulib_show_switching_cmd_labels>
    </switching>
   </mpls>
  </show>
 </nf:data>
</nf:rpc-reply>
]]>]]>
```

```
switch# show mpls switching | json
{"TABLE_vrf": {"ROW_vrf": {"vrf_name": "default", "TABLE_inlabel":
{"ROW_inlabel
": {"in_label": "100", "out_label_stack": "200", "ipv4_prefix":
"192.168.0.1/32"
, "out_interface": "Eth1/23", "ipv4_next_hop": "1.12.23.2",
"nhlfe_p2p_flag": nu
ll}}}}}
```

スタティック MPLS 統計の表示

スタティック MPLS 統計を監視するには、次のいずれかのタスクを実行します。

コマンド	目的
show forwarding [**ipv6**] **adjacency mpls stats**	MPLS IPv4 または IPv6 隣接関係統計を表示します。
show forwarding mpls drop-stats	MPLS 転送パケット ドロップの統計情報を表示します。
show forwarding mpls ecmp [**module** *slot* | **platform**]	等コスト マルチパス（ECMP）の MPLS 転送統計を表示します。
show forwarding mpls label *label* **stats** [**platform**]	MPLS ラベル転送の統計情報を表示します。
show mpls forwarding statistics [**interface** *type slot/port*]	MPLS 転送の統計情報を表示します。
show mpls switching labels *low-label-value* [*high-label-value*] [**detail**]	MPLS ラベル スイッチングの統計情報を表示します。ラベル値の範囲は 0 ～ 524286 です。

次に、**show forwarding adjacency mpls stats** コマンドの出力例を示します。

```
FEC             next-hop   interface  tx packets  tx bytes  Label info
--------------- ---------- ---------- ----------- --------- ----------
1.255.200.0/32  1.21.1.1   Po21       87388       10836236  POP 3
1.255.200.0/32  1.24.1.1   Po24       0           0         SWAP 2001
switch(config)#
switch(config)# show forwarding mpls drop-stats

Dropped packets : 73454
Dropped bytes : 9399304
```

次に、**show forwarding ipv6 adjacency mpls stats** コマンドの出力例を示します。

```
FEC                    next-hop        interface  tx packets  tx bytes  Label info
---------------------- --------------  ---------- ----------- --------- -----------
2000:1:255:201::1/128  2000:1.21.1.1   Po21       46604       5778896   POP 3
2000:1:255:201::1/128  2000:1:24:1::1  Po24       0           0         SWAP 3001
```

次に、**show forwarding mpls label 2000 stats** コマンドの出力例を示します。

```
--------+-----------+------------------+----------------+-------------+-------
Local   |Prefix     |FEC               |Next-Hop        |Interface    |Out
Label   |Table Id   |(Prefix/Tunnel id) |                |             |Label
--------+-----------+------------------+----------------+-------------+-------
2000    |0x1        |1.255.200.0/32     |1.21.1.1        |Po21         |Pop Label
 HH: 100008, Refcount: 1
Input Pkts : 77129               Input Bytes : 9872512
Output Pkts: 77223               Output Bytes: 9575652
```

次に、**show mpls forwarding statistics** コマンドの出力例を示します。

```
MPLS software forwarding stats summary:
    Packets/Bytes sent           : 0/0
    Packets/Bytes received       : 0/0
    Packets/Bytes forwarded      : 0/0
```

```
Packets/Bytes originated     : 0/0
Packets/Bytes consumed       : 0/0
Packets/Bytes input dropped  : 0/0
Packets/Bytes output dropped : 0/0
```

スタティック MPLS 統計情報のクリア

MPLS 統計情報をクリアするには、次の作業を行います。

コマンド	目的
clear forwarding [**ipv6**] **adjacency mpls stats**	MPLS IPv4 または IPv6 隣接関係統計を消去します。
clear forwarding mpls drop-stats	MPLS 転送パケット ドロップ統計情報をクリアします。
clear forwarding mpls stats	入力 MPLS 転送統計情報をクリアします。
clear mpls forwarding statistics	MPLS 転送統計情報をクリアします。
clear mpls switching label statistics [**interface** *type slot/port*]	MPLS スイッチング ラベルの統計情報をクリアします。

スタティック MPLS の設定例

次に、スタティック割り当てに使用するラベルを予約する例を示します。

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# mpls label range 17 99 static 100 10000
switch(config)# show mpls label range
Downstream Generic label region: Min/Max label: 17/99
Range for static labels: Min/Max Number: 100/10000
```

次の例は、トップオブラック構成（スワップ構成）で MPLS スタティック ラベルと IPv4 プレフィックス バインディングを構成する方法を示しています。

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# interface ethernet 1/1
switch(config-if)# mpls ip forwarding
switch(config-if)# mpls static configuration
switch(config-mpls-static)# address-family ipv4 unicast
switch(config-mpls-static-af)# local-label 2000 prefix 1.255.200.0/32
switch(config-mpls-static-af-lbl)# next-hop auto-resolve out-label 2000
```

次の例は、トップオブラック構成（スワップ構成）で MPLS スタティック ラベルと IPv6 プレフィックス バインディングを構成する方法を示しています。

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# interface ethernet 1/1
switch(config-if)# mpls ip forwarding
switch(config-if)# mpls static configuration
switch(config-mpls-static)# address-family ipv6 unicast
switch(config-mpls-static-af)# local-label 3001 prefix 2000:1:255:201::1/128
switch(config-mpls-static-af-lbl)# next-hop auto-resolve out-label 3001
```

次の例は、アグリゲータ構成（ポップ構成）で MPLS スタティック ラベルと IPv4 プレフィックス バインディングを構成する方法を示しています。

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# interface ethernet 1/1
switch(config-if)# mpls ip forwarding
switch(config-if)# mpls static configuration
switch(config-mpls-static)# address-family ipv4 unicast
switch(config-mpls-static-af)# local-label 2000 prefix 1.255.200.0/32
switch(config-mpls-static-af-lbl)# next-hop 1.31.1.1 out-label implicit-null
switch(config-mpls-static-af-lbl)# next-hop backup Po34 1.34.1.1 out-label 2000
```

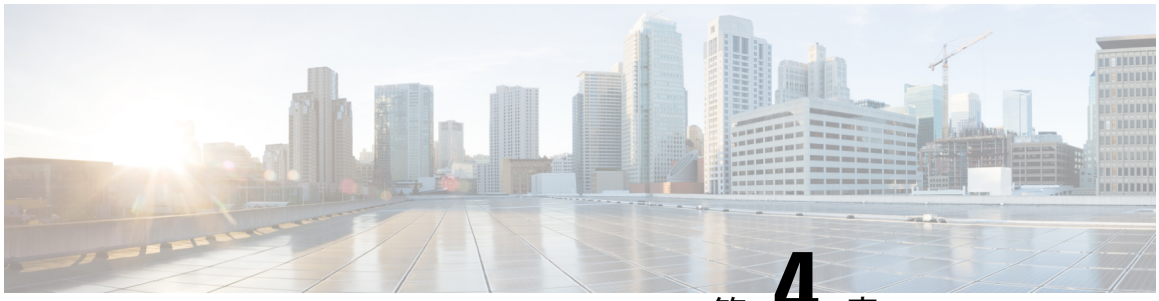
次の例は、アグリゲータ構成（ポップ構成）で MPLS スタティック ラベルと IPv6 プレフィックス バインディングを構成する方法を示しています。

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# interface ethernet 1/1
switch(config-if)# mpls ip forwarding
switch(config-if)# mpls static configuration
switch(config-mpls-static)# address-family ipv6 unicast
switch(config-mpls-static-af)# local-label 3001 prefix 2000:1:255:201::1/128
switch(config-mpls-static-af-lbl)# next-hop 2000:1:31:1::1 out-label implicit-null
switch(config-mpls-static-af-lbl)# next-hop backup Po34 2000:1:34:1::1 out-label 3001
```

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
TCAM リージョン	詳細については、*ACL TCAM* リージョン サイズの設定のセクション（Cisco Nexus 9000 シリーズ セキュリティ設定ガイド）を参照してください。



第 4 章

MPLS ラベル インポジションの設定

この章では、マルチプロトコル ラベル スイッチング（MPLS）ラベル インポジションの設定方法について説明します。

MPLS ラベル インポジションについて

MPLS ラベル スタック インポジション機能を使用して、1 つ以上のラベルを持つ発信ラベル スタックを静的にプロビジョニングできます。発信ラベル スタックは、次の 2 種類の静的に設定された MPLS バインディングで使用されます。

- ラベル スタックへのプレフィックスとラベル：ここでは、静的 MPLS と同様に、IP プレフィックスまたは着信ラベルが発信スタックにマッピングされます。着信プレフィックスは、IP のみの入力トラフィックの out-label-stack にマッピングされます。
- ラベル スタックへのラベル：ここでは、受信ラベルのみがプレフィックスなしで送信スタックにマップされます。

新しい MPLS バインディング タイプは静的 MPLS コンポーネントに実装され、**feature mpls segment-routing** コマンドが有効になっている場合にのみ使用できます。

MPLS ラベル インポジションの設定されたネクスト ホップが SR 再帰ネクスト ホップ（RNH）である場合、それらは RIB を使用して実際のネクスト ホップに解決されます。out-label スタックの外部ラベルは、SR によって割り当てられたラベルから自動的にインポジションされます。

ECMP は、いくつかのパス構成を追加することによってもサポートされます。

（注） 静的 MPLS プロセスは、**feature mpls segment-routing** コマンドまたは **feature mpls static** コマンドのいずれかが実行されたときに開始されます。**feature mpls segment-routing** コマンドを使用してスタティック MPLS を実行すると、一部の標準スタティック MPLS コマンドを使用できなくなり、**feature mpls static** コマンドを実行すると、MPLS バインディングのコマンドを使用できなくなります。

MPLS ラベル インポジションに関する注意事項と制限事項

MPLS ラベル インポジションに関する注意事項と制約事項は次のとおりです。

- MPLS ラベル インポジションは、以下のスイッチでサポートされています。
  - 9400、9500、9600、9700-EX、および 9700-FX ラインカードを搭載した Cisco Nexus 9200、9300、9300-EX、9300-FX、および 9500 プラットフォーム スイッチ。
  - Cisco Nexus 3164Q、31128PQ、3232C、および 3264Q スイッチ。
  - Cisco NX-OS リリース 9.2(1) リリース以降、Cisco Nexus 9364C スイッチでサポートされています。
  - Cisco NX-OS リリース 9.3(3) 以降、Cisco Nexus 9364C-GX、Cisco Nexus 9316D-GX、および Cisco Nexus 93600CD-GX スイッチでサポートされています。
- MPLS ラベル インポジションは、IPv4 のみをサポートします。
- アウトラベル スタックのラベルの最大数は、Cisco Nexus 9200、9300-EX、および 9300-FX プラットフォーム スイッチの場合は 5、Cisco Nexus 9300 と 9500 プラットフォーム スイッチおよび Cisco Nexus 3164Q、31128PQ、3232C、および 3264Q スイッチの場合は 3 です。。これより多くのラベルをインポーズしようとすると、後続のラベルが自動的に切り捨てられ、syslog エラー メッセージが表示され、構成を修正するように通知されます。
- マルチキャストは、MPLS ラベル インポジションではサポートされていません。
- マルチラベル スタック構成では、発信パスの変更は Cisco Nexus 9200 および 9300-EX シリーズ スイッチでのみ許可されます。
- サブインターフェイスとポート チャネルは、MPLS ラベル インポジションではサポートされていません。
- ルーティング プロトコル (スタティック ルートを含む) から学習したプレフィックスおよび関連するサブネット マスクは、ラベル スタック インポジション ポリシーの一部として使用できません。
- ラベル スタック インポジションの検証済みスケーラビリティ制限については、お使いのデバイスの『検証済みスケーラビリティ ガイド』を参照してください。

MPLS ラベル インポジションの設定

MPLS ラベル インポジションの有効化

MPLS ラベル インポジションを設定するには、MPLS 機能セットをインストールして有効にしてから、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)#`
		グローバル設定モードを開始します。

ステップ 2	[**no**] **install feature-set mpls** 例： `switch(config)# install feature-set mpls`	MPLS機能セットを有効化します。このコマンドの **no** 形式は、MPLS 機能セットをアンインストールします。
ステップ 3	[**no**] **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。このコマンドの **no** 形式は、MPLS 機能セットを無効化します。
ステップ 4	[**no**] **feature mpls segment-routing** 例： `switch(config)# feature mpls segment-routing`	MPLS セグメント ルーティング機能を有効化します。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 5	（任意） **show feature-set**
	例：
	`switch(config)# show feature-set`
	`Feature Set Name ID State`
	`-------------------- -------- --------`
	`mpls 4 enabled`
		MPLS機能セットのステータスを表示します。

ステップ 6	（任意） **show feature | grep segment-routing**
	例：
	`switch(config)# show feature | grep segment-routing`
	`segment-routing 1 enabled`
		MPLSセグメントルーティングのステータスを表示します。

	コマンドまたはアクション	目的

MPLS ラベル インポジション用のラベルの予約

スタティックに割り当てるラベルを予約します。動的なラベル割り当てはサポートされていません。

始める前に

MPLS セグメント ルーティング機能が有効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	[**no**] **mpls label range** *min-value max-value* [**static** *min-static-value max-static-value*] 例： `switch(config)# mpls label range 17 99 static 100 10000`	スタティック ラベル割り当てに使用する一連のラベルを予約します。 最小値と最大値の範囲は 16 ～ 471804 です。
ステップ 3	（任意） **show mpls label range** 例： `switch(config)# show mpls label range`	スタティック MPLS に設定されているラベル範囲を表示します。
ステップ 4	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	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 ラベル インポジションの設定

デバイスに MPLS ラベル インポジションを設定できます。

（注） **feature mpls segment-routing** コマンドは、**feature nv overlay**、**nv overlay evpn**、**feature vpc**、および **feature vn-segment-vlan-based** コマンドが使用されている場合、有効にすることはできません。

始める前に

MPLS セグメント ルーティング機能が有効になっていることを確認します。

静的ラベル範囲を次のように設定します。**mpls label range 16 16 static 17 50000**

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	**mpls static configuration**
	例：
	`switch(config-if)# mpls static configuration`
	`switch(config-mpls-static)#`
		MPLS 静的グローバル 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	**address-family ipv4 unicast**
	例：
	`switch(config-mpls-static)# address-family ipv4 unicast`
	`switch(config-mpls-static-af)#`
		指定された 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 6	**lsp** *name*
	例：
	`switch(config-mpls-static-af)# lsp lsp1`
	`switch(config-mpls-static-lsp)#`
		LSP の名前を指定します。

ステップ 7	**in-label** *value* **allocate policy** *prefix*
	例：
	`switch(config-mpls-static-lsp)# in-label 8100 allocate policy 15.15.1.0/24`
	`switch(config-mpls-static-lsp-inlabel)#`
		in-label 値とプレフィックス値を設定します（オプション）。

	コマンドまたはアクション	目的

ステップ 8	**forward**
	例：
	`switch(config-mpls-static-lsp-inlabel)# forward`
	`switch(config-mpls-static-lsp-inlabel-forw)#`
		転送モードに入ります。

ステップ 9	**path** *number* **next-hop** *ip-address* **out-label-stack** *label-id label-id* 例： `switch(config-mpls-static-lsp-inlabel-forw)# path 1 next-hop 13.13.13.13 out-label-stack 16 3000`	パスを指定します。サポートされるパスの最大数は 32 です。
ステップ 10	（任意） **copy running-config startup-config** 例： `switch(config-mpls-static-lsp-inlabel-forw)# copy running-config startup-config`	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 ラベル インポジション設定の確認

MPLS ラベル インポジション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 feature | grep segment-routing	MPLS ラベル インポジションのステータスを表示します。
show feature-set	MPLS機能セットのステータスを表示します。
show forwarding mpls label *label*	特定のラベルの MPLS ラベル転送統計情報を表示します。
show mpls label range	MPLS ラベル インポジションに設定されているラベル範囲を表示します。
show mpls static binding {**all** | **ipv4**}	設定された静的プレフィックスまたはラベル バインディングを表示します。
show mpls switching [**detail**]	MPLS ラベル スイッチングの情報を表示します。
show running-config mpls static	実行中の静的 MPLS 設定を表示します。

次に、**show forwarding mpls label 8100** コマンドの出力例を示します。

```
slot 1
=======
--------+-----------+------------------+---------------+-------------+----------------------
Local|Prefix|FEC          |Next-Hop  |Interface | Out Label |Table Id |(Prefix/Tunnel
id)|Label
--------+-----------+------------------+---------------+-------------+----------------------
8100 |0x1   |25.25.0.0/16 |12.12.1.2 |Po121     |3131 SWAP  |         |
  | 17
"    |0x1   |25.25.0.0/16 |12.12.2.2 |Eth1/51   |3131 SWAP  |         |
  | 17
"    |0x1   |25.25.0.0/16 |12.12.3.2 |Vlan122   |3131 SWAP  |         |
  | 17
"    |0x1   |25.25.0.0/16 |12.12.4.2 |Vlan123   |3131 SWAP  |         |
  | 17
```

次に、**show mpls static binding all** コマンドの出力例を示します。

```
LI_TEST1 25.25.0.0/16: (vrf: default) Incoming label: 8100
LSP Type: POLICY
  Outgoing labels:
     (path 1)  12.12.1.2 3131,17
     (path 2)  12.12.2.2 3131,17
     (path 3)  12.12.3.2 3131,17
     (path 4)  12.12.4.2 3131,17

LI_TEST2 (vrf: default) Incoming label: 8200
LSP Type: XC
  Outgoing labels:
     (path 1)  12.12.3.2 3132,16
     (path 2)  12.12.4.2 3132,16
     (path 3)  12.12.1.2 3132,16
     (path 4)  12.12.2.2 3132,16
```

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

```
Legend:
(P)=Protected, (F)=FRR active, (*)=more labels in stack.

Local      Out-Label  FEC                                          Out-Interface
 Next-Hop
8200       3132       Label 8200
 12.12.3.2                                      *
8200       3132       Label 8200
 12.12.4.2                                      *
8200       3132       Label 8200
 12.12.1.2                                      *
8200       3132       Label 8200
 12.12.2.2                                      *

Local      Out-Label  FEC                                          Out-Interface
 Next-Hop
8100       3131       Pol 25.25.0.0/16
 12.12.1.2                                      *
8100       3131       Pol 25.25.0.0/16
 12.12.2.2                                      *
8100       3131       Pol 25.25.0.0/16
 12.12.3.2                                      *
8100       3131       Pol 25.25.0.0/16
 12.12.4.2                                      *
```

次に、**show running-config mpls static** コマンドの出力例を示します。

```
mpls static configuration
  address-family ipv4 unicast
    lsp LI_TEST2
      in-label 8100 allocate policy 25.25.0.0 255.255.0.0
        forward
          path 1 next-hop 12.12.1.2 out-label-stack 3131 17
          path 2 next-hop 12.12.2.2 out-label-stack 3131 17
          path 3 next-hop 12.12.3.2 out-label-stack 3131 17
          path 4 next-hop 12.12.4.2 out-label-stack 3131 17
```

次に、**show running-config mpls static all** コマンドの出力例を示します。

```
switch# show running-config mpls static all

!Command: show running-config mpls static all
!Time: Mon Aug 21 14:59:46 2017

version 7.0(3)I7(1)
logging level mpls static 5
mpls static configuration
address-family ipv4 unicast
lsp 9_label_stack_LPM
in-label 72000 allocate policy 71.200.11.0 255.255.255.0
forward
path 1 next-hop 27.1.32.4 out-label-stack 21901 29701 27401 24501 25801
lsp 9_label_stack_LPM_01
in-label 72001 allocate policy 72.201.1.1 255.255.255.255
lsp DRV-01
in-label 71011 allocate policy 71.111.21.0 255.255.255.0
forward
path 1 next-hop 27.1.31.4 out-label-stack implicit-null
lsp DRV-02
in-label 71012 allocate policy 71.111.22.0 255.255.255.0
forward
path 1 next-hop 8.8.8.8 out-label-stack 28901
lsp DRV-03
switch# show forwarding mpls label 72000

slot 1
=======

--------+-----------+------------------+----------------+-------------+-------
Local |Prefix |FEC |Next-Hop |Interface |Out
Label |Table Id |(Prefix/Tunnel id) | | |Label
--------+-----------+------------------+----------------+-------------+-------
72000 |0x1 |71.200.11.0/24 |27.1.32.4 |Eth1/21 |21901 SWAP
| | | | | 29701
| | | | | 27401
| | | | | 24501
| | | | | 25801
```

MPLS ラベル インポジション統計の表示

MPLS ラベル インポジションの統計情報を監視するには、次のいずれかのタスクを実行します。

コマンド	目的
show forwarding [**ipv4**] **adjacency mpls stats**	MPLS IPv4 隣接関係統計を（パケットとバイトの両方で）表示します。 （注） Cisco Nexus 9200 および 9300-EX シリーズスイッチは、このコマンドをサポートしていません。
show forwarding mpls label *label* **stats** [**platform**]	MPLS ラベル転送の統計情報を表示します。
show mpls forwarding statistics [**interface** *type slot/port*]	MPLS 転送の統計情報を表示します。
show mpls switching labels *low-label-value* [*high-label-value*] [**detail**]	MPLS ラベル スイッチングの統計情報を表示します。ラベル値の範囲は 0 ～ 524286 です。

次に、**show forwarding adjacency mpls stats** コマンドの出力例を示します。

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

 FEC   next-hop    interface  tx packets   tx bytes  Label info
------ ---------- ---------- ------------ --------- --------------
       12.12.3.2   Vlan122      0            0         SWAP 3131 17
       12.12.3.2   Vlan122      0            0         SWAP 3132 16
       12.12.4.2   Vlan123      0            0         SWAP 3131 17
       12.12.4.2   Vlan123      0            0         SWAP 3132 16
       12.12.1.2   Po121        0            0         SWAP 3131 17
       12.12.1.2   Po121        0            0         SWAP 3132 16
       12.12.2.2   Eth1/51      0            0         SWAP 3131 17
       12.12.2.2   Eth1/51      0            0         SWAP 3132 16
```

次に、**show forwarding mpls label 8100 stats** コマンドの出力例を示します。

```
slot  1
=======
--------+-----------+-------------------+----------------+-------------+-------
Local   |Prefix     |FEC                |Next-Hop        |Interface    |Out
Label   |Table Id   |(Prefix/Tunnel id) |                |             |Label
--------+-----------+-------------------+----------------+-------------+-------
8100    |0x1        |25.25.0.0/16       |12.12.1.2       |Po121        |3131
  SWAP
        |           |                   |                |             | 17
    "   |0x1        |25.25.0.0/16       |12.12.2.2       |Eth1/51      |3131
  SWAP
        |           |                   |                |             | 17
    "   |0x1        |25.25.0.0/16       |12.12.3.2       |Vlan122      |3131
  SWAP
        |           |                   |                |             | 17
    "   |0x1        |25.25.0.0/16       |12.12.4.2       |Vlan123      |3131
  SWAP
        |           |                   |                |             | 17

Input Pkts : 126906012            Input Bytes : 64975876096
SWAP Output Pkts: 126959183       SWAP Output Bytes: 65764550340
TUNNEL Output Pkts: 126959053     TUNNEL Output Bytes: 66272319384
```

次に、**show mpls forwarding statistics** コマンドの出力例を示します。

```
MPLS software forwarding stats summary:
    Packets/Bytes sent           : 0/0
    Packets/Bytes received       : 0/0
    Packets/Bytes forwarded      : 0/0
    Packets/Bytes originated     : 0/0
    Packets/Bytes consumed       : 0/0
    Packets/Bytes input dropped  : 0/0
    Packets/Bytes output dropped : 0/0
```

MPLS ラベル インポジション統計のクリア

MPLS ラベル インポジションの統計情報をクリアするには、次の作業を行います。

コマンド	目的
clear forwarding [**ipv4**] **adjacency mpls stats**	MPLS IPv4 隣接関係の統計情報を消去します。
clear forwarding mpls stats	入力 MPLS 転送統計情報をクリアします。
clear mpls forwarding statistics	MPLS 転送統計情報をクリアします。
clear mpls switching label statistics [**interface** *type slot/port*]	MPLS スイッチング ラベルの統計情報をクリアします。

MPLS ラベル インポジションの設定例

次の例は、プレフィックスと incoming-label を out-label-stack バインディングに割り当てることにより、MPLS ラベル インポジションを設定する方法を示しています。

```
switch(config-if)# mpls static configuration
switch(config-mpls-static)# address-family ipv4 unicast
switch(config-mpls-static-af)# lsp LI_TEST1
switch(config-mpls-static-lsp)# in-label 8100 allocate policy 25.25.0.0/16
switch(config-mpls-static-lsp-inlabel)# forward
switch(config-mpls-static-lsp-inlabel-forw)# path 1 next-hop 12.12.1.2 out-label-stack
3131 17
switch(config-mpls-static-lsp-inlabel-forw)# path 2 next-hop 12.12.2.2 out-label-stack
3131 17
switch(config-mpls-static-lsp-inlabel-forw)# path 3 next-hop 12.12.3.2 out-label-stack
3131 17
switch(config-mpls-static-lsp-inlabel-forw)# path 4 next-hop 12.12.4.2 out-label-stack
3131 17
```

next-hop を削除するには、次を使用できます：

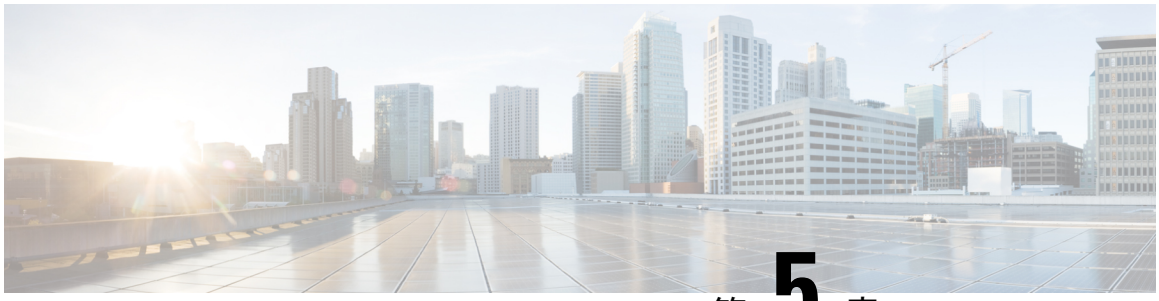
```
no path 1
```

指定された lsp を削除するには、次を使用できます：

```
no lsp LI_TEST1
```

次の例は、incoming-label を out-label-stack バインディングに割り当てることにより、MPLS ラベル インポジションを設定する方法を示しています（プレフィックスなし）。

```
switch(config-if)# mpls static configuration
switch(config-mpls-static)# address-family ipv4 unicast
switch(config-mpls-static-af)# lsp LI_TEST1
switch(config-mpls-static-lsp)# in-label 8200 allocate
switch(config-mpls-static-lsp-inlabel)# forward
switch(config-mpls-static-lsp-inlabel-forw)# path 1 next-hop 12.12.3.2 out-label-stack
3132 16
switch(config-mpls-static-lsp-inlabel-forw)# path 2 next-hop 12.12.4.2 out-label-stack
3132 16
switch(config-mpls-static-lsp-inlabel-forw)# path 3 next-hop 12.12.1.2 out-label-stack
3132 16
switch(config-mpls-static-lsp-inlabel-forw)# path 4 next-hop 12.12.2.2 out-label-stack
3132 16
```



第 5 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。

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。

DSCP のデフォルト動作

入力および出力に設定された DiffServ トンネリング モードによって、DSCP フィールド処理のデフォルト動作が決まります。DiffServ 仕様の MPLS ネットワーク サポートでは、次のトンネリング モードが定義されています。

- **均一モード**

  入力 トンネル エンドポイントによって、着信 IP パケットの DSCP ビットが、カプセル化中にインポーズされたラベルの MPLS EXP ビットにコピーされます。出力トンネル エンドポイントでのカプセル化解除中は、元の DSCP 値は保持されません。代わりに、外部ヘッダー MPLS EXP が内部 IP ヘッダーの DSCP にコピーされます。

- **パイプ モード**

  出力トンネル エンドポイントでのカプセル化解除時に、外部ヘッダーの MPLS EXP は廃棄されますが、内部 IP ヘッダーの DSCP は保持されます。

デフォルトでは、すべての Cisco Nexus プラットフォーム スイッチは、カプセル化に均一モードを使用します。

デフォルトでは、次のスイッチはカプセル化解除にパイプ モードを使用します。

- Cisco Nexus 92348GC-X プラットフォーム スイッチ
- Cisco Nexus 9300-FX/FX2/FX3/GX/GX2 プラットフォーム スイッチ
- Cisco Nexus 9500-EX/FX/GX ライン カード
- Cisco Nexus 9800 プラットフォーム スイッチ

（注） Cisco Nexus 9500-R ライン カードは、カプセル化解除に Uniform モードを使用します。

セグメント ルーティング（SR）Traceroute のサポートを提供するためには、残りのプラットフォームとのモード動作の違いが必要です。SR traceroute は、MPLS エコー要求を伝送するパケットの存続可能時間（TTL）値の有効期限に依存しています。

パイプ モードの動作では、TTL 値はパケットが最初にカプセル化されたときに内部 IP ヘッダーの元の値を保持し、外部ヘッダーを削除します。これは、SR traceroute の誤動作に影響します。ただし均一モードでは、外部ヘッダーの TTL 値は、カプセル化解除中に内部 IP ヘッダーにコピーされ、MPLS トンネルを通過することで減少した値が保持されます。

必要に応じて、内部 IP ヘッダーの DSCP 値を保持するために、トンネルのカプセル化解除モードを Uniform to Pipe および Pipe to Uniform モードに構成できます。

```
switch(config)#mpls qos pipe-mode
```

このコマンドの no 形式が存在し、スイッチをデフォルトの均一モードに設定します。

（注） このコマンドは、Cisco Nexus 9500-R ライン カードにのみ適用されます。

このコマンドは、MPLS インターフェイス/ラベルを作成する前に構成する必要があります。Pipe-mode コマンドを使用して構成する場合、SR traceroute は使用できません。

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、ポリシングに基づくマーキング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。
- 入力ライン カードからのトラフィックがライン カードへのファブリック モジュール パスを経由する場合、MPLS 入力 LSR ノードとして機能するライン カードは ECN マーキングをサポートしません。このことは、N9K-X9700-EX および N9K-X9700-FX ライン カードを搭載した Cisco Nexus 9500 プラットフォーム スイッチで発生します。

- 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。
- Cisco NX-OS リリース 9.3(5) 以降、MPLS QoS は Cisco Nexus 9364C-GX、Cisco Nexus 9316D-GX、および Cisco Nexus 93600CD-GX スイッチでサポートされています。
- PFC は、MPLS QoS および VXLAN MPLS DCI ではサポートされていません。
- インターフェイスからキューイング ポリシーを削除しても、以前のマイクロ バースト統計情報は残ります。残りのレコードをクリアするには、clear queuing burst-detect コマンドを使用します。
- 出力 PE（sr decap）の入力ポートの RACL はサポートされていません。
- ラベルに EXP 値を書き込むには、PE に明示的なポリシーが必要です。ポリシーがない場合、デフォルトの EXP 値は 7 です。

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：
	`switch(config)# policy-map type qos pmap1`
	`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-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。

	コマンドまたはアクション	目的

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

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

	コマンドまたはアクション	目的

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)#`
		グローバル設定モードを開始します。

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

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。

	コマンドまたはアクション	目的

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。

	コマンドまたはアクション	目的
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。

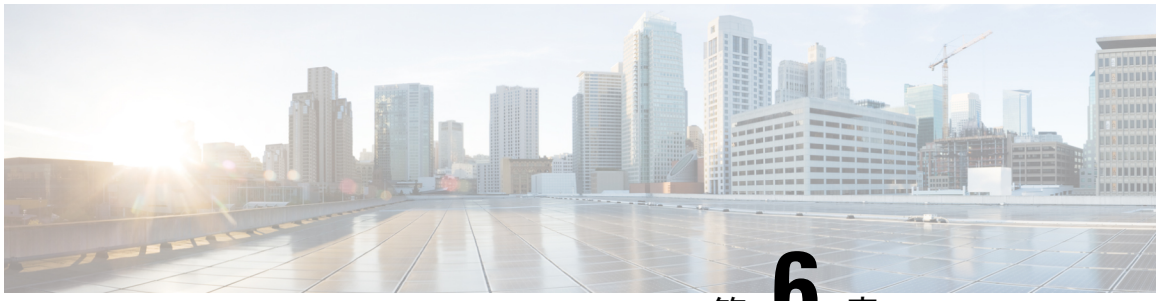
コマンド	説明
show hardware internal forwarding table utilization	MAX ラベル エントリと Used ラベル エントリに関する情報を表示します。
show class-map	インターフェイス クラス マッピングの統計情報を表示します。
show policy-map system type qos input	すべてのインターフェイスのすべてのクラスに一致したパケットを示す累積統計を表示します（EVPN トンネルの場合のみ）。詳細については、この表に続く出力例を参照してください。
show policy-map type qos interface 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 6 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。

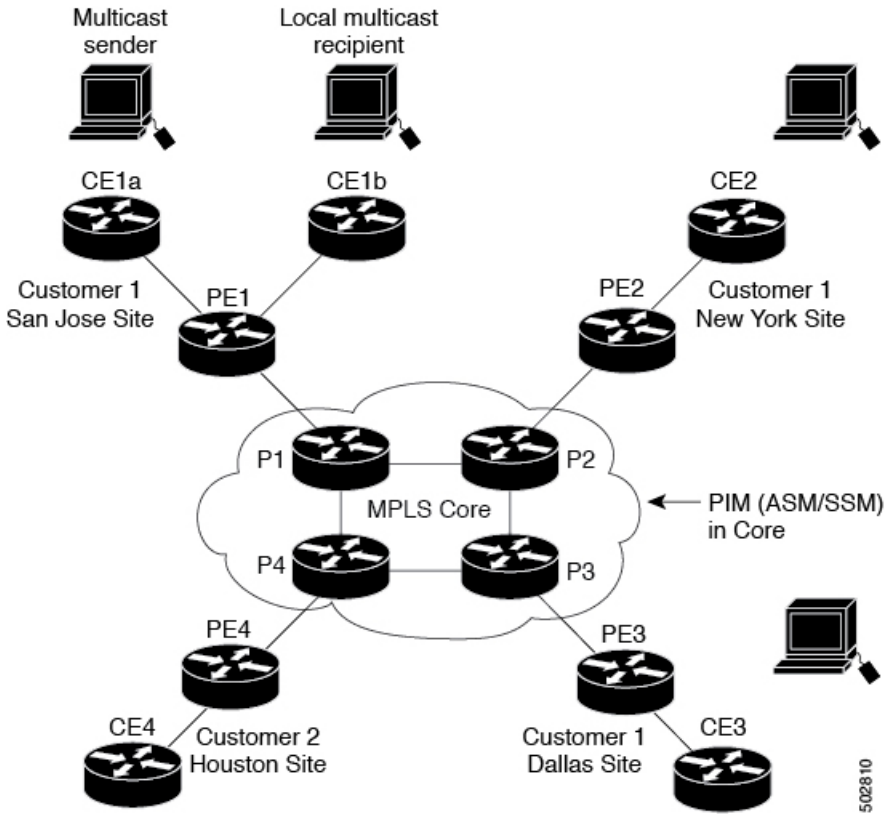
マルチキャスト配信ツリー

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。

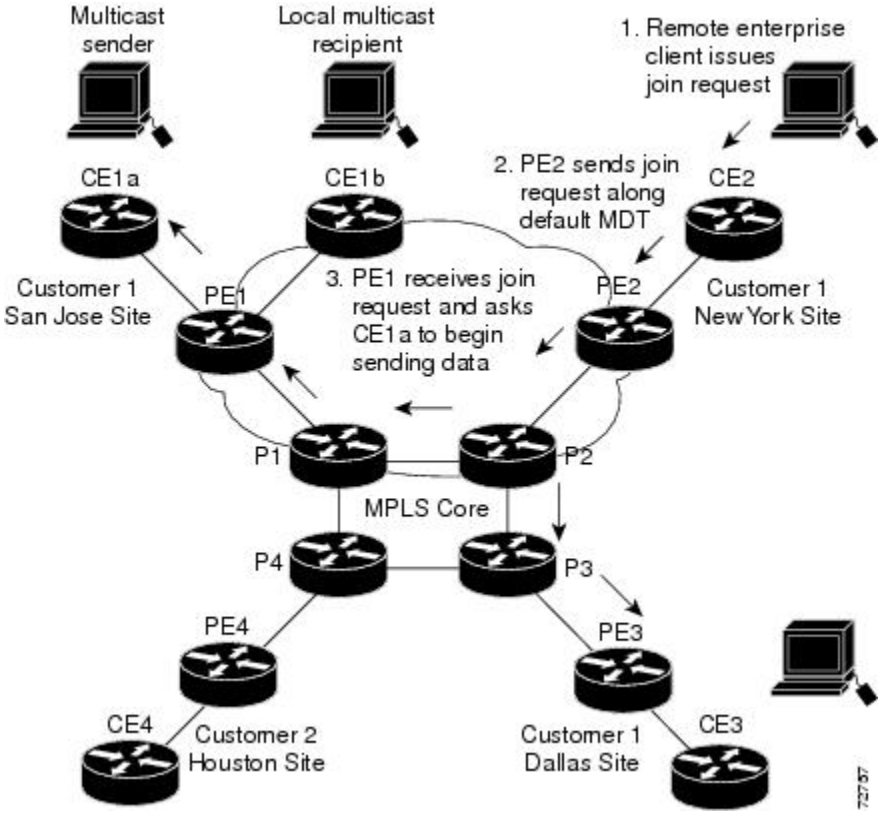
次の例のサービス プロバイダには、San Jose、New York、Dallas にオフィスがあるマルチキャストカスタマーがいます。San Joseでは、一方向のマルチキャストプレゼンテーションが行われています。サービス プロバイダー ネットワークでは、このカスタマーと関連する 3 つすべてのサイト、および別のエンタープライズカスタマーのHoustonサイトがサポートされます。エンタープライズ カスタマーのデフォルト MDT は、プロバイダのルータ P1、P2、P3、およびその関連 PE ルータから構成されています。PE4 は別のカスタマーに関連付けられているため、デフォルト MDT の一部ではありません。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 *2:* 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 *3:* 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。

MVPN の前提条件

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。
- MVPN は、-R/-RX ライン カード（N9K-X96136YC-R ライン カードを除く）を搭載した Nexus 9500 プラットフォーム スイッチでサポートされます。
- 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。
- VRF にデフォルト以外の MTU を設定できるのは、VRF から MDT MTU 設定を削除した場合に限られます。これは、デフォルト以外の MDT MTU を持つ VRF が使用可能なスイッチで MTI がダウンしている場合に発生します。
- ハードウェアの制限により、MTI TX パケット数はサポートされていません。ただし、すべての MTI RX パケットとバイト カウントがサポートされます。

MVPN のデフォルト設定

表 *2:* 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 9500-R 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。

	コマンドまたはアクション	目的
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

	コマンドまたはアクション	目的
		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。

MVPN のための BGP における MDT アドレス ファミリの設定

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。

	コマンドまたはアクション	目的
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 ipv4unicast`
	`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

	コマンドまたはアクション	目的
		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。

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：	ネイバー 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-switch-af)# neighbor 192.168.1.1`	
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。

データ MDT の設定

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：	グローバル 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	**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 の設定の確認

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 *3 : 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。

コマンド	目的
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 の設定例

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
```

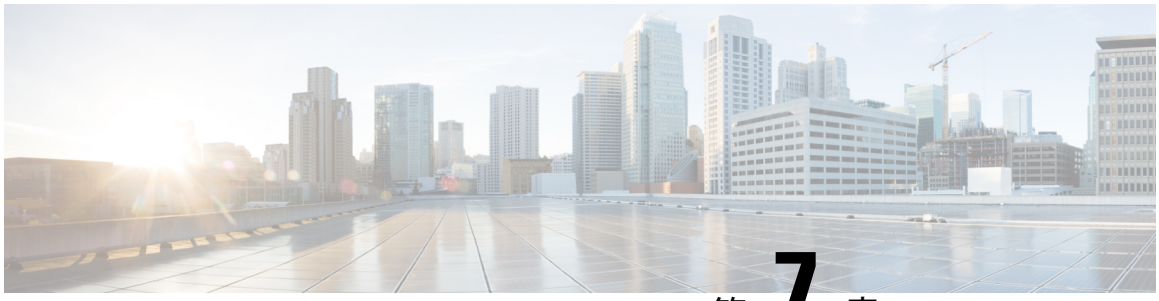
```
  ip pim ssm range 232.0.0.0/8
  mdt default 235.1.1.1
  mdt asm-use-shared-tree
ip pim rp-address 10.11.0.2 -list 224.0.0.0/8
ip pim rp-address 10.11.0.4 -list 235.0.0.0/8
ip pim ssm range 232.0.0.0/8
```

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 I 部

MPLS レイヤ 3 VPNs



第 7 章

『Configuring MPLS Layer 3 VPNs』

この章では、Cisco Nexus 9508 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

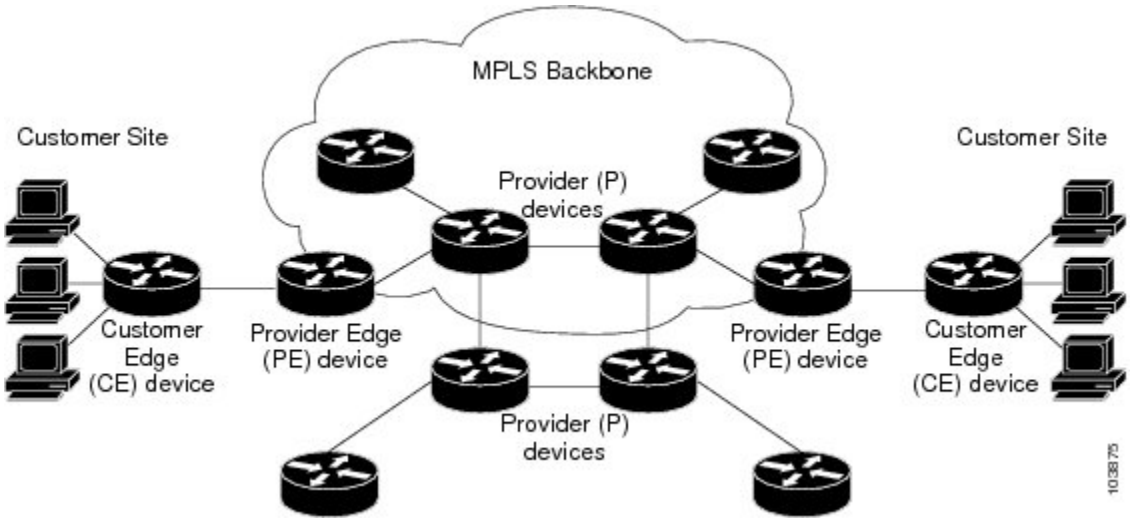
MPLS レイヤ 3 VPN はピア モデルに基づいており、これにより、サービス プロバイダーおよびカスタマーは、レイヤ3のルーティング情報を交換できます。プロバイダーは、カスタマーサイト間でデータをリレーします。このとき、カスタマーが直接何かを行う必要はありません。

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。
- 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 *4: 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。

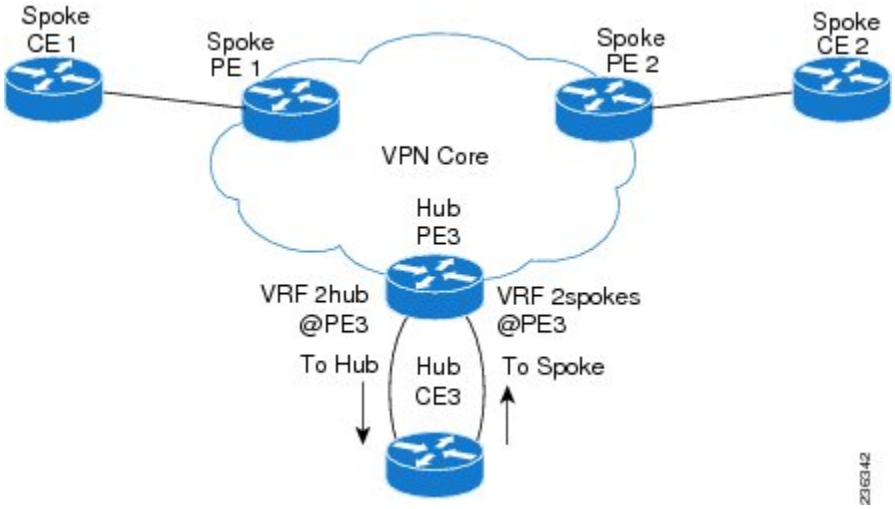
1 対 1 の関係は、カスタマー サイトと VPNs 間に必ずしも存在する必要はありません。1 つのサイトを複数の VPNs のメンバにできます。ただし、サイトは、1 つの VRF とだけ関連付けることができます。カスタマー サイトの VRF には、そのサイトがメンバとなっている VPNs からサイトへの、利用できるすべてのルートが含まれています。

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 *5:* 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。

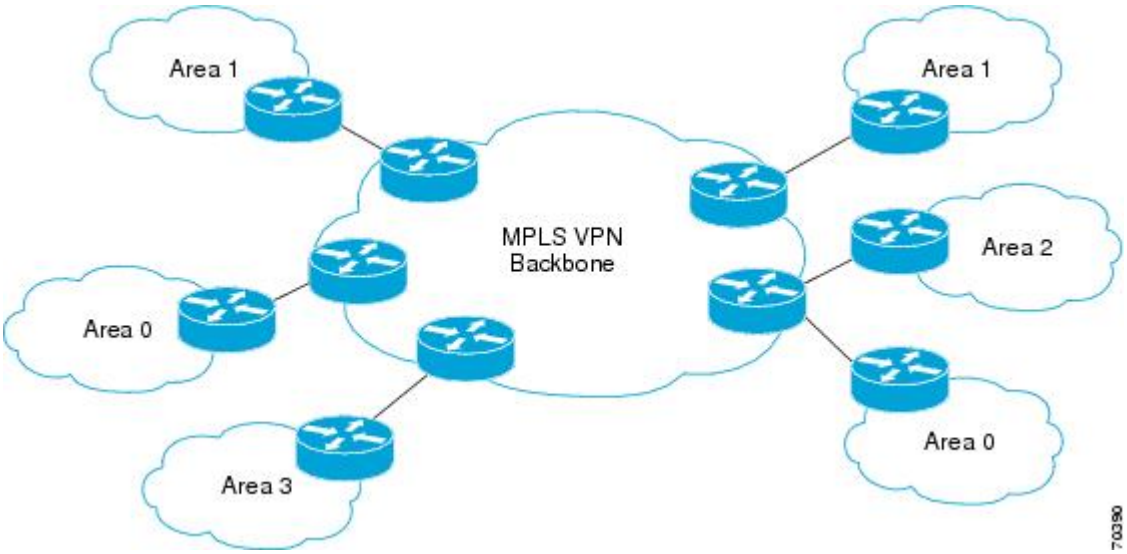
MPLS VPN のための OSPF 模造リンクのサポート

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。
- MPLS IP 転送はサポートされていないため、トンネルエンドポイントを終端するインターフェイスで有効になっていないことを確認してください。
- 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。
- N9K-X9636C-R および N9K-X9636Q-R ラインカードを搭載した Cisco Nexus 9508 プラットフォーム スイッチのトラップ解決のハードウェア制限のため、インバンド経由でのスーパーバイザ バウンド パケットに uRPF が適用されない場合があります。
- -R シリーズ ラインカードを備えた Cisco Nexus 9500 プラットフォーム スイッチでは、ブリッジ トラフィックが RACL にヒットしないように、RACL はルーティングされたトラフィックにのみ適用されます。これは、すべてのマルチキャスト OSPF 制御トラフィックに適用されます。
- -R シリーズ ラインカードを備えたCisco Nexus 9500 プラットフォーム スイッチでは、SUP への送信時に、明示的NULL ラベルを持つ制御パケットは優先されません。これにより、明示的にNULL が設定されている場合、制御プロトコルのフラッピングが発生する可能性があります。
- 500K の規模でのラベルごとの統計は、ハードウェアの制限のため、-R シリーズ ライン カードを備えた Cisco Nexus 9500 プラットフォーム スイッチではサポートされていません。
- -R シリーズ ラインカードを備えた Cisco Nexus 9500 プラットフォーム スイッチでの ARP スケーリングは、すべての 64K MAC が異なる場合、64K に制限されます。この制限は、インターフェイスに複数の等コストマルチパス（ECMP）が構成されている場合にも適用されます。
- 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。
- MPLS レイヤ 3 VPN は、M3 シリーズ モジュールでサポートされています。
- 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 トラフィック エンジニアリング（RSVP）は、N9K-X9636C-R および N9K-X9636Q-R ラインカードを備えた Cisco Nexus 9508 プラットフォーム スイッチではサポートされていません。
- Cisco NX-OS リリース 9.3(1) 以降、BGP プレベスト パス挿入ポイント (POI) の動作が変更されました。このリリースでは、NX-OS RPM、BGP、および HMM ソフトウェアは単一のコスト コミュニティ ID（内部ルートの場合は 128、外部ルートの場合は 129）を使用して、BGP VPNv4 ルートを EIGRP 発信ルートとして識別します。コスト コミュニティ ID 128 または 129 に設定されたプレベスト パス値を持つルートのみが、コスト外部コミュニティとともに URIB にインストールされます。上記のコスト コミュニティ ID を伝える非 EIGRP 発信ルートは、プレベスト パス コスト コミュニティとともに URIB にインストールされます。その結果、URIB はこのコストを使用して、管理的距離とは異なる、iBGP を介して学習したルートとバックドア EIGRP の間のより適切なルートを識別します。

  コスト コミュニティ ID 128 または 129 に設定されたプレベスト パス値を持つルートのみが、コスト外部コミュニティとともに URIB にインストールされます。

MPLS レイヤ 3 VPNs のデフォルト設定

表 *4:* 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
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』

OSPF ドメイン ID とタグについて

VRF 内の OSPF ルータ インスタンスの domain_ID を設定できます。OSPF では、Cisco NX-OS は domain_ID とドメイン タグを使用して、プロバイダー エッジ（PE）またはカスタマー エッジ（CE）での BGP ルート再配布の側面を制御します。

- 再配布される OSPF ルートのプライマリおよびセカンダリ domain_ID を設定できます。
- OSPF は、ドメイン タグを使用して OSPF プロセス ID を識別します。

ドメイン ID とドメイン タグの Cisco NX-OS 実装は、RFC 4577 に準拠しています。

（注） OSPF のプライマリとセカンダリの domain_ID とドメイン タグは、MPLS L3VPN 機能が有効になっている場合にのみ使用できます。

PE および CE 境界での OSPF の設定

ドメイン ID とドメイン タグを使用することで、NX-OS を設定して OSPF ルートを BGP ネットワークに再配布できます。また、BGP 再配布ルートを PE と CE の境界で OSPF に受信させることができます。次の項を参照してください。

- OSPF ドメイン ID とタグについて （78 ページ）
- OSPF ドメイン ID の構成 （79 ページ）

• セカンダリ ドメイン ID の構成 （80 ページ）
• OSPF ドメイン タグの設定 （79 ページ）

OSPF ドメイン タグの設定

ドメイン タグは、NX-OS が PE または CE で BGP に再配布する OSPF プロセス インスタンス番号を指定します。

始める前に

MPLS と OSPFv2 が有効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-1# configure terminal`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`switch-1(config)#`
		端末の構成を開始します

ステップ 2	**router ospf** *process-tag*
	例：
	`switch-1(config)# router ospf 101`
	`switch-1(config-router)#`
		ルータ構成モードを開始して、OSPF ルータ インスタンスを構成します。プロセス タグは、ルータを識別する 1 ～ 20 文字の英数字文字列です。

ステップ 3	**vrf** *vrf-name*
	例：
	`switch-1(config-router)# vrf pubstest`
	`switch-1(config-router-vrf)#`
		OSPF の特定の VRF インスタンスを入力します。VRF 名は、VRF を識別する 1 ～ 32 文字の英数字文字列です。

ステップ 4	**ospf domain-tag** *as-number*
	例：
	`switch-1(config-router-vrf)# domain-tag 9999`
	`nxosv2(config-router-vrf)#`
		ドメイン タグを設定します。ドメイン タグは、AS 番号を識別する 0 ～ 2147483647 の英数字の文字列です。

OSPF ドメイン ID の構成

VRF 内の OSPF ルータ インスタンスの domain_ID を設定して、CE または PE での OSPF への BGP ルートの再配布を制御できます。

この機能を削除するには、このコマンドの **no domain-id** 形式を使用します。

始める前に

OSPF domain_ID 機能を使用するには、MPLS L3VPN 機能と OSPFv2 機能の両方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-1# configure terminal`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`switch-1(config)#`
		端末の構成を開始します

ステップ 2	**router ospf** *process-tag*
	例：
	`switch-1(config)# router ospf 101`
	`switch-1(config-router)#`
		ルータ構成モードを開始して、OSPF ルータ インスタンスを構成します。プロセス タグは、ルータを識別する 1 ～ 20 文字の英数字文字列です。

ステップ 3	**vrf** *vrf-name*
	例：
	`switch-1(config-router)# vrf pubstest`
	`switch-1(config-router-vrf)#`
		OSPF の特定の VRF インスタンスを入力します。VRF 名は、VRF を識別する 1 ～ 32 文字の英数字文字列です。

ステップ 4	**domain-id** { *id* | **type** *domain-type* **value** *value* | **Null** } 例： `switch-1(config-router-vrf)# domain-id 19.0.2.0`	domain_ID と追加のパラメータを設定します。 • *id* は、ドメイン ID をドット付き 10 進表記で指定します (例: 1.2.3.4)。 • *type* は、0005 などの 4 バイト表記でドメイン タイプを指定します。 • *value* は、ドメイン値を 6 バイトの 16 進表記で指定します (例: 0x0005)。 Null 引数を使用して、domain_ID をクリアすることができます。

セカンダリ ドメイン ID の構成

VRF 内の OSPF ルータ インスタンスにセカンダリ domain_ID を設定して、CE または PE での OSPF への BGP ルートの再配布を制御できます。

domain-id Null コマンドを使用して、domain_ID を構成解除します。

始める前に

OSPFv2 および MPLS 機能が有効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-1# configure terminal`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`switch-1(config)#`
		端末の構成を開始します

ステップ 2	**router ospf** *process-tag*
	例：
	`switch-1(config)# router ospf 101`
	`switch-1(config-router)#`
		ルータ構成モードを開始して、OSPF ルータ インスタンスを構成します。プロセス タグは、ルータを識別する 1 ～ 20 文字の英数字文字列です。

ステップ 3	**vrf** *vrf-name*
	例：
	`switch-1(config-router)# vrf pubstest`
	`switch-1(config-router-vrf)#`
		OSPF の特定の VRF インスタンスを入力します。VRF 名は、VRF を識別する 1 ～ 32 文字の英数字文字列です。

ステップ 4	**domain-id** { *id* | **type** *domain-type* **value** *value* | **Null** } 例： `switch-1(config-router-vrf)# domain-id 19.0.2.0`	自律システムの domain_ID を設定します。

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）。
- MPLS トラフィック エンジニアリング リソース予約プロトコル（RSVP）。

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：	MPLS フィーチャ 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` `switch(config)#`	

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 mpls l3vpn**
	例：
	`switch(config)# feature 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。

	コマンドまたはアクション	目的
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 フィーチャ 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

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。

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

	コマンドまたはアクション	目的
		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	**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。

	コマンドまたはアクション	目的
ステップ 12	**copy running-config startup-config** 例： `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 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 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、

	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 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`	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

	コマンドまたはアクション	目的
	`switch(config-vrf-af)#`	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 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。

	コマンドまたはアクション	目的

ステップ 5	**address-family ipv4 unicast**
	例：
	`switch(config-router-vrf)# 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 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 は、大文字と小文字が区別される 20 文字以下の任意の英数字文字列にできます。
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：	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

	`switch(config)# feature ospf`
	
	`switch(config)#`	

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	**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 7	**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 8	**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 9	**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 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。

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 eigrp**
	例：
	`switch(config)# feature eigrp`
	`switch(config)#`
		EIGRP 機能を有効にします。

ステップ 3	**router eigrp** *instance-tag* 例： `switch(config)# router eigrp 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)#`
	`redistribute bgp 235354 route-map`
	`mtest1`
		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 番号としては、16 ビット整数または 32 ビット整数があり得ます。後者の場合、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。

ステップ 6	**update-source loopback [0 | 1]** 例：	BGP セッションの送信元アドレスを指定します。

	コマンドまたはアクション	目的
	`switch(config-router-neighbor)# update-source loopback 0#`	
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。
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。

	コマンドまたはアクション	目的
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 フィーチャ セットをインストールします。

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：
	`switch(config)# vrf context 2hub`
	`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 ハブの VPN ルーティング イン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：	ルート識別子を設定します。route-distinguisher 引数によって、8 バイ

	コマンドまたはアクション	目的

	`switch(config-vrf)# rd 1.2:1`
	
	`switch(config-vrf)#`
		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。
		  • 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 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。 • 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 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** コマンドを設定するか、受信 CE ルータで **allowas-in** 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
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。

	コマンドまたはアクション	目的

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： `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。

	コマンドまたはアクション	目的

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。

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。

	コマンドまたはアクション	目的

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：	グローバル 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	**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。

	コマンドまたはアクション	目的
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 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。

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：	グローバル 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	**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。

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。

	コマンドまたはアクション	目的
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。 。
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。

- 認識しているスポーク ルータで allowas-in 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 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：	エントリを iBGP ネイバー テーブルに追加します。

	コマンドまたはアクション	目的
	`switch(config-router)# neighbor 63.63.0.63 remote-as 100` `switch(config-router-neighbor)#`	• 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は、次のコマンドを構成します。 （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。
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。

	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。

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、N9K-X9636C-R、N9K-X9636C-RX、および N9K-X9636Q-R ラインカードを備えた Cisco Nexus 9508 スイッチは、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

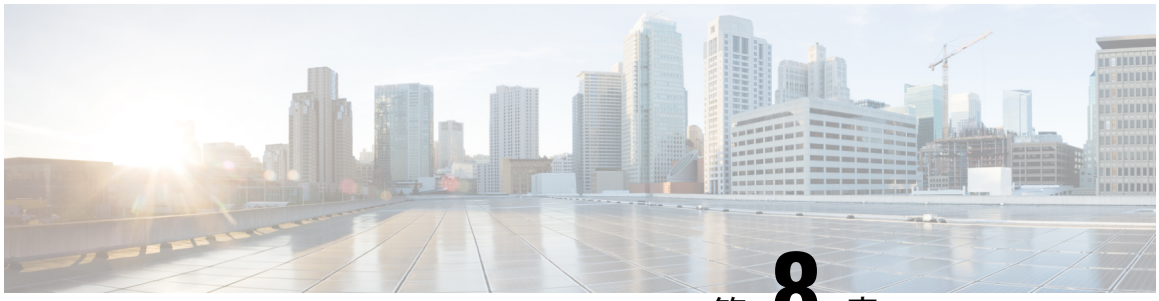
ステップ 2	**feature bgp**
	例：
	`switch(config)# feature bgp`
	`switch(config)#`
		BGP 機能をイネーブルにします。

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。。

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：	スイッチのソフトウェア情報を表示します。

	コマンドまたはアクション	目的
	`switch(config)# show module internal sw info`	
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 8 章

MPLS レイヤ 3 VPN ラベル割り当ての設定

この章では、Cisco Nexus 9508 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 レイヤ 3 VPN ラベル割り当てについて

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 レイヤ 3 VPN ラベル割り当ての前提条件

レイヤ 3 VPN ラベルの割り当てには、次の前提条件があります。

- ネットワークに MPLS、および LDP と RSVP TE のいずれか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 レイヤ3VPN ラベル割り当てに関する注意事項と制限事項

レイヤ 3 VPN 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。

- 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。

- 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 レイヤ 3 VPN ラベル割り当てのデフォルト設定

表 *5:* デフォルトのレイヤ *3 VPN* ラベル割り当てパラメータ

パラメータ	デフォルト
レイヤ 3 VPN 機能	無効
ラベル割り当てモード	プレフィックス単位

MPLS レイヤ 3 VPN ラベル割り当ての設定

VRF 単位でのレイヤ 3 VPN ラベル割り当てモードの設定

レイヤ 3 VPN の VRF 単位でのレイヤ 3 VPN 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。PE 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。

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 *6:* 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 たは撤回？
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 トバタイズ
2				SAFI-4	アドバタイ
3			NHU	SAFI-1	アドバタイ

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 たは撤
4				SAFI-4	出金
5		いいえ	NHS	SAFI-1	アドバ
6				SAFI-4	出金
7			NHU	SAFI-1	アドバ
8				SAFI-4	出金
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 トバタ NbrKno 回。
10				SAFI-4	アドバ
11			NHU	SAFI-1	出金
12				SAFI-4	アドバ
13		いいえ	NHS	SAFI-1	アドバ
14				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	アドバタイたは撤回?
15			NHU	SAFI-1	出金
				SAFI-4	アドバタイ

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

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

	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 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：
	`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 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。

	コマンドまたはアクション	目的
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 レイヤ 3 VPN ラベル割り当ての設定の確認

レイヤ 3 VPN 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 *7: MPLS* レイヤ *3 VPN* ラベル割り当ての設定の確認

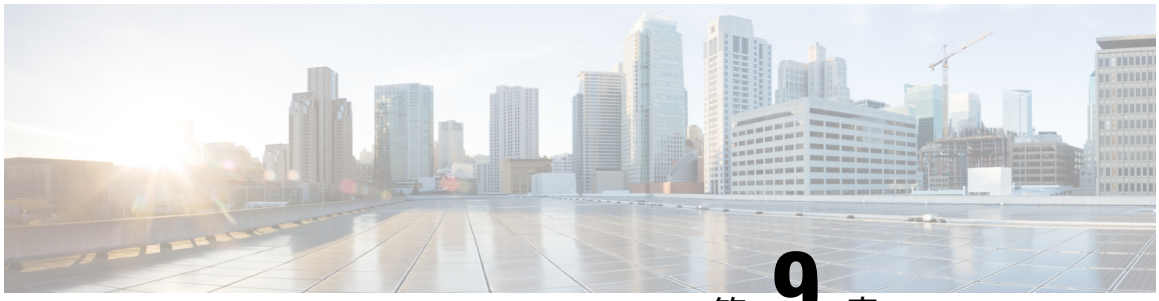
コマンド	目的
show bgp l3vpn [**detail**] [**vrf** *v rf-name*]	VRF での BGP のレイヤ 3 VPN 情報を表示します。
show bgp vpnv4 unicast labels [**vrf** *v rf-name*]	BGP のラベル情報を表示します。
show ip route [**vrf** *v rf-name*]	ルートのラベル情報を表示します。

MPLS レイヤ 3 VPN 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 9 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 Nexus 9508 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）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。

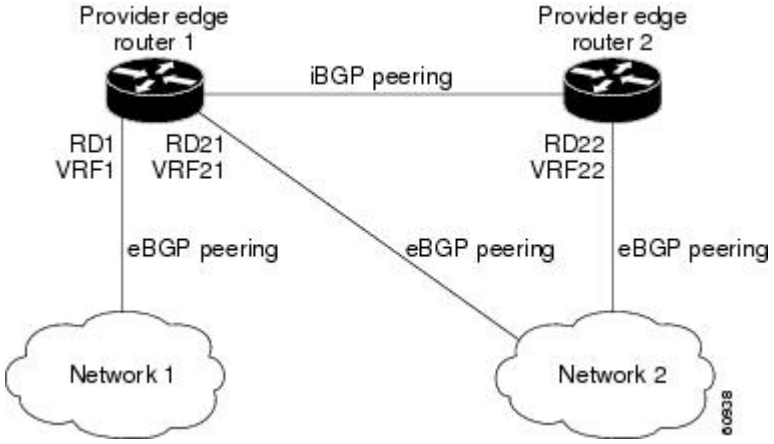
レイヤ 3 VPN ロード バランシングは、PE ルーターと VPN で IPv4 と IPv6 をサポートします。

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 *6 : 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

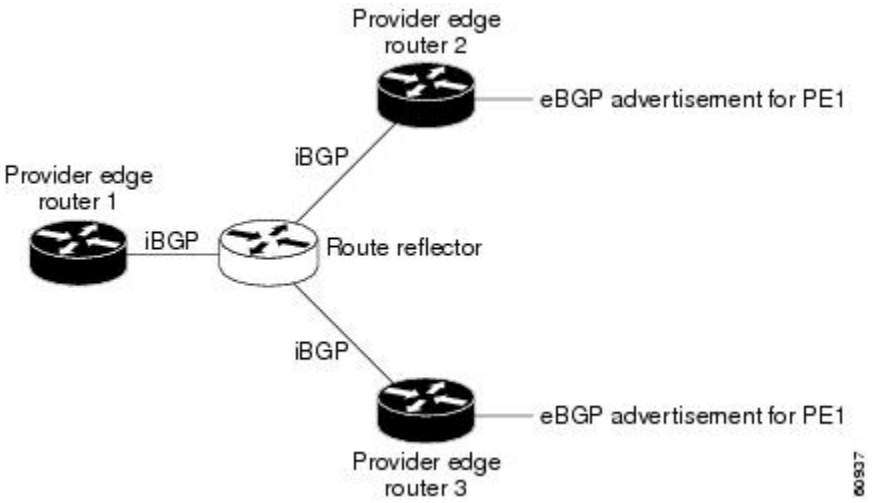
ルート リフレクタは、PE ルータでのセッション数を減らし、レイヤ 3 VPN ネットワークの拡張性を向上させます。ルート リフレクタは、PE ルータとピアリングするために、受信したすべての VPN ルートを保持します。異なる PE では、異なるルート ターゲット タグ付き VPNv4 および VPNv6 ルートが必要になる場合があります。ルート リフレクタはまた、VRF 設定が変更されたときに特定のルート ターゲットのリフレッシュを PE に送信する必要がある場合があります。すべてのルートを保存すると、ルート リフレクタのスケーラビリティ要件が増大します。ルート リフレクタはルート ターゲット 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 *7:* 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 *8 :* 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*）

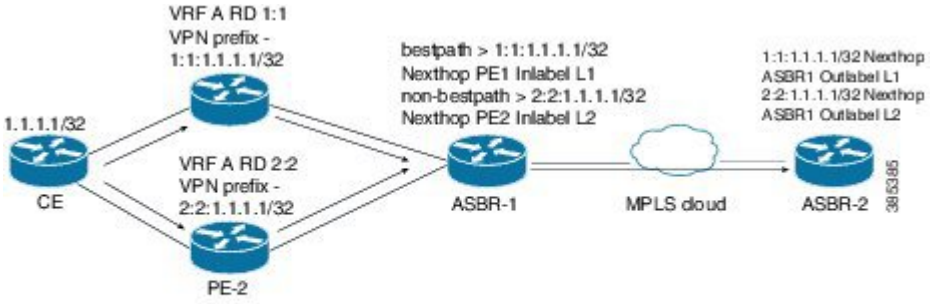
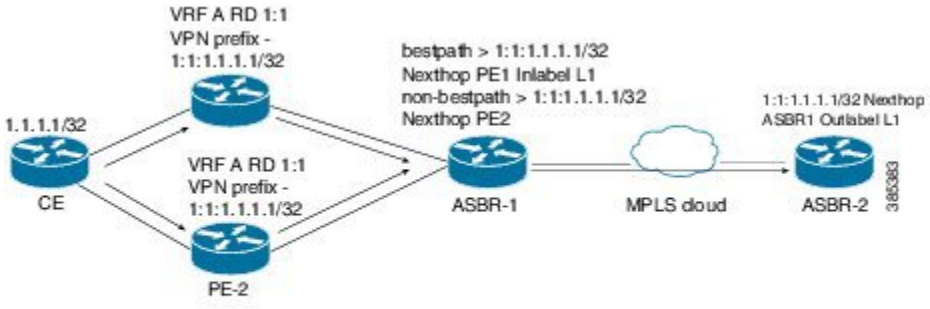


図 22-4 に示すように、BGP VPN マルチパス機能を使用すると、両方の PE デバイスの VRF が同じ RD を使用できるようになります。このようなシナリオでは、ASBR-1 は両方の PE デバイスから同じプレフィックスを受信します。ASBR-1 は、受信したプレフィックスに 1 つの Inter-AS オプション B ラベル、Inlabel L1 のみを割り当て、VPN ルートを ASBR-2 にアドバタイズします。この場合、両方の PE デバイスを使用するトラフィック フローが ASBR-1 の 1 つのプレフィックスとラベルだけで確立されるため、スケール性が強化されます。

図 *9:* 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、N9K-X9636C-R、N9K-X9636C-RX、および N9K-X9636Q-R ライン カードを搭載した Cisco Nexus 9508 プラットフォーム スイッチで設定できます。
- Cisco NX-OS リリース 9.3(3) 以降であれば、MPLS レイヤ 3 VPN ロード バランシングは、Cisco Nexus 9364C-GX、Cisco Nexus 9316D-GX、および Cisco Nexus 93600CD-GX スイッチでは IP ACL を設定できます。
- Cisco NX-OS リリース 10.4(1)F 以降では、ポート チャネル ロード バランシングでスイッチの mpls ロード バランシングを構成できます。この機能は、Cisco Nexus 9300 -EX/FX/FX2/FX3/GX/GX2 TOR および EOR プラットフォーム スイッチでサポートされます。構成に関する詳細については、『*Cisco Nexus 9000* シリーズ *NX-OS* インターフェイス構成ガイド』を参照してください。
- Cisco Nexus 9348GC-FX3PH スイッチには、ポート 41 ー 48 の半二重による機能制限があります。
- 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。
- N9K-X9636Q-R および N9K-X9636C-R ラインカードを搭載した Cisco Nexus 9508 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、N9K-X9636C-RX ラインカードを搭載した Cisco Nexus 9508 プラットフォーム スイッチでは最大 470K 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。
- Cisco NX-OS リリース 10.1(1) 以降、Cisco Nexus 9300-FX2、9300-GX、9300-GX2 プラットフォーム スイッチでは、mpls ip 転送が有効になっているインターフェイスでパケットが受信された場合の dot1q タグの追加または削除はサポートされていません。 以前のリリースで、CLI **feature mpls segment-routing** が有効になっている場合、または **mpls load-sharing** [**label-only** | [**label-ip**] が設定されている場合、dot1q タグの追加または削除はサポートされていません。
- Cisco Nexus 9300-EX、9300-FX、9300-EX-LC、9300-FX-LC、および N9K-C9364C、N9K-C9508-FM-E2、N9K-C9516-FM-E2、および N9K-C9332C プラットフォームスイッチで、CLI **feature mpls segment-routing** が有効になっている場合、または **mpls load-sharing** [**label-only** | [**label-ip**] が設定されている場合、dot1q タグの追加または削除はサポートされていません。
- Cisco Nexus 9300-EX および 9300-EX-LC プラットフォーム スイッチでは、mpls ラベルまたは SRC/DST-IP に基づくポート チャネルおよび ecmp ロード シェアリングは、CLI **mpls**

load-sharing label-ip が設定されている場合でも機能しません。ただし、**label-only** は機能します。

- VXLAN BUM トラフィックは、mpls ロード バランシングが有効になっている純粋な L2 スイッチを通過してはなりません （**mpls load-sharing** [**label-only** | [**label-ip**]）。

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 *8:* 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：	MPLS フィーチャ セットをイネーブルにします。

	コマンドまたはアクション	目的
	`switch(config)# feature-set mpls`	
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。

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。

	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。

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： `switch(config-router-af)# maximum-paths eibgp 3`	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します。指定できる範囲は 1 ～ 32 です。

MPLS ECMP 負荷共有の設定

Cisco NX-OS リリース 9.3(1) 以降、ラベルに基づいて MPLS ECMP 負荷共有を設定できます。この機能は、Cisco Nexus N9K-X9700-EX および N9K-X9700-FX ラインカードを搭載した Cisco Nexus 9200、Cisco Nexus 9300-EX、Cisco Nexus 9300-FX、および Cisco Nexus 9500 プラットフォーム スイッチでサポートされています。

Cisco NX-OS リリース 9.3(3) 以降、この機能は Cisco Nexus 9364C-GX、Cisco Nexus 9316D-GX、および Cisco Nexus 93600CD-GX スイッチでサポートされています。

手順

	コマンドまたはアクション	目的

ステップ 1	**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	**mpls load-sharing [label-only | [label-ip]**
	例：
	`switch(config)# mpls load-sharing label-only`
	`switch(config)# mpls load-sharing label-ip`
		mpls ラベルに基づいて負荷共有を設定します。label-only オプションはラベルに基づいて負荷共有を設定し、label-ip オプションはラベルと IP アドレスに基づいて負荷共有を設定します。

ステップ 4	**copy running-config startup-config** 例： `switch(config)# 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 ECMP 負荷共有の確認

ECMP 負荷共有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 *9: MPLS ECMP* 負荷共有設定の確認

コマンド	目的
show mpls load-sharing	mplsハッシュに使用されるラベルの数と、ハッシュに使用される IP フィールドを表示します。

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
```

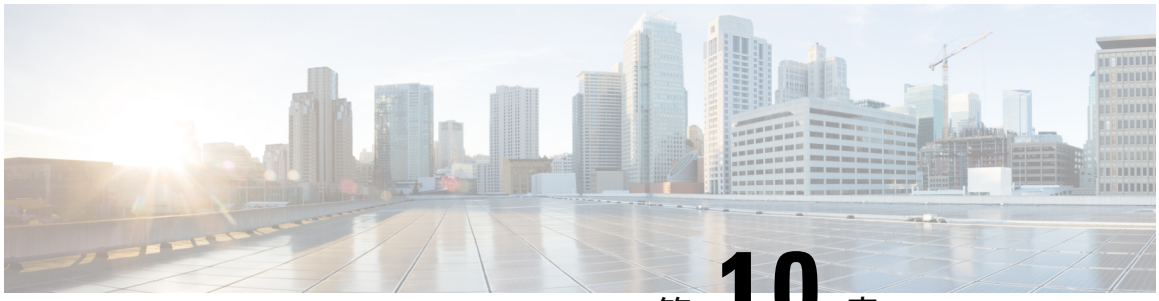
例：MPLS レイヤ 3 VPN コスト コミュニティ

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 II 部

セグメント ルーティング



第 10 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。

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 （153 ページ）を参照してください。

MPLS の NetFlow

NetFlow は入力 IP パケットについてパケット フローを識別し、これらのパケット フローに基づいて統計情報を提供します。NetFlowのためにパケットやネットワーキングデバイスを変更する必要はありません。フロー用に NetFlow が収集したデータをエクスポートするには、フロー エクスポータを使用し、このデータを Cisco Stealthwatch などのリモート NetFlow コレクタにエクスポートします。Cisco NX-OS は、NetFlow エクスポート用のユーザ データグラム プロトコル（UDP）データグラムの一部としてフローをエクスポートします。フロー用にNetFlow が収集したデータをエクスポートするには、フローエクスポータを使用し、このデータをCisco Stealthwatch などのリモートNetFlow コレクタにエクスポートします。Cisco NX-OS は、NetFlow エクスポート用のユーザ データグラム プロトコル（UDP）データグラムの一部としてフローをエクスポートします。

Cisco NX-OS リリース 9.3(1) 以降、セグメント ルーティング上の NetFlow Collector は、Cisco Nexus 9300-EX、9300-FX、9300-FX2、9500-EX、および9500-FX プラットフォーム スイッチでサポートされます。

Cisco NX-OS リリース 9.3(5) 以降、セグメント ルーティング上の NetFlow Collector は、Cisco Nexus 9300-FX3 プラットフォーム スイッチでサポートされます。

Netflow は Cisco Nexus 9300-GX プラットフォーム スイッチではサポートされません。

NetFlow Collector は、シングルおよびダブル MPLS ラベルの両方をサポートします。エクスポータの宛先設定のデフォルトおよび非デフォルト VRF の両方がサポートされます。NetFlow は、MPLS データ パスをサポートしていません。

セグメントルーティングは単一のラベルをサポートしないため、BGP ネイバーで **address-family ipv4labeled-unicast** コマンドを設定し、bgp 設定で **allocate-label** コマンドを設定する必要があります。

sFlow コレクタ

サンプリングされた Flow（sFlow）を使用すると、スイッチやルータを含むデータネットワーク内のリアルタイムトラフィックをモニターできます。sFlow では、トラフィックをモニタするためにスイッチとルータ上の sFlow エージェント ソフトウェアでサンプリング メカニズムを使用して、サンプル データを中央のデータ コレクタに転送します。

Cisco NX-OS リリース 9.3(1) 以降、セグメント ルーティング上の sFlow コレクタは Cisco Nexus 9300-EX、9300-FX、9300-FX2、9500-EX、および9500-FX プラットフォーム スイッチでサポートされます。

Cisco NX-OS リリース 9.3(5) 以降、セグメント ルーティング上の sFlow コレクタは Cisco Nexus 9300-FX3 プラットフォーム スイッチでサポートされます。

sFlow は Cisco Nexus 9364C-GX、Cisco Nexus 9316D-GX、および Cisco Nexus 93600CD-GX スイッチではサポートされていません。

sFlow 設定の詳細については、「*sFlow* の設定」のセクションを参照してください。『*Cisco Nexus 9000* シリーズ *NX-OS* システム管理設定ガイド、リリース *9.3(x)*』に掲載されています。

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、FEX モジュールではサポートされていません。
- Cisco NX-OS リリース 9.3(1) 以降、**segment-routing mpls** コマンドは **segment-routing** に変更されました。
- -R シリーズ ラインカードを搭載した Cisco Nexus 9504 および 9508 プラットフォーム スイッチで MPLS セグメント ルーティングを有効にすると、BFD セッションがダウンしたり、戻ったりする場合があります。BGP ピアリングも、BFD で構成されている場合、ダウンしてからアップします。BGPセッションがダウンすると、ハードウェアからルートが取り消されます。これにより、BGPセッションが再確立されてルートが再インストールされるまで、パケット損失が発生します。ただし、いったん BFD が起動すると、追加のフラップは発生しません。

- セグメント ルーティングは、IGP (OSPF など) の下で、または BGP での AF ラベル付きユニキャストによって実行できます。
- セグメント ルーティングは、Cisco Nexus 9300-FX プラットフォーム スイッチおよび Cisco Nexus N9K-X9736C-FX ラインカードでサポートされています。
- セグメント ルーティングと SR-EVPN は、Cisco Nexus C31108PC-V、C31108TC-V、および C3132Q-V スイッチでサポートされています。
- Cisco NX-OS リリース 9.3(3) 以降、Cisco Nexus 9300-GX プラットフォーム スイッチ上ではレイヤ 3 VPN を設定できます。
- Cisco NX-OS リリース 9.3(3) 以降、セグメント ルーティングと SR-EVPN は Cisco Nexus 9364C-GX、Cisco Nexus 9316D-GX、および Cisco Nexus 93600CD-GX プラットフォーム スイッチでサポートされています。
- Cisco NX-OS リリース 9.3(3) 以降、隣接関係 SID と OSPF は Cisco Nexus 9364C-GX、Cisco Nexus 9316D-GX、および Cisco Nexus 93600CD-GX プラットフォーム スイッチでサポートされています。
- Cisco NX-OS リリース 9.3(3) 以降、OSPF でのセグメント ルーティング、IS-IS アンダーレイ、および BGP ラベル付きユニキャストは Cisco Nexus 9364C-GX、Cisco Nexus 9316D-GX、および Cisco Nexus 93600CD-GX プラットフォーム スイッチでサポートされています。
- 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 を設定すると、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 フィルタリングの対象）。
- Cisco Nexus 9300-EX および 9300-FX プラットフォーム スイッチの MPLS 機能では、無停止の ISSU はサポートされていません。
- スタティック MPLS、MPLS セグメント ルーティング、および MPLS ストリッピングを同時に有効にすることはできません。
- Cisco NX-OS リリース 9.3(5) 以降、MPLS ストリッピングは Cisco Nexus 9300-GX プラットフォーム スイッチでサポートされます。以下の注意事項が当てはまります。
  - MPLS ストリップ機能を動作させるには、スイッチのリロード後に、**mpls strip** および **hardware acl tap-agg** コマンドを設定する必要があります。
  - Cisco Nexus 9300-GX プラットフォーム スイッチで MPLS ストリップが有効になっている場合、ACL ログ プロセスは表示されません。
  - dot1q VLAN を使用した MPLS ストリップはサポートされていません。
  - すべての二重 VLAN タグについて、2 番目の VLAN 範囲は 2 ～ 510 である必要があります。
  - dot1q を使用した MPLS ストリップはサポートされていません。

  - PACL リダイレクトをサポートするには、入力タップ インターフェイスで **mode tap-aggregation** コマンドを実行する必要があります。

- 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) のアウトラベル 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。
- Cisco Nexus 9200 シリーズ スイッチの場合、レイヤ 3 または MPLS 隣接の隣接統計は維持されません。
- Cisco Nexus 9500 シリーズ スイッチの場合、MPLS LSP およびセグメント ルーテッド パスは、サブインターフェイス（ポート チャネルまたは通常のレイヤ 3 ポートのいずれか）ではサポートされていません。
- Cisco Nexus 9500 プラットフォーム スイッチの場合、セグメント ルーティングは非階層ルーティング モードでのみサポートされます。
- 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。
- RFC 2973 および RFC 3270 で定義されている統一モデルはサポートされていません。したがって、IP DSCP ビットはインポーズされた MPLS ヘッダーにコピーされません。
- 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 デルタ値が設定された範囲外にある場合、割り当てられたラベルは動的に生成されます。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、ラベルは動的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。
- Cisco Nexus 9000 シリーズ スイッチのオンデマンド ネクスト ホップを使用して、セグメント ルーティング トラフィック エンジニアリングを設定できます。
- セグメント ルーティングのレイヤ 3 VPN およびレイヤ 3 EVPN ステッチングは、Cisco Nexus 9000 シリーズ スイッチでサポートされています。
- Cisco NX-OS リリース 9.3(3) 以降、セグメント ルーティング用のレイヤ 3 VPN およびレイヤ 3 EVPN ステッチングは、9300-GX プラットフォーム スイッチでサポートされています。
- OSPFv2 は、Cisco Nexus 9000 シリーズ スイッチのセグメント ルーティングの IGP コントロール プレーンとして設定できます。
- セグメント ルーティングのレイヤ 3 VPN およびレイヤ 3 EVPN ステッチングは、-EX ラインカードを備えた Cisco Nexus 9364C、9200、9300-EX、および 9500 プラットフォーム スイッチではサポートされていません。
- OSPF セグメント ルーティング コマンドおよびオンデマンド ネクスト ホップを使用したセグメント ルーティング トラフィック エンジニアリングは、Cisco Nexus 9364C スイッチではサポートされていません。
- セグメント ルーティングは、Cisco Nexus 9300-FX2 および 9300-FX3 プラットフォーム スイッチでサポートされています。
- セグメント ルーティングのためのレイヤ 3 VPN およびレイヤ 3 EVPN ステッチング、OSPF セグメント ルーティング コマンド、およびオンデマンド ネクスト ホップを使用したセグメント ルーティング トラフィック エンジニアリングは、Cisco Nexus 9364C スイッチでサポートされています。
- セグメント ルーティングを介したレイヤ 3 VPN は、Cisco Nexus 3100、3200、9200、9300、9300-EX/FX/FX2/FX3 プラットフォーム スイッチ、および EX\FX と R ライン カードを搭載した Cisco Nexus 9500 プラットフォーム スイッチでサポートされています。
- セグメント ルーティング設定を削除すると、MPLS およびトラフィック エンジニアリング設定を含む、関連するすべてのセグメント ルーティング設定が削除されます。
- 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 の以前の設定がすべて失われます。

- 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 MPLS 隣接統計は、出力ラベル スタックと中間ノードのネクスト ホップに基づいて収集されます。ただし、PHP モードでは、同じスタックがすべての FEC で共有されるため、統計はすべての隣接で表示されます。
- スイッチでセグメント ルーティングが有効になっている場合、dot1Q タグ付き MPLS パケットの Q-in-Q タギングはサポートされておらず、パケットは外部タグのみで出力されます。

  例：VLAN 100 を使用する、アクセス dot1q トンネル モードの入力ポートについて考えます。着信 MPLS トラフィックには、200 の dot1Q タグがあります。通常、トラフィックは外部タグ 100、内部タグ 200 (着信パケットのタグと同じ) で送信されます。ただし、パケットは外部タグ付きで送信され、内部タグは失われます。
- 着信 MPLS パケットにタグが付いておらず、入力ポートがアクセス VLAN モードの場合、セグメント ルーティングが有効になっていれば、パケットはタグなしで出力されます。
- BGP、OSPF、および IS-IS アンダーレイを同時に使用してセグメントルーティングを構成しないことをお勧めします。
- Cisco NX-OS リリース 10.2(1q)F 以降、SR-MPLS は N9K-C9332D-GX2B プラットフォーム スイッチでサポートされます。ただし、SR PBR および MPLS ストリップ dot1q 機能は、GX2 スイッチではまだサポートされていません。

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。

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。

	コマンドまたはアクション	目的
ステップ 4	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	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。

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：
	`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	[**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：	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

	コマンドまたはアクション	目的
	`switch(config-sr-mpls)# global-block 16000 471804`	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。
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。

ラベル インデックスの構成

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。（**network** コマンド

の詳細については、Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide の「Configuring Basic BGP」の章を参照してください)。

（注） 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設定するために使用されます。BGP は、プレフィックス SID の設定のために、ルート マップの下の **set label-index** <*value*> 設定と、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には、SR-APP の設定が優先されます。

（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。

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 マップ設定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

	コマンドまたはアクション	目的
		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。

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。

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)
```

IS-IS プロトコルでのセグメント ルーティングの設定

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。

OSPFv2 プロトコルでのセグメント ルーティングの設定

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
```

トラフィック エンジニアリング用のセグメント ルーティングの設定

トラフィック エンジニアリング用のセグメント ルーティングについて

トラフィック エンジニアリング用のセグメント ルーティング（SR-TE）は、送信元と宛先のペア間のトンネルを通じて行われます。トラフィック エンジニアリング用のセグメント ルーティングでは、送信元ルーティングの概念が使用されます。送信元はパスを計算し、パケットヘッダーでセグメントとしてエンコードします。トラフィック エンジニアリング（TE）トンネルは、トンネルの入力とトンネルの宛先との間でインスタンス化された TE LSP のコンテナです。TE トンネルは、同じトンネルに関連付けられた 1 つ以上の SR-TE LSP をインスタンス化できます。

トラフィック エンジニアリング用のセグメント ルーティング（SR-TE）では、ネットワークはアプリケーション単位およびフロー単位の状態を維持する必要はありません。代わりに、パケットで提供されている転送指示に従うだけです。

SR-TE は、すべてのセグメント レベルで ECMP を使用することにより、従来の MPLS-TE ネットワークよりも効果的にネットワーク帯域幅を利用します。単一のインテリジェント ソースを使用し、残りのルータをネットワーク経由で必要なパスを計算するタスクから解放します。

SR-TE ポリシー

トラフィック エンジニアリングを実現するためのセグメント ルーティング（SR-TE）では、ネットワークを介してトラフィックを誘導する「ポリシー」を使用します。SR-TE ポリシーは、セグメントまたはラベルのセットを含むコンテナです。このセグメントのリストは、ステートフルPCEであるオペレータによってプロビジョニングされます。ヘッドエンドは、SR-TE ポリシ - を介して伝送されるトラフィック フローに、対応する MPLS ラベル スタックを付します。SR-TE ポリシー パスに沿った各通過ノードは、パケットが最終的な宛先に到達するまで、着信トップ ラベルを使用してネクストホップを選択し、ラベルをポップまたはスワップし、ラベル スタックの残りの部分を使用して次のノードにパケットを転送します。

SR-TEポリシーは、タプル（カラー、エンドポイント）によって一意に識別されます。カラーは 32 ビットの数値で表され、エンドポイントは IPv4 です。すべての SR-TE ポリシーにはカラー値があります。同じノード ペア間の各ポリシーには、一意のカラー値が必要です。ポリシーに異なるカラーを選択することで、同じ2つのエンドポイント間で複数のSR-TEポリシーを作成できます。

Cisco Nexus 9000 シリーズ スイッチは、次の 2 種類の SR-TE ポリシーをサポートしています。

- ダイナミック SR-TE ポリシー：SR-TE ポリシー構成またはオンデマンド カラー構成でダイナミック パス プリファレンスを構成すると、パス計算エンジン（PCE）が宛先アドレスへのパスを計算します。PCE でのダイナミック パス計算の結果、ヘッドエンド SR-TE ポリシーに適用されるセグメント/ラベルのリストが生成されます。したがって、トラフィックは、SR-TE ポリシーが保持するセグメントにヒットすることによってネットワークを介してルーティングされます。
- 明示 SR-TE ポリシー：明示パスはラベルのリストであり、明示パスのノードまたはリンクを示します。この機能をイネーブルにするには、**explicit-path** コマンドを使用します。このコマンドにより、明示パスを作成し、パスを指定するためのコンフィギュレーション サブモードを開始できます。

SR-TE ポリシー パス

SR-TE ポリシー パスは、セグメント ID（SID）リストと呼ばれるパスを指定するセグメントのリストです。すべての SR-TE ポリシーは、動的パスまたは明示パスのいずれかである 1 つ以上の候補パスで構成されます。SR-TE ポリシーは 1 つのパスをインスタンス化します。この選択されたパスが優先される有効な候補パスとなります。

動的パス オプションを使用してオンデマンドでカラーを追加し、同じカラーとエンドポイントに対して明示的なパス オプションを使用して明示的なポリシー構成を追加することもできます。この場合、単一のポリシーがヘッドエンドで作成され、設定された優先番号が最も高いパスがトラフィックの転送に使用されます。

SR-TE ポリシー パスの計算には、以下の 2 つの方法が使用されます。

- 動的パス：オンデマンド カラー構成またはポリシー構成でパス プリファレンスを構成するときに動的 PCEP オプションを指定すると、パス計算はパス計算エンジン（PCE）委任されます。
- 明示的なパス：このパスは明示的に指定された SID リストまたは SID リストのセットです。

Cisco NX-OS リリース 10.2(2)F 以降では、SR-TE ポリシーをロックダウンまたはシャットダウンするか、その両方を実行すること、 SR-TE ポリシーまたはオンデマンド カラー テンプレートのシャットダウン設定を行うこと、特定の優先順位を SRTE ポリシーのアクティブ パス オプションに強制すること、または、すべてまたは特定の SRTE ポリシーのパスの再最適化を強制することができます。この機能は、Cisco Nexus 9300-EX、9300-FX、9300-FX2、9300-GX、および N9K-C9332D-GX2B プラットフォーム スイッチでサポートされています。詳細については、SR-TE 手動プレファレンス選択の設定 （187 ページ）を参照してください。

リリース 7.0(3)I7(1) から現在のリリースまでのさまざまな機能をサポートする Cisco Nexus 9000 スイッチの詳細については、Nexus スイッチ プラットフォーム サポート マトリックスを参照してください。

アフィニティおよびディスジョイント制約について

アフィニティ制約：パス計算エンジン（PCE）にアドバタイズされるリンクには、属性を割り当てることができます。SRTE プロセスは、アフィニティ マップとインターフェイス レベルの

構成をホストします。ルーティング プロトコル (IGP) はインターフェイスの更新を登録し、SRTE は IGP にインターフェイスの更新を通知します。IGP tlv は BGP に渡され、外部ピアにアドバタイズされます。アフィニティ制約には 3 つのタイプがあります。

- exclude-any: 指定されたアフィニティ カラーのいずれかを持つリンクをパスが通過してはならないことを指定します。
- include-any: 指定されたアフィニティ カラーのいずれかを持つリンクのみをパスが通過しなければならないことを指定します。したがって、指定されたアフィニティ カラーを持たないリンクを使用してはなりません。
- include-all: 指定されたアフィニティ カラーをすべて持つリンクのみをパスが通過しなければならないことを指定します。したがって、指定されたアフィニティ カラーのすべてを持たないリンクを使用してはなりません。

ディスジョイント制約 - PCE にアドバタイズされる SR-TE ポリシーにディスジョイント制約を割り当てることができます。次に、PCE は、同じアソシエーション グループ ID およびディスジョイントのディスジョイントネス タイプを共有するポリシーに、ディスジョイント パスを提供します。

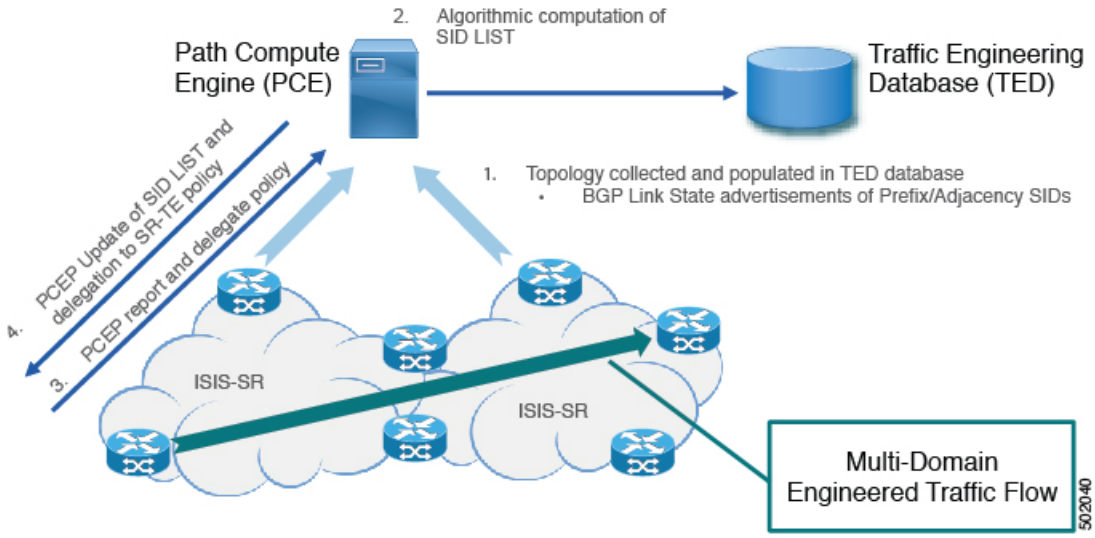
Cisco NX-OS リリース 9.3(1) は、次のディスジョイント パス レベルをサポートします。

- リンク：パスは異なるリンクを通過します（ただし、同じノードを通過する場合があります）。
- ノードのディスジョイントネス：パスは異なるリンクを通過しますが、同じノードを通過する場合があります。

セグメント ルーティング オン デマンド ネクスト ホップ

オン デマンド ネクスト ホップ（ODN）は、BGP ダイナミック SR-TE 機能を活用し、要件に基づいてエンド ツー エンド パスを検索してダウンロードするためのパス計算（PCE）機能を追加します。ODN は定義された BGP ポリシーに基づいて SR-TE 自動トンネルをトリガーします。次の図に示すように、ToR1 と AC1 間のエンド ツー エンド パスは、IGP メトリックに基づいて両端から確立できます。ODN のワークフローは次のようにまとめられます。

図 *10 : ODN* 操作



SR-TE に関する注意事項と制限事項

SR-TE には、次の注意事項と制限事項 があります。

- IPv4 および IPv6 オーバーレイの両方の SR-TE ODN がサポートされています。
- SR-TE ODN は、IS-IS アンダーレイでのみサポートされます。
- 転送では、再帰ネクスト ホップがバインド SID を持つルートに解決される場合、再帰ネクスト ホップを持つルートはサポートされません。
- 転送は、同じルートに対するバインディング ラベルを持つパスとバインディング ラベルのないパスの混合をサポートしていません。
- アフィニティとディスジョイントの制約は、動的な PCEP オプションを持つ SR-TE ポリシーにのみ適用されます。
- XTC は、同じグループ内でディスジョイントになっている 2 つのポリシーのみをサポートします。
- SR-TE アフィニティ インターフェイスを構成する場合、インターフェイス範囲はサポートされません。
- プリファレンスは、動的 PCEP と明示的なセグメント リストの両方を同じプリファレンスに対し一緒に設定することはできません。
- ポリシーごとに動的 PCEP オプションを持つことができるプリファレンスは 1 つだけです。
- 明示的なポリシーについては、同じプリファレンスで ECMP パスを構成する場合、最初のホップ（NHLFE）が両方の ECMP パスで同じであるなら、ULB はスイッチングに 1 つの

パスのみをインストールします。このことは、NHLFE が両方で同じであるため、両方の ECMP パスが同じ SRTE FEC を構築するので発生します。

- Cisco NX-OS リリース 9.3(1) では、アフィニティ設定による非保護モードは PCE（XTC）でサポートされていません。
- Cisco NX-OS リリース 9.3(3) 以降、SR-TE ODN、ポリシー、ポリシー パス、およびアフィニティとディスジョイントの制約は、Cisco Nexus 9364C-GX、Cisco Nexus 9316D-GX、および Cisco Nexus 93600CD-GX スイッチでサポートされています。
- Cisco NX-OS リリース 10.2(2)F 以降、SR-TE ポリシーの新しい show コマンドがいくつか導入されました。また、既存の SR-TE ポリシー コマンドの一部にオートコンプリート機能が提供され、使いやすさが向上しています。この機能は、Cisco Nexus 9300-EX、9300-FX、9300-FX2、9300-GX、および N9K-C9332D-GX2B プラットフォーム スイッチでサポートされています。

（注） リリース 7.0(3)I7(1) から現在のリリースまでのさまざまな機能をサポートする Cisco Nexus 9000 スイッチの詳細については、Nexus スイッチ プラットフォーム サポート マトリックスを参照してください。

SR-TE の設定

トラフィック エンジニアリング用にセグメント ルーティングを設定することができます。

始める前に

mpls セグメント ルーティング機能が有効になっていることを確認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**	グローバル 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**	セグメントルーティング モードを開始します。
ステップ 3	**traffic-engineering**	トラフィック エンジニアリング モードに入ります。
ステップ 4	**encapsulation mpls source ipv4** *tunnel_ip_address*	SR-TE トンネルの送信元アドレスを設定します。
ステップ 5	**pcc**	PCC 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 6	**source-address ipv4** *pcc_source_address*	PCC の送信元アドレスを設定する
ステップ 7	**pce-address ipv4 pce_source_address** *precedence* **num**	PCE の IP アドレスを設定します。最も小さい番号の PCE が優先され、その他はバックアップとして使用されます。
ステップ 8	**on-demand color** *color_num*	オンデマンドモードに入り、カラーを設定します。
ステップ 9	**candidate-paths**	ポリシーの候補パスを指定します。
ステップ 10	**preference** *preference_number*	候補パスの優先順位を指定します。
ステップ 11	**dynamic**	パス オプションを指定します。
ステップ 12	**pcep**	PCE から実行する必要があるパス計算を指定します。

アフィニティ制約の設定

SR-TE ポリシーに対するアフィニティ制約を設定できます。

始める前に

mpls セグメント ルーティング機能が有効になっていることを確認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	**segment-routing**
	例：
	`switch(config)# segment-routing`
	`switch(config-sr)#`
		MPLS セグメント ルーティング機能を有効にします。

ステップ 3	**traffic-engineering**
	例：
	`switch(config-sr)# traffic-engineering`
	`switch(config-sr-te)#`
		トラフィック エンジニアリング モードに入ります。

ステップ 4	**pcc**	PCC 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 5	**source-address ipv4** *pcc_source_address*	PCC の送信元アドレスを設定する
ステップ 6	**pce-address ipv4 pce_source_address** *precedence* **num**	PCE の IP アドレスを設定します。 最も小さい番号の PCE が優先され、その他はバックアップとして使用されます。

ステップ 7	**affinity-map**
	例：
	`switch(config-sr-te)#affinity-map`
	`switch(config-sr-te-affmap)#`
		アフィニティマップ コンフィギュレーション モードを設定します。

ステップ 8	**color** *name* **bit-position** *position*
	例：
	`switch(config-sr-te-affmap)# color red bit-position 2`
	`switch(config-sr-te-affmap)#`
		アフィニティ ビットマップ内の特定のビット位置へのユーザー定義名のマッピングを構成します。

ステップ 9	**interface** *interface-name*
	例：
	`Enter SRTE interface config mode`
	`switch(config-sr-te-if)#interface eth1/1`
	`switch(config-sr-te-if)#`
		インターフェイスの名前を指定します。これは、アフィニティ ビットマップの特定のビットを参照するアフィニティ マッピング名です。

ステップ 10	**affinity**
	例：
	`switch(config-sr-te-if)# affinity`
	`switch(config-sr-te-if-aff)#`
	`switch(config-sr-te-if-aff)# color red`
	`switch(config-sr-te-if-aff)#`
		インターフェイスにアフィニティ カラーを追加します。

ステップ 11	**policy** *name* | **on-demand color** *color_num*
	例：
	`switch(config-sr-te)# on-demand color 211`
	または
	`switch(config-sr-te-color)# policy test_policy`
		ポリシーを設定します。

ステップ 12	**color** *color end-point address* 例： `switch(config-sr-te-pol)#color 200 endpoint 2.2.2.2`	ポリシーのカラーとエンドポイントを設定します。これは、「ポリシー名」設定モードを使用してポリシーを設定するときに必要です。

	コマンドまたはアクション	目的

ステップ 13	**candidate-path**
	例：
	`switch(config-sr-te-color)# candidate-paths`
	`switch(cfg-cndpath)#`
		ポリシーの候補パスを指定します。

ステップ 14	**preference** *preference_number*
	例：
	`switch(cfg-cndpath)# preference 100`
	`switch(cfg-pref)#`
		候補パスの優先順位を指定します。

ステップ 15	**dynamic**
	例：
	`switch(cfg-pref)# dynamic`
	`switch(cfg-dyn)#`
		パス オプションを指定します。

ステップ 16	**pcep**
	例：
	`switch(cfg-dyn)# pcep`
	`switch(cfg-dyn)#`
		ヘッドエンドが PCEP を使用して、それ自体からセグメントルーティングのポリシーのエンドポイントまでのパスを計算するようにPCEに要求することを指定します。

ステップ 17	**constraints**
	例：
	`switch(cfg-dyn)# constraints`
	`switch(cfg-constraints)#`
		候補パス優先制約モードに入ります。

ステップ 18	**affinity**
	例：
	`switch(cfg-constraints)# affinity`
	`switch(cfg-const-aff)#`
		ポリシーのアフィニティ制約を指定します。

ステップ 19	**exclude-any** |**include-all** | **include-any**
	例：
	`switch(cfg-const-aff)# include-any`
	`switch(cfg-aff-inclany)#`
		アフィニティ制約タイプを指定します。次のアフィニティタイプを使用できます。
		• exclude-any - 指定されたアフィニティカラーのいずれかを持つリンクをパスが通過してはならないことを指定します。
		• include-any - 指定されたアフィニティカラーのいずれかを持つリンクのみをパスが通過する必要があることを指定します。

	コマンドまたはアクション	目的
		• include-all - 指定されたアフィニティ カラーをすべて持つリンクのみをパスが通過する必要があることを指定します。

ステップ 20	**color** *color_name*
	例：
	`switch(cfg-aff-inclany)# color blue`
	`switch(cfg-aff-inclany)#`
		アフィニティ カラーの定義を指定します。

ディスジョイント パスの構成

SR-TE ポリシーに対するディスジョイント制約を設定できます。

始める前に

mpls セグメント ルーティング機能が有効になっていることを確認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	**segment-routing**
	例：
	`switch(config)# segment-routing`
	`switch(config-sr)#`
		MPLS セグメント ルーティング機能を有効にします。

ステップ 3	**traffic-engineering**
	例：
	`switch(config-sr)# traffic-engineering`
	`switch(config-sr-te)#`
		トラフィック エンジニアリング モードに入ります。

ステップ 4	**pcc**	PCC モードに入ります。
ステップ 5	**source-address ipv4** *pcc_source_address*	PCC の送信元アドレスを設定する
ステップ 6	**pce-address ipv4 pce_source_address** *precedence* **num**	PCE の IP アドレスを設定します。 最も小さい番号の PCE が優先され、その他はバックアップとして使用されます。

	コマンドまたはアクション	目的

ステップ 7	**policy** *name* | **on-demand color** *color_num*
	例：
	`switch(config-sr-te)# on-demand color 211`
	または
	`switch(config-sr-te-color)# policy test_policy`
		ポリシーを設定します。

ステップ 8	**color** *color end-point address* 例： `switch2(config-sr-te-pol)# color 200 endpoint 2.2.2.2`	ポリシーのカラーとエンドポイントを設定します。これは、「ポリシー名」設定モードを使用してポリシーを設定するときに必要です。

ステップ 9	**candidate-path**
	例：
	`switch(config-sr-te-color)# candidate-paths`
	`switch(cfg-cndpath)#`
		ポリシーの候補パスを指定します

ステップ 10	**preference** *preference_number*
	例：
	`switch(cfg-cndpath)# preference 100`
	`switch(cfg-pref)#`
		候補パスの優先順位を指定します。

ステップ 11	**dynamic**
	例：
	`switch(cfg-pref)# dynamic`
	`switch(cfg-dyn)#`
		パス オプションを指定します。

ステップ 12	**pcep**
	例：
	`switch(cfg-dyn)# pcep`
	`switch(cfg-dyn)#`
		ヘッドエンドが PCEP を使用して、それ自体からセグメントルーティングのポリシーのエンドポイントまでのパスを計算するようにPCEに要求することを指定します。

ステップ 13	**constraints**
	例：
	`switch(cfg-dyn)# constraints`
	`switch(cfg-constraints)#`
		候補パス優先制約モードに入ります。

ステップ 14	**association-group**
	例：
	`switch(cfg-constraints)# association-group`
	`switch(cfg-assoc)#`
		アソシエーション グループ タイプを指定します。

	コマンドまたはアクション	目的

ステップ 15	**disjoint**
	例：
	`switch(cfg-assoc)# disjoint`
	`switch(cfg-disj)#`
		ディスジョイントネスアソシエーション グループに属するパスを指定します。

ステップ 16	**type** | **link** | **node** 例： `switch(config-if)#type link`	ディスジョイントネス グループ タイプを指定します。
ステップ 17	**id** *number* 例： `switch(config-if)#id 1`	アソシエーション グループの識別子を指定します。

SR-TE の設定例

このセクションの例は、アフィニティおよびディスジョイントの設定を示しています。

この例は、ユーザー定義名から管理グループへのマッピングを示しています。

```
segment-routing
 traffic-eng
  affinity-map
   color green bit-position 0
   color blue bit-position 2
   color red bit-position 3
```

この例では、eth1/1 の隣接のアフィニティ リンクの色が赤と緑、eth1/2 の隣接のアフィニティ リンクの色が緑であることを示しています。

```
segment-routing
 traffic-eng
  interface eth1/1
   affinity
    color red
    color green
  !
  interface eth1/2
   affinity
    color green
```

この例は、ポリシーのアフィニティ制約を示しています。

```
segment-routing
  traffic-engineering
    affinity-map
      color blue bit-position 0
      color red  bit-position 1
    on-demand color 10
      candidate-paths
        preference 100
          dynamic
            pcep
          constraints
            affinity
```

```
            [include-any|include-all|exclude-any]
              color <col_name>
              color <col_name>
    policy new_policy
      color 201 endpoint 2.2.2.0
      candidate-paths
        preference 200
          dynamic
            pcep
          constraints
            affinity
              include-all
                color red
```

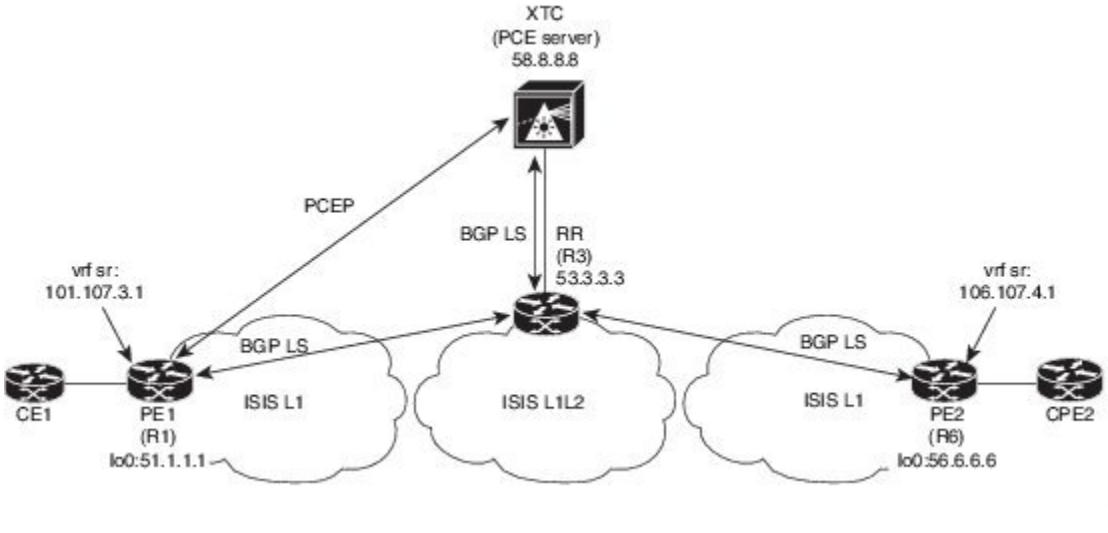
この例は、ポリシーのディスジョイント制約を示しています。

```
segment-routing
 traffic-eng
  on-demand color 99
   candidate-paths
     preference 100
       dynamic
         pcep
       constraints
         association-group
           disjoint
             type link
             id 1
```

SR-TE ODN の設定例 - ユースケース

SR-TE の ODN を設定するには、次のステップを実行します。設定ステップを説明するため、次の図を参考として使用します。

図 *11* : 参照トポロジ



1. PE1 から PE2 への IS-IS ポイントツーポイントセッションですべてのリンクを設定します。また、上記のトポロジーに従ってドメインを設定します。
2. R1、R3、および R6 の IS-IS セッションに対して「リンク状態の配布」を有効にします。

```
router isis 1
  net 31.0000.0000.0000.712a.00
  log-adjacency-changes
  distribute link-state
  address-family ipv4 unicast
    bfd
    segment-routing mpls
    maximum-paths 32
    advertise interface loopback0
```

3. ルータ R1（ヘッドエンド）と R6（テールエンド）に VRF インターフェイスを設定します。

   R1 上の VRF 設定：

```
interface Ethernet1/49.101
encapsulation dot1q 201
  vrf member sr
  ip address 101.10.1.1/24
  no shutdown

vrf context sr
  rd auto
  address-family ipv4 unicast
    route-target import 101:101
    route-target import 101:101 evpn
    route-target export 101:101
    route-target export 101:101 evpn
router bgp 6500
  vrf sr
    bestpath as-path multipath-relax
    address-family ipv4 unicast
      advertise l2vpn evpn
```

4. R6（テールエンド）での BGP コミュニティで VRF プレフィックスをタグ付けします。

```
route-map color1001 permit 10
  set extcommunity color 1001
```

5. R6（テールエンド）およびR1（ヘッドエンド）上のBGPを有効にしてVRF SRプレフィックスのアドバタイズと受信を行い、R6（テールエンド）上のコミュニティ設定とマッチングします。

   R6 < EVPN > R3 < EVPN > R1

   BGP の設定 R6：

```
router bgp 6500
  address-family ipv4 unicast
     allocate-label all
  neighbor 53.3.3.3
    remote-as 6500
    log-neighbor-changes
    update-source loopback0
    address-family l2vpn evpn
      send-community extended
     route-map Color1001 out
      encapsulation mpls
```

   BGP の設定 R1：

```
router bgp 6500
  address-family ipv4 unicast
```

```
     allocate-label all
  neighbor 53.3.3.3
    remote-as 6500
    log-neighbor-changes
    update-source loopback0
    address-family l2vpn evpn
      send-community extended
       encapsulation mpls
```

6. R3 での BGP 構成と、R1、R3.abd での XTC による BGP LS の有効化

   BGP の設定 R3：

```
router bgp 6500
  router-id 2.20.1.2
address-family ipv4 unicast
allocate-label all
address-family l2vpn evpn
retain route-target all
  neighbor 56.6.6.6
    remote-as 6500
    log-neighbor-changes
    update-source loopback0
    address-family l2vpn evpn
      send-community extended
       route-reflector-client
       route-map NH_UNCHANGED out
      encapsulation mpls
  neighbor 51.1.1.1
    remote-as 6500
    log-neighbor-changes
    update-source loopback0
    address-family l2vpn evpn
      send-community extended
      route-reflector-client
      route-map NH_UNCHANGED out
      encapsulation mpls
neighbor 58.8.8.8
    remote-as 6500
    log-neighbor-changes
    update-source loopback0
    address-family link-state

route-map NH_UNCHANGED permit 10
  set ip next-hop unchanged
```

   BGP の設定 R1：

```
router bgp 6500
neighbor 58.8.8.8
              remote-as 6500
               log-neighbor-changes
               update-source loopback0
               address-family link-state
```

   BGP の設定 R6：

```
outer bgp 6500
   neighbor 58.8.8.8
    remote-as 6500
       log-neighbor-changes
       update-source loopback0
       address-family link-state
```

7. R1 で PCE および SR-TE トンネル設定を有効にします。

```
segment-routing
  traffic-engineering
    pcc
      source-address ipv4 51.1.1.1
      pce-address ipv4 58.8.8.8
    on-demand color 1001
      metric-type igp
```

SR-TE 手動プレファレンス選択の設定

このセクションでは、手動プレファレンス選択機能をサポートするために導入された設定および実行コマンドについて説明します。

SR-TE 手動優先順位選択の注意事項と制限事項

次の注意事項と制限事項は、SR-TE 手動優先順位選択機能に適用されます。

- Cisco NX-OS リリース 10.2(2)F 以降、SR-TE の手動優先順位選択機能により、SRTE ポリシーまたはオンデマンド カラー テンプレートの両方でロックダウン、シャットダウン、またはその両方を実行できます（SR-TE ポリシーまたはオンデマンド カラー テンプレートのシャットダウン優先順位）。さらに、この機能により、SR-TE ポリシーに対して特定の優先順位を強制的にアクティブにし、すべてまたは特定の SR-TE ポリシーに対してパスの再最適化を強制することもできます。

  この機能は、Cisco Nexus 9300-EX、9300-FX、9300-FX2、9300-GX、および N9K-C9332D-GX2B プラットフォーム スイッチでサポートされています。

SR-TE 手動設定について：ロックダウンとシャットダウン

Cisco NX-OS リリース 10.2(2)F 以降、必要に応じて次のアクションを実行できます。

- SRTE ポリシーのロックダウン：オンデマンドのカラー テンプレートまたは明示的なポリシーでロックダウンを有効にできます。ロックダウンは、ポリシーのパス設定の自動再最適化を無効にします。ロックダウンされたポリシーに対して新しい優先パスが発生した場合、新しいパスを使用するように自動的に切り替えることはなく、有効になるまで現在のアクティブなパス オプションを使用し続けます。

（注） オンデマンドテンプレートと同じカラーの明示ポリシー構成が存在する場合、ポリシー構成はロックダウンのテンプレート構成よりも優先されます。

例

ポリシーに複数の設定があるシナリオを考えてみましょう。ネットワークの障害により、優先度の高いパスがダウンしたと仮定します。障害は、優先度の高いパスにあるノードの

差し迫った障害である可能性があります。障害を調査して修正するとき、運用チームは問題のあるノードをリロードまたは無効にして、これが発生している間の中断を防ぐ必要がある場合があります。次に、優先度の低いパスをロックダウンし、優先度の高いパスに戻らないようにすることは、使用するのに適したオプションです。

- SRTE ポリシーのシャットダウン：オンデマンドのカラー テンプレートまたは明示ポリシーでシャットダウンを有効にすることができます。ポリシーの状態が管理状態ダウンに変わり、ポリシーに関係するすべてのクライアントにポリシー ダウン通知が送信されます。オンデマンドのカラー構成でシャットダウンを無効にすると、ポリシーのパスの有効性に基づいて、ポリシーの状態がアップまたはダウンに変更されます。

（注） オンデマンドテンプレートと同じ色の明示ポリシー設定が存在する場合、シャットダウンのテンプレート構成よりもポリシー構成が優先されます。

- SRTE ポリシーのシャットダウン設定 – オンデマンドのカラー テンプレート構成または明示ポリシー構成のパス設定で、パス設定をシャットダウンできます。これにより、そのパスプリファレンスが無効になり、プリファレンスが解除されるまで、将来のパスの再最適化が開始されなくなります。パスプリファレンスは、設定でシャットダウンされているかシャットダウンされていないかに基づいて、`show srte policy` の出力に管理状態ダウンまたはアップとして表示されます。

SR-TE 手動設定の構成 - ロックダウン/シャットダウン

SR-TEポリシーまたはオンデマンドカラーテンプレートで、ロックダウン、シャットダウン、またはその両方を構成できます。SR-TE ポリシーまたはオンデマンド カラー テンプレートの下で構成をシャットダウンすることもできます。

始める前に

mpls セグメント ルーティング機能が有効になっていることを確認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**	グローバル 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**	セグメントルーティング モードを開始します。
ステップ 3	**traffic-engineering**	トラフィック エンジニアリング 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 4	**on-demand color***color_num* または **policy** *name*	オンデマンド モードを開始し、カラーを構成します または SR-TE ポリシーを個別に構成します。
ステップ 5	（オプション） [**no**] **lockdown**	オンデマンドのカラー テンプレートまたは明示的なポリシー構成でロックダウンを有効にします。 （注） オンデマンド テンプレートと同じ色の明示的なポリシー構成が存在する場合、ポリシー構成がテンプレート構成よりも優先され、ポリシーがロックダウンされます。
ステップ 6	（オプション） [**no**] **shutdown**	必要に応じて、オンデマンド カラー テンプレートまたは構成済みの SR-TE ポリシーから作成されたポリシーをシャットダウンします。 （注） オンデマンド テンプレートと同じ色の明示的なポリシー構成が存在する場合、ポリシー構成がテンプレート構成よりも優先され、ポリシーがシャットダウンされます。
ステップ 7	**candidate-paths**	ポリシーの候補パスを指定します。
ステップ 8	**preference** *preference_number*	候補パスの優先順位を指定します。
ステップ 9	（オプション） [**no**] **shutdown**	SR-TE ポリシー構成またはオンデマンド カラー テンプレート構成の下でパス プリファレンスをシャットダウンします。

SRTE ポリシーの特定のパス設定を適用する

特定の設定を SRTE ポリシーのアクティブ パス オプションに適用するには、`segment-routing traffic-engineering switch name <policy_name> pref <preference_number>` 実行コマンドを使用します。このコマンドは、有効になるまで設定を使用します。

次のような出力例を示します。

```
NX2# show srte policy Green_White
Policy: 8.8.8.0|801
Name: Green_White
Source: 2.2.2.0
End-point: 8.8.8.0
State: UP
Color: 801
Authorized: Y
Binding-sid Label: 22
Policy-Id: 3
Path type = MPLS Active path option
Path-option Preference:180 ECMP path count: 1
1. PCE Weighted: No
Delegated PCE: 11.11.11.11
Index: 1 Label: 16005
Index: 2 Label: 16008
NX2# segment-routing traffic-engineering switch name Green_White preference 170
NX2(cfg-pref)# show srte policy Green_white detail
Policy: 8.8.8.0|801
Name: Green_White
…..
Path type = MPLS Path options count: 4
Path-option Preference:180 ECMP path count: 1 Admin: UP Forced: No
1. PCE Weighted: No
Delegated PCE: 11.11.11.11
Index: 1 Label: 16005
Index: 2 Label: 16008
Path-option Preference:170 ECMP path count: 1 Admin: UP Forced: Yes Active path option
1. Explicit Weighted: No
Name: Yellow
Index: 1 Label: 16006
Index: 2 Label: 16008
```

この手動で選択した設定を元に戻すには、次のオプションのいずれかを実行します。

- `segment-routing traffic-engineering reoptimize name <policy_name>` コマンドを使用します。詳細については、SRTE ポリシーまたはすべての SRTE ポリシーのパス再最適化の適用 （190 ページ）を参照してください。
- 別の設定に切り替えます
- このポリシーを閉じます
- 選択した設定を閉じます

SRTE ポリシーまたはすべての SRTE ポリシーのパス再最適化の適用

SRTE ポリシーに複数の設定がある場合、ポリシーを再最適化でき、利用可能な最適なパスを選択できます。

特定の SRTE ポリシーのパスの再最適化を適用するには、`segment-routing traffic-engineering reoptimize name <policy_name>` コマンドを使用します。`<policy_name>` は、ポリシー名またはエイリアス名にすることができます。このコマンドは、前のセクションで説明した設定スイッチ コマンドを取り消し、構成されている場合はロックダウンをオーバーライドします。

次のような出力例を示します。

```
NX2# show srte policy Green_White
Policy: 8.8.8.0|801
Name: Green_White
Source: 2.2.2.0
End-point: 8.8.8.0
State: UP
Color: 801
Authorized: Y
Binding-sid Label: 22
Policy-Id: 3
Path type = MPLS Active path option
Path-option Preference:170 ECMP path count: 1
1. Explicit Weighted: Yes Weight: 1
Name: Yellow
Index: 1 Label: 16006
Index: 2 Label: 16008
NX2# segment-routing traffic-engineering reoptimize name Green_White
NX2# show srte policy Green_White
Policy: 8.8.8.0|801
Name: Green_White
Source: 2.2.2.0
End-point: 8.8.8.0
State: UP
Color: 801
Authorized: Y
Binding-sid Label: 22
Policy-Id: 3
Path type = MPLS Active path option
Path-option Preference:180 ECMP path count: 1
1. PCE Weighted: No
Delegated PCE: 11.11.11.11
Index: 1 Label: 16005
Index: 2 Label: 16008
```

すべてのSRTEポリシーのパスの再最適化を強制するには、`segment-routing traffic-engineering reoptimize all` コマンドを使用して、システムに存在するすべての SRTE ポリシーのパスの再最適化を適用します。このコマンドは、前のポイントで説明した設定スイッチ コマンドを取り消し、構成されている場合はロックダウンをオーバーライドします。

SRTE フローベース トラフィック ステアリングの構成

この章では、Cisco Nexus 9000-FX、9000-FX2、9000-FX3、9000-GX、および 9300 プラットフォーム スイッチで SRTE フローベースのトラフィック ステアリングを構成する方法について説明します。

SRTE フローベース トラフィック ステアリング

Cisco NX-OS リリース 10.1(2) のフローベースのトラフィック ステアリング機能は、直接的で柔軟な、ステアリングするトラフィックを選択する代替方法を提供します。この方法では、出力ノードではなく、ヘッドエンド ノードでソース ルーティングを直接構成できます。フローベースのトラフィック ステアリングにより、ユーザーは、宛先アドレス、UDP または TCP ポート、DSCP ビット、その他のプロパティなどの着信パケットのフィールドを一致させることにより、SRTEポリシーに誘導されるパケットを選択できます。一致は、パケットをポリシーに導くように ACL をプログラミングすることによって行われます。

トラフィックを一致させて誘導するために、ポリシーベース ルーティング（PBR）機能が拡張され、SRTE ポリシーをサポートするようになりました。現在の PBR 機能には、RPM、ACL Manager、および AclQoS コンポーネントが含まれます。Cisco NX-OS リリース 10.1(2) 以降、SRTE サポートを追加するために、RPM コンポーネントは SRTE および ULIB とも通信し、URIB との通信が強化されています。

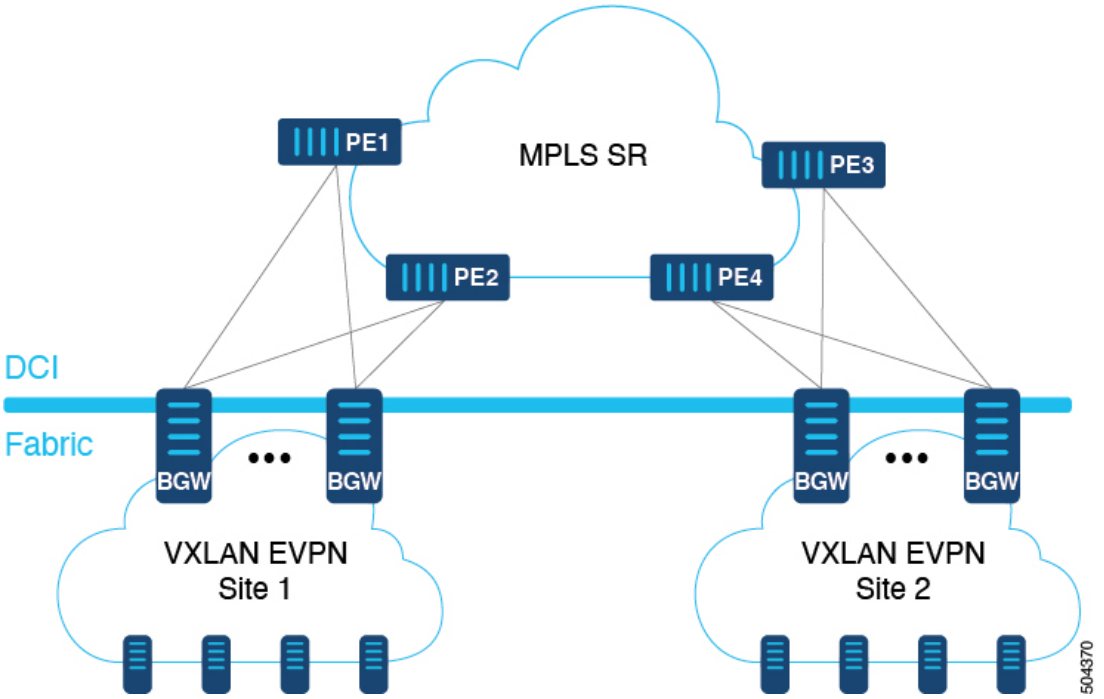
したがって、SRTE のフローベースのトラフィック ステアリング機能には、次のものが含まれます。

- MPLS SR データプレーン
- IPv4 トラフィックのステアリングはデフォルト VRF でサポートされ、IPv4 および IPv6 トラフィックのステアリングはデフォルト以外の VRF でサポートされます
- 5 つのタプル フィールド（送信元アドレス、宛先アドレス、プロトコル、tcp/udp 送信元ポート、tcp/udp 宛先ポート）の組み合わせに基づく ACL によるトラフィックの一致
- 一致したトラフィックを SRTE ポリシーに導く
- IPv4 パケットのパケット内の DSCP/TOS ビットのマッチング。Cisco NX-OS リリース 10.3(1)F 以降では、VXLAN パケットの外部ヘッダーの DSCP/TOS ビットのマッチングもサポートされています。
- IPv6 パケットのパケットのトラフィック クラス フィールドの一致
- 期間の定義に基づく ACL の自動有効化および無効化
- VRF ケースをステアリングするとき、ネクスト ホップを指定せずに SRTE ポリシーへのステアリングをサポートします。
- エニーキャスト エンドポイントを使用したオーバーレイ ECMP
- ACL に一致するパケットは、通常のルートよりも優先されます
- ToS/DSCP およびタイマーベースの ACL に基づくフロー選択
- next-hop-ip は、あるエンドポイントから別のエンドポイントへの SRTE ポリシーへのトラフィックのステアリングに使用されます。

DSCP ベース SRTE トラフィック ステアリング

セグメント ルーティング（SR）コアによって接続される VXLAN マルチサイト構成では、PE は VXLAN サイトの BGW に接続されます。パケットが PE1 で受信されると、パケットは VXLAN カプセル化パケットまたは純粋な IP パケットのいずれかになります。VXLAN パケットの場合、PBR ポリシー ACL フィルタは VXLAN 外部 IP ヘッダー フィールドに適用されます。

図 *12 : MPLS SR* コアを使用した *VXLAN* マルチサイト トポロジ



SRTE のフローベース トラフィック ステアリングの注意事項と制限事項

次の注意事項と制限事項は、SRTE 機能のフローベース トラフィック ステアリングに適用されます。

- Cisco NX-OS リリース 10.1(2) 以降、SRTE のフローベースのトラフィック ステアリング機能は、Cisco Nexus 9000-FX、9000-FX2、9000-FX3、9000-GX、および 9300 プラットフォーム スイッチでサポートされます。
- SRTE ポリシーが VRF のインターフェイスに割り当てられたルート マップに適用されるとき（L3VPN/L3EVPN トラフィックを誘導するため）、set statement のネクスト ホップが BGP プレフィックスに解決され、その BGP プレフィックスがすでに SRTE を使用してトラフィックを誘導し、ルートマップはトラフィックを誘導しません。
- アンダーレイ ECMP は、ポリシー内のアクティブな各 SRTE パス（ECMP メンバー）のラベル スタックが同じ場合にのみサポートされます。9000-GX プラットフォームには、この制限はありません。
- ルートマップ トラッキング機能はサポートされていません。
- SRTE ポリシーを操作する場合、1 つのルート マップ シーケンス エントリに複数の**ネクスト ホップを設定**することはサポートされていません。

- SRTE ポリシーが VRF のインターフェイスに割り当てられたルート マップに適用される場合（L3VPN/L3EVPN トラフィックを誘導するため）、set ステートメントのネクスト ホップが RIB で複数のネクスト ホップを有する BGP ルート（オーバーレイ ルート）に対して解決される場合、 トラフィックはルートの最初のネクスト ホップにのみ誘導され、すべてのネクスト ホップで ECMP は行われません。
- SRTE ポリシー名がルート マップ セット ステートメントで使用されている場合、カラーとエンドポイントではなく、デフォルトの VRF ステアリングにのみ使用できます。そうでない場合は、明示的に定義されている SRTE パスを選択する必要があります。具体的には、これは、ラベルの代わりにポリシーエンドポイント キーワードを含むセグメントリストを使用するように定義された SRTE ポリシーを選択するためには使用できません。
- **set ip next-hop <>** で指定されたネクスト ホップ IP に適用される次のキーワードは、SRTE ポリシーにステアリングするときのルート マップではサポートされません。
  - verify-availability
  - drop-on-fail
  - force-order
  - load-share
- 必要な機能（セグメンティング ルーティング、l3 evpn または l3vpn）がデバイスで有効になっていない場合でも、srte-policy を使用したルート マップをインターフェイスに適用できます。ただし、srte-policy を使用した set-actions は抑制されます。つまり、これらのフローに対してデフォルト ルーティングが実行されます。
- ルート マップには、srte-policy ありおよび srte-policy なしの set コマンドを含めることができます。
- srte-policy 情報のない set-command の場合、ステアリングは next-hop-ip への到達可能性が MPLS ラベルを必要としない場合にのみ実行されます。
- ルート マップがデフォルト以外の VRF のインターフェイスに関連付けられており、そのルート マップにネクスト ホップ IP アドレス **N** と SRTE ポリシーを指定するシーケンスが含まれている場合、そのルートマップ上の他のすべてのシーケンスと、同じネクストホップ IP アドレスを使用する同じ VRF に関連付けられたその他すべてのルート マップにも SRTE ポリシーが必要です。同じネクスト ホップ IP と異なる SRTE ポリシーを使用して、別のルート マップまたはルート マップ シーケンスを同じ VRF に関連付けることはできません。
- 同様に、ルート マップがデフォルト以外の VRF のインターフェイスに関連付けられていて、そのルート マップが SRTE ポリシーを指定していないが、ネクスト ホップ IP アドレス **N** を指定している場合、同じネクスト ホップ IP アドレス N を使用し、SRTE ポリシーを指定する、そのルートマップまたは別のルートマップ内の別のシーケンスは適用されません。
- SRTE フローベースのトラフィック ステアリングは、VXLAN または EoMPLS PBR と同時に使用することはできません。

- SRTE 入力ノードのポリシー ベースのルーティング トラフィックでは、SR ラベル統計はサポートされていません。ただし、ACL リダイレクト統計はサポートされています。
- デフォルト VRF の IPv6 トラフィックは、SRTE ポリシーに誘導できません。MPLS SR アンダーレイは、IPv4 でのみサポートされます。ただし、IPv6 SR アンダーレイが必要な場合は、代わりに SRv6 を使用します。
- 9000-FX、9000-FX2、9000-FX3、および 9300 プラットフォーム ハードウェアは、ECMP メンバーごとに一意のアンダーレイ ラベル スタックをプッシュできず、これらのプラットフォームのアンダーレイ ECMP に影響します。つまり、セグメントリストの最初のホップが異なる SRTE ポリシーに複数のアクティブ セグメントリストがある場合（1 つの設定が複数のセグメントリストで構成されている場合）、そのような構成はサポートされません。このような場合、回避策として、エニーキャスト SID を構成して、すべての ECMP メンバーでラベル スタックが同じになるようにします。
- モジュラ プラットフォームは、Cisco NX-OS リリース 10.1(2) ではサポートされていません。
- Cisco NX-OS リリース 10.2(2)F 以降、SRTE のフローベースのトラフィック ステアリング機能は、Cisco N9K-C9332D-GX2B プラットフォーム スイッチでサポートされます。
- Cisco NX-OS リリース 10.3(1)F 以降、DSCP ベースの SR-TE フロー ステアリング機能により、IP ヘッダーの DSCP フィールドを使用して照合され、SRTE パスに誘導される VXLAN パケットのソースルーティングが可能になります。以下はこの機能の注意事項と制限事項です。
  - この機能は、Cisco Nexus 9300-FX2、9300-FX3、9300-GX、9300-GX2 TOR スイッチでのみサポートされます。
  - VXLAN パケットが終了していない場合、ACL フィルタは VXLAN パケットの外部 IP ヘッダ フィールド（IPv4）に適用されます。
- Cisco NX-OS リリース 10.3(2)F 以降、SRTE 向けフローベース トラフィック ステアリング機能は、Cisco Nexus 9700-FX および 9700-GX ライン カードでサポートされます。以下はこの機能の注意事項と制限事項です。
  - Cisco Nexus 9508 プラットフォーム スイッチが VXLAN EVPN から MPLS SR L3VPN へのハンドオフ モードで、MPLS カプセル化パケットが L2 ポートで転送される場合、dot1q ヘッダーは追加されません。
  - Cisco Nexus 9500 プラットフォーム スイッチが EVPN から MPLS SR L3VPN へのハンドオフ モードとして設定されている場合、SVI/サブインターフェイスは、コアに面したアップリンク (MPLS または VXLAN) ではサポートされません。
  - DSCP から MPLS EXP へのプロモーションは、DCI モードの FX TOR/ラインカードでは機能しません。MPLS EXP への内部 DSCP 値のコピーは、このハンドオフ モードの FX TOR/ライン カードでは機能しません。MPLS EXP は 0x7 に設定されます。

- Cisco NX-OS リリース 10.3(2)F 以降、DSCP ベースの SRTE フロー ステアリング機能は、Cisco Nexus 9300-FX プラットフォームおよび Cisco Nexus 9700-FX と 9700-GX ライン カードでサポートされます。以下はこの機能の注意事項と制限事項です。
  - Cisco Nexus 9500 プラットフォーム スイッチが VXLAN EVPN から MPLS SR L3VPN へのハンドオフモードで、MPLS カプセル化パケットが L2 ポートで転送される場合、dot1q ヘッダーは追加されません。
  - Cisco Nexus 9500 プラットフォーム スイッチが EVPN から MPLS SR L3VPN へのハンドオフモードとして設定されている場合、SVI/サブインターフェイスは、コアに面したアップリンク (MPLS または VXLAN) ではサポートされません。
  - DSCP から MPLS EXP へのプロモーションは、DCI モードの FX TOR/ラインカードでは機能しません。MPLS EXP への内部 DSCP 値のコピーは、このハンドオフモードの FX TOR/ライン カードでは機能しません。MPLS EXP は 0x7 に設定されます。

構成プロセス：SRTE フローベース トラフィック ステアリング

SRTE フローベースのトラフィック ステアリング機能の構成プロセスは次のとおりです。

1. 特に IP アクセス リストの基準に一致する IP アクセス リストを構成します。

   詳細については、『*Cisco Nexus Series NX-OS* セキュリティ構成ガイド』の「*IP ACL* の構成」章を参照してください。

2. SRTE ポリシーを定義します。

   SRTE の設定の詳細については、『*Cisco Nexus 9000* シリーズ *NX-OS* ラベル スイッチ構成ガイド』の「トラフィック エンジニアリング用セグメント ルーティングの構成」の章を参照してください。

3. 一致（ステップ 1 で設定した IP アクセス リスト）とアクションをバインドするルートマップを構成します。一致は、パケットで一致するフィールドを参照し、アクションは、どの SRTE ポリシーを誘導するか、および使用する VPN ラベルを参照します（存在する場合）。

ToS/DSCP およびタイマーベース ACL に基づいたフロー選択の構成

SRTE フローベースのトラフィック ステアリング機能では、フロー選択は ToS/DSCP およびタイマー ベースの ACL に基づいています。

デフォルトおよびデフォルト以外の VRF のルート マップを、さまざまな基準によって選択されたポリシーに構成して正しく動作させるには、次の構成手順を実行します。

始める前に

MPLS セグメント ルーティング トラフィック エンジニアリングおよび PBR 機能が有効になっていることを確認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	[**ip** | **ipv6**] **access-list** *acl_name*
	例：
	`switch(config)# ip access-list L4_PORT`
	`switch(config)#`
		名前を使用して IP または IPv6 アクセス リストを定義し、 IP または IPv6 アクセスリストコンフィギュレーション モードを開始します。

ステップ 3	**10 permit ip** *ip_address* **any**
	例：
	`switch(config)# 10 permit ip any 5.5.0.0/16`
	`switch(config)#`
		スイッチで構成された IP または IPv6 アクセス リストを表示します。

ステップ 4	**20 permit tcp** *tcp_address* [**any**]
	例：
	`switch(config)# 20 permit tcp any 5.5.0.0/16`
	`switch(config)#`
		IPv6 アクセス リストに TCP 許可条件を設定します。
		（注） **any** キーワードは、IPv6 にのみ使用されます。

ステップ 5	[**ip** | **ipv6**] **access-list** *dscp_name*
	例：
	`switch(config)# ip access-list dscp`
	`switch(config)#`
		名前を使用して IP または IPv6 アクセス リストの DSCP 定義し、 IP または IPv6アクセス リスト コンフィギュレーション モードを開始します。

ステップ 6	**10 permit tcp any** *tcp_address* **dscp** <*dscp value*>
	例：
	`switch(config)# 10 permit tcp any 5.5.0.0/16 dscp af11`
	`switch(config)#`
		IP または IPv6 アクセス リストの DSCP 値を設定します。
		（注） **any** キーワードは、IPv6 にのみ使用されます。

ステップ 7	[**ip** | **ipv6**] **access-list** *acl_name*
	例：
	`switch(config)# ip access-list acll`
	`switch(config)#`
		名前を使用して IP または IPv6 アクセス リストを定義し、 IP または IPv6 アクセスリストコンフィギュレーション モードを開始します。

ステップ 8	**10 permit tcp any** *tcp_address* **acl** *acl_name*
	例：
	`switch(config)# 10 permit tcp any 5.5.0.0/16 eq www dscp af11`
	`switch(config)#`
		IPv6 アクセス リストに TCP 許可条件を設定します。
		（注） **any** キーワードは、IPv6 にのみ使用されます。

	コマンドまたはアクション	目的

ステップ 9	[**ip** | **ipv6**] **access-list** *acl_name*
	例：
	`switch(config)# ip access-list acl1`
	`switch(config)#`
		名前を使用して IP または IPv6 アクセス リストを定義し、 IP または IPv6 アクセス リスト コンフィギュレーション モードを開始します。

ステップ 10	**10 permit tcp any** *any time - range tl*
	例：
	`switch(config-acl)# 10 permit tcp any 5.5.0.0/16 eq www dscp af11`
	`switch(config)#`
		IP または IPv6 アクセス リストの TCP の時間範囲を定義する時間範囲値を設定します。
		（注） **any** キーワードは、IPv6 にのみ使用されます。

ステップ 11	**time-range** *name*
	例：
	`switch(config-acl)# time-range t1`
	`switch(config)#`
		名前を使用して、IP または IPv6 アクセス リストの時間範囲を定義します。

ステップ 12	**F2(config-time-range)#**
	WOLF2(config-time-range)#
	例：
	`switch(config-time-range)# 10 absolute start 20:06:56 8 february 2021`
	`end 20:10:56 8 february 2021`
		構成の時間範囲を定義します。

フローベース トラフィック ステアリングのデフォルトおよび非デフォルト VRF でのルート マップの構成

次のセクションでは、SRTE フローベースのトラフィック ステアリング機能のデフォルトおよび非デフォルト VRF でルート マップを構成する方法を示します。

カラーおよびエンドポイントによって選択されているポリシーへのデフォルト VRF のルート マップの構成

デフォルト VRF のトラフィックを、色とエンドポイントで選択されたポリシーに導くルート マップを構成するには、次の手順を実行します。

始める前に

MPLS セグメント ルーティング トラフィック エンジニアリングおよび PBR 機能が有効になっていることを確認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	**route-map FLOW1** *seq_num* 例：	ルート マップに FLOW1 という名前を付けます。

	コマンドまたはアクション	目的
	`switch(config)# route-map FLOW1 seq 10` `switch(config-route-map)#`	

ステップ 2	**match** [**ip** | **ipv6**] **address** *acl_name*
	例：
	`switch(config-route-map)# match ip address L4_PORT`
	`switch(config-route-map)#`
		フィールドを説明する ACL を追加することにより、ルート マップが一致する必要のあるフィールドを指定します。

ステップ 3	**set srte-policy color** *num* **endpoint** *ip address*
	例：
	`switch(config-route-map)# set srte-policy color 121 endpoint 10.0.0.1`
	`switch(config-route-map)#`
		SRTE ポリシー カラーとポリシーのエンド ポイントを構成します。
		（注） IPv4 アドレスのみをエンド ポイントにできます。

ステップ 4	**interface** *interface-type/slot/port*
	例：
	`switch(config-route-map)# interface ethernet 1/1`
	`switch(config-route-map-if)#`
		インターフェイス コンフィギュレーション モードを開始します。

ステップ 5	[**ip** | **ipv6**] **policy route-map FLOW1**
	例：
	`switch(config-route-map-if)# ip policy route-map FLOW1`
	`switch(config-route-map-if)#`
		IP または IPv6 ポリシーベース ルーティングをインターフェイスに割り当てます。これにより、インターフェイスに入力するすべてのトラフィックのルート マップが適用されます。

名前で選択されたポリシーへのデフォルト VRF のルート マップ構成例

デフォルト VRF のトラフィックを名前で選択されたポリシーに導くルート マップを構成するには、次の手順を実行します。

始める前に

MPLS セグメント ルーティング トラフィック エンジニアリングおよび PBR 機能が有効になっていることを確認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	**route-map FLOW1** *seq_num*
	例：
	`switch(config)# route-map FLOW1 seq 10`
	`switch(config-route-map)#`
		ルート マップに FLOW1 という名前を付けます。

	コマンドまたはアクション	目的

ステップ 2	**match** [**ip** | **ipv6**] **address** *acl_name*
	例：
	`switch(config-route-map)# match ip address L4_PORT`
	`switch(config-route-map)#`
		フィールドを説明する ACL を追加することにより、ルート マップが一致する必要のあるフィールドを指定します。

ステップ 3	**set srte-policy name** *policy-name*
	例：
	`switch(config-route-map)# set srte-policy name policy1`
	`switch(config-route-map)#`
		SRTE ポリシー名を構成します。

ステップ 4	**interface** *interface-type/slot/port*
	例：
	`switch(config-route-map)# interface ethernet 1/1`
	`switch(config-route-map-if)#`
		インターフェイス コンフィギュレーション モードを開始します。

ステップ 5	[**ip** | **ipv6**] **policy route-map FLOW1**
	例：
	`switch(config-route-map-if)# ip policy route-map FLOW1`
	`switch(config-route-map-if)#`
		IP または IPv6 ポリシーベース ルーティングをインターフェイスに割り当てます。これにより、インターフェイスに入力するすべてのトラフィックのルート マップが適用されます。

ネクストホップ、カラー、およびエンドポイントで選択されたポリシーへのデフォルト以外の VRF のルート マップ構成

デフォルト以外の VRF のトラフィックを、カラーとエンドポイントで選択されたポリシーに導くルートマップを構成するには、次の手順を実行します。この手順では、正しいMPLS VPNラベルがトラフィックに適用されるようにネクストホップを指定します。

始める前に

MPLS セグメント ルーティング トラフィック エンジニアリングおよびPBR 機能が有効になっていることを確認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	**route-map FLOW1** *seq_num*
	例：
	`switch(config)# route-map FLOW1 seq 10`
	`switch(config-route-map)#`
		ルート マップに FLOW1 という名前を付けます。

	コマンドまたはアクション	目的

ステップ 2	**match** [**ip** | **ipv6**] **address** *acl_name*
	例：
	`switch(config-route-map)# match ip address L4_PORT`
	`switch(config-route-map)#`
		フィールドを説明する ACL を追加することにより、ルート マップが一致する必要のあるフィールドを指定します。

ステップ 3	**set** [**ip** | **ipv6**] **next-hop** *destination-ip-next-hop* **srte-policy color** *num* **endpoint** *ip address*
	例：
	`switch(config-route-map)# set ip next-hop 5.5.5.5 srte-policy color 121 endpoint 10.0.0.1`
	`switch(config-route-map)#`
		srte-policy（カラーおよびエンドポイント）を介して、構成されたネクストホップにパケットをリダイレクトします。

ステップ 4	**exit**
	例：
	`switch(config-route-map)# exit`
	`switch(config)#`
		ルートマップ構成モードを終了し、グローバル構成モードに戻ります。

ステップ 5	**interface** *interface-type/slot/port*
	例：
	`switch(config)# interface ethernet 1/1`
	`switch(config-if)#`
		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 6	**vrf member** *vrf-name*
	例：
	`switch(config-if)# vrf member vrf1`
	`switch(config-if)#`
		このインターフェイスを VRF に追加します。

ステップ 7	[**ip** | **ipv6**] **policy route-map FLOW1**
	例：
	`switch(config-if)# ip policy route-map FLOW1`
	`switch(config-if)#`
		IP または IPv6 ポリシーベース ルーティングをインターフェイスに割り当てます。これにより、インターフェイスに入力するすべてのトラフィックのルートマップが適用されます。

ステップ 8	[**no**] **shutdown**
	例：
	`switch(config-if)# no shutdown`
	`switch(config-if)#`
		インターフェイスをディセーブルにします。

デフォルト以外の VRF のルート マップをネクストホップおよびカラー別に選択されたポリシーに構成する

次の手順を実行し、デフォルト VRF のトラフィックを色とエンドポイントで選択されたポリシーに誘導するルートマップを構成しますが、エンドポイントは明示的に構成されていません。 ネクストホップが指定されているため、正しい MPLS VPN ラベルがトラフィックに適用され、正しい SRTE エンドポイントがネクストホップに一致するルートから取得されます。

始める前に

MPLS セグメント ルーティング トラフィック エンジニアリングおよび PBR 機能が有効になっていることを確認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	**route-map FLOW1** *seq_num*
	例：
	`switch(config)# route-map FLOW1 seq 10`
	`switch(config-route-map)#`
		ルート マップに FLOW1 という名前を付けます。

ステップ 2	**match** [**ip** | **ipv6**] **address** *acl_name*
	例：
	`switch(config-route-map)# match ip address L4_PORT`
	`switch(config-route-map)#`
		フィールドを説明する ACL を追加することにより、ルート マップが一致する必要のあるフィールドを指定します。

ステップ 3	**set** [**ip** | **ipv6**] **next-hop** *destination-ip-next-hop* **srte-policy color** *num*
	例：
	`switch(config-route-map)# set ip next-hop 5.5.5.5 srte-policy color 121`
	`switch(config-route-map)#`
		srte-policy（カラー）を介して、構成されたネクストホップにパケットをリダイレクトします。

ステップ 4	**exit**
	例：
	`switch(config-route-map)# exit`
	`switch(config)#`
		ルートマップ構成モードを終了し、グローバル構成モードに戻ります。

ステップ 5	**interface** *interface-type/slot/port*
	例：
	`switch(config)# interface ethernet 1/1`
	`switch(config-if)#`
		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 6	**vrf member** *vrf-name*
	例：
	`switch(config-if)# vrf member vrf1`
	`switch(config-if)#`
		このインターフェイスを VRF に追加します。

ステップ 7	[**ip** | **ipv6**] **policy route-map FLOW1**
	例：
	`switch(config-if)# ip policy route-map FLOW1`
	`switch(config-if-route-map)#`
		IP または IPv6 ポリシーベース ルーティングをインターフェイスに割り当てます。これにより、インターフェイスに入力するすべてのトラフィックのルートマップが適用されます。

	コマンドまたはアクション	目的

ステップ 8	[**no**] **shutdown**
	例：
	`switch(config-if-route-map)# no shutdown`
	`switch(config-if-route-map)#`
		インターフェイスをディセーブルにします。

デフォルト以外の VRF のルート マップをネクストホップおよび名前別に選択されたポリシーに構成する

次の手順を実行して、デフォルト以外の VRF のトラフィックを名前別に選択されたポリシーに誘導するルート マップを構成します。ネクストホップは、正しい MPLS VPN ラベルがトラフィックに課されるように指定されます

始める前に

MPLS セグメント ルーティング トラフィック エンジニアリングおよび PBR 機能が有効になっていることを確認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	**route-map FLOW1** *seq_num*
	例：
	`switch(config)# route-map FLOW1 seq 10`
	`switch(config-route-map)#`
		ルート マップに FLOW1 という名前を付けます。

ステップ 2	**match** [**ip** | **ipv6**] **address** *acl_name*
	例：
	`switch(config-route-map)# match ip address L4_PORT`
	`switch(config-route-map)#`
		フィールドを説明する ACL を追加することにより、ルート マップが一致する必要のあるフィールドを指定します。

ステップ 3	**set** [**ip** | **ipv6**] **next-hop** *destination-ip-next-hop* **srte-policy** *name*
	例：
	`switch(config-route-map)# set ip next-hop 5.5.5.5 srte-policy policy1`
	`switch(config-route-map)#`
		srte-policy（名前）を介して、構成されたネクストホップにパケットをリダイレクトします。

ステップ 4	**exit**
	例：
	`switch(config-route-map)# exit`
	`switch(config)#`
		ルートマップ構成モードを終了し、グローバル構成モードに戻ります。

ステップ 5	**interface** *interface-type/slot/port* 例：	インターフェイス コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
	`switch(config)# interface ethernet 1/1` `switch(config-if)#`	

ステップ 6	**vrf member** *vrf-name*
	例：
	`switch(config-if)# vrf member vrf1`
	`switch(config-if)#`
		このインターフェイスを VRF に追加します。

ステップ 7	[**ip** | **ipv6**] **policy route-map FLOW1**
	例：
	`switch(config-if)# ip policy route-map FLOW1`
	`switch(config-if)#`
		IP または IPv6 ポリシーベース ルーティングをインターフェイスに割り当てます。これにより、インターフェイスに入力するすべてのトラフィックのルートマップが適用されます。

ステップ 8	[**no**] **shutdown**
	例：
	`switch(config-if)# no shutdown`
	`switch(config-if)#`
		インターフェイスをディセーブルにします。

カラーとエンドポイントで選択されたポリシーへのデフォルト以外の VRF のルート マップ構成例

デフォルト以外の VRF のトラフィックを、カラーとエンドポイントで選択されたポリシーに導くルートマップを構成するには、次の手順を実行します。この手順では、指定するネクストホップは必要ありません。VPN ラベルは、ローカル スイッチで VRF に割り当てられたラベルを検索することによって取得されます。 これは、すべてのスイッチの VRF の BGP 割り当てインデックス構成を使用して、すべてのスイッチの VRF に同じラベルが割り当てられている場合にのみ構成可能です。

始める前に

MPLS セグメント ルーティング トラフィック エンジニアリングおよび PBR 機能が有効になっていることを確認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	**route-map FLOW1** *seq_num*
	例：
	`switch(config)# route-map FLOW1 seq 10`
	`switch(config-route-map)#`
		ルート マップに FLOW1 という名前を付けます。

	コマンドまたはアクション	目的

ステップ 2	**match** [**ip** | **ipv6**] **address** *acl_name*
	例：
	`switch(config-route-map)# match ip address L4_PORT`
	`switch(config-route-map)#`
		フィールドを説明する ACL を追加することにより、ルートマップが一致する必要のあるフィールドを指定します。

ステップ 3	**set srte-policy color** *num* **endpoint** *ip address*
	例：
	`switch(config-route-map)# set srte-policy color 121 endpoint 10.0.0.1`
	`switch(config-route-map)#`
		SRTE ポリシー カラーとポリシーのエンド ポイントを構成します。
		（注） IPv4 アドレスのみをエンドポイントにできます。

ステップ 4	**interface** *interface-type/slot/port*
	例：
	`switch(config-route-map)# interface ethernet 1/1`
	`switch(config-route-map-if)#`
		インターフェイス コンフィギュレーション モードを開始します。

ステップ 5	**vrf member** *vrf-name*
	例：
	`switch(config-route-map-if)# vrf member vrf1`
	`switch(config-route-map-if)#`
		このインターフェイスを VRF に追加します。

ステップ 6	[**ip** | **ipv6**] **policy route-map FLOW1**
	例：
	`switch(config-route-map-if)# ip policy route-map FLOW1`
	`switch(config-route-map-if)#`
		IP または IPv6 ポリシーベース ルーティングをインターフェイスに割り当てます。これにより、インターフェイスに入力するすべてのトラフィックのルート マップが適用されます。

ステップ 7	[**no**] **shutdown**
	例：
	`switch(config-route-map-if)# no shutdown`
	`switch(config-route-map-if)#`
		インターフェイスをディセーブルにします。

ステップ 8	**exit**
	例：
	`switch(config-route-map)# exit`
	`switch(config)#`
		ルートマップ構成モードを終了し、グローバル構成モードに戻ります。

ステップ 9	**feature bgp**
	例：
	`switch(config)# feature bgp`
	`switch(config)#`
		BGP 機能を開始します。

	コマンドまたはアクション	目的

ステップ 10	**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。

ステップ 11	**vrf** *vrf-name*
	例：
	`switch(config-router)# vrf vrf1`
	`switch(config-router-vrf)#`
		BGP プロセスを VRF に関連付けます。

ステップ 12	**allocate-index** *index*
	例：
	`switch(config-router-vrf)#`
	`allocate-index 10`
		VRF にインデックスを割り当てます。これにより、VRF にスタティック MPLS ローカル VPN ラベルを割り当てるように BGP に指示されます。VRF に割り当てられた MPLS VPN ラベルは、指定された値から取得されます。インデックスは、MPLS ラベル値の特別な範囲へのオフセットとして使用されます。 指定されたインデックス値の場合、同じローカル ラベルが常に許可されます。

名前で選択されたポリシーへのデフォルト以外のルート マップ構成例

次の手順を実行して、デフォルト以外の VRF のトラフィックを名前別に選択されたポリシーに誘導するルートマップを構成します。この手順では、指定するネクストホップは必要ありません。VPN ラベルは、ローカル スイッチで VRF に割り当てられたラベルを検索することによって取得されます。 これは、すべてのスイッチの VRF の BGP 割り当てインデックス構成を使用して、すべてのスイッチの VRF に同じラベルが割り当てられている場合にのみ構成可能です。

始める前に

MPLS セグメント ルーティング トラフィック エンジニアリングおよび PBR 機能が有効になっていることを確認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	**route-map FLOW1** *seq_num*
	例：
	`switch(config)# route-map FLOW1 seq 10`
	`switch(config-route-map)#`
		ルート マップに FLOW1 という名前を付けます。

	コマンドまたはアクション	目的

ステップ 2	**match** [**ip** | **ipv6**] **address** *acl_name*
	例：
	`switch(config-route-map)# match ip address L4_PORT`
	`switch(config-route-map)#`
		フィールドを説明する ACL を追加することにより、ルートマップが一致する必要のあるフィールドを指定します。

ステップ 3	**set srte-policy** *name*
	例：
	`switch(config-route-map)# set srte-policy policy1`
	`switch(config-route-map)#`
		SRTE ポリシー名を構成します。

ステップ 4	**interface** *interface-type/slot/port*
	例：
	`switch(config-route-map)# interface ethernet 1/1`
	`switch(config-route-map-if)#`
		インターフェイス設定モードを開始します。

ステップ 5	**vrf member** *vrf-name*
	例：
	`switch(config-route-map-if)# vrf member vrf1`
	`switch(config-route-map-if)#`
		このインターフェイスを VRF に追加します。

ステップ 6	[**ip** | **ipv6**] **policy route-map FLOW1**
	例：
	`switch(config-route-map-if)# ip policy route-map FLOW1`
	`switch(config-route-map-if)#`
		IP または IPv6 ポリシーベース ルーティングをインターフェイスに割り当てます。これにより、インターフェイスに入力するすべてのトラフィックのルート マップが適用されます。

ステップ 7	[**no**] **shutdown**
	例：
	`switch(config-route-map-if)# no shutdown`
	`switch(config-route-map-if)#`
		インターフェイスをディセーブルにします。

ステップ 8	**exit**
	例：
	`switch(config-route-map)# exit`
	`switch(config)#`
		ルートマップ構成モードを終了し、グローバル構成モードに戻ります。

ステップ 9	**feature bgp**
	例：
	`switch(config)# feature bgp`
	`switch(config)#`
		BGP 機能を開始します。

	コマンドまたはアクション	目的

ステップ 10	**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。

ステップ 11	**vrf** *vrf-name*
	例：
	`switch(config-router)# vrf vrf1`
	`switch(config-router-vrf)#`
		BGP プロセスを VRF に関連付けます。

ステップ 12	**allocate-index** *index*
	例：
	`switch(config-router-vrf)#`
	`allocate-index 10`
		VRF にインデックスを割り当てます。これにより、VRF にスタティック MPLS ローカル VPN ラベルを割り当てるように BGP に指示されます。VRF に割り当てられた MPLS VPN ラベルは、指定された値から取得されます。インデックスは、MPLS ラベル値の特別な範囲へのオフセットとして使用されます。 指定されたインデックス値の場合、同じローカル ラベルが常に許可されます。

SRTE フローベース トラフィック ステアリングの構成例

このセクションには、SRTE フローベースのトラフィック ステアリングを構成するための次の例が含まれています。

ToS/DSCP および時間ベース ACL に基づくフロー選択の構成例

```
switch# configure terminal
switch(config)# ip access-list L4_PORT
switch(config)# 10 permit ip any 5.5.0.0/16
switch(config)# 20 permit tcp any 5.5.0.0/16
switch(config)# ip access-list dscp
switch(config)# 10 permit tcp any 5.5.0.0/16 dscp af11
switch(config)# ip access-list acll
switch(config)# 10 permit tcp any 5.5.0.0/16 eq www dscp af11
switch(config)# ip access-list acl1
switch(config-acl)# 10 permit tcp any 5.5.0.0/16 eq www dscp af11
switch(config-acl)# time-range t1
start 20:06:56 8 february 2021 end 20:10:56 8 february 2021
```

カラーおよびエンドポイントで選択されたポリシーへのデフォルト VRF のルートマップ構成例

```
switch(config)# route-map FLOW1 seq 10
switch(config-route-map)# match ip address L4_PORT
switch(config-route-map)# set srte-policy color 121 endpoint 10.0.0.1
```

```
switch(config-route-map)# interface ethernet 1/1
switch(config-route-map-if)# ip policy route-map FLOW1
```

名前別に選択されたポリシーへのデフォルトの VRF でのルートマッピング構成例

```
switch(config)# route-map FLOW1 seq 10
switch(config-route-map)# match ip address L4_PORT
switch(config-route-map)# set srte-policy name policy1
switch(config-route-map)# interface ethernet 1/1
switch(config-route-map-if)# ip policy route-map FLOW1
```

ネクストホップ、カラー、エンドポイントで選択されたポリシーへのデフォルト以外の VRF のルートマップ構成例

```
switch(config)# route-map FLOW1 seq 10
switch(config-route-map)# match ip address L4_PORT
switch(config-route-map)# set ip next-hop 5.5.5.5 srte-policy color 121 endpoint 10.0.0.1
switch(config-route-map)# interface ethernet 1/1
switch(config-route-map)# vrf member vrf1
switch(config-route-map-if)# ip policy route-map FLOW1
```

ネクストホップおよびカラーで選択されたポリシーへのデフォルト以外の VRF のルートマップの構成例

```
switch(config)# route-map FLOW1 seq 10
switch(config-route-map)# match ip address L4_PORT
switch(config-route-map)# set ip next-hop 5.5.5.5 srte-policy color 121
switch(config-route-map)# interface ethernet 1/1
switch(config-route-map)# vrf member vrf1
switch(config-route-map-if)# ip policy route-map FLOW1
```

ネクストホップ名別に選択されたポリシーへのデフォルト以外の VRF でのルートマッピング構成例

```
switch(config)# route-map FLOW1 seq 10
switch(config-route-map)# match ip address L4_PORT
switch(config-route-map)# set ip next-hop 5.5.5.5 srte-policy policy1
switch(config-route-map)# interface ethernet 1/1
switch(config-route-map)# vrf member vrf1
switch(config-route-map-if)# ip policy route-map FLOW1
```

デフォルト以外の VRF でのルートマップの構成例を色とエンドポイントで選択したポリシーにマッピングする

```
switch(config)# route-map FLOW1 seq 10
switch(config-route-map)# match ip address L4_PORT
switch(config-route-map)# set srte-policy color 121 endpoint 10.0.0.1
switch(config-route-map)# interface ethernet 1/1
switch(config-route-map)# vrf member vrf1
switch(config-route-map-if)# ip policy route-map FLOW1
switch(config)# feature bgp
switch(config)# router bgp 1.1
switch(config-router)# vrf vrf1
switch(config-router-vrf)# allocate-index 10
```

名前別に選択されたポリシーへのデフォルト以外の VRF でのルートマッピング構成例

```
switch(config)# route-map FLOW1 seq 10
switch(config-route-map)# match ip address L4_PORT
switch(config-route-map)# set srte-policy policy1
switch(config-route-map)# interface ethernet 1/1
switch(config-route-map)# vrf member vrf1
switch(config-route-map-if)# ip policy route-map FLOW1
switch(config)# feature bgp
switch(config)# router bgp 1.1
switch(config-router)# vrf vrf1
switch(config-router-vrf)# allocate-index 10
```

SRTE のフローベース トラフィック ステアリング構成の確認

SRTE 構成のフローベースのステアリングに関する適切な詳細を表示するには、次のいずれかのタスクを実行します。

表 *10* : *SRTE* のフローベース トラフィック ステアリング構成の確認

コマンド	目的
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 route-map [**name**]	ルート マップの情報を表示します。

コマンド	目的
show forwarding mpls srte module	転送情報ベース - FIB モジュールの SRTE 情報を表示します。

SRTE ポリシーの MPLS OAM モニタリングの構成

SRTE ポリシーの MPLS OAM モニタリングについて

Cisco NX-OS リリース 10.1(2) 以降、MPLS OAM モニタリングにより、1 つ以上の SRTE ポリシーが構成されているスイッチで、SRTE ポリシーのアクティブ パスに障害が発生したかどうかをプロアクティブに検出できます。現在アクティブな優先度の高いパスがすべて失敗した場合、SRTEはその優先度の高いパスがダウンしていると見なし、そのような優先順位があれば、ポリシーで次に高い優先順位をアクティブにします。そうでない場合は、ポリシーをダウンとしてマークします。

この機能の前は、SRTE 優先順位とポリシーの状態は、優先順位内のパスの最初のホップ（最初の MPLS ラベル）の状態によってのみ決定されていました。ラベルがプログラムされている場合、パスは稼働していると見なされ、ラベルがないか無効な場合、パスは停止していると見なされます。

MPLS OAM モニタリングは、MPLS LSPV Nil-FEC ping 要求を SRTE パスに沿って継続的に送信することにより、この検証を強化します。各ping要求には、SRTEポリシーに従うトラフィックに課されるものと同じラベル スタックが含まれているため、pingは同じパスをたどります。ping は、各 ping 間の構成可能な間隔で送信され、パスの最終ノードからの ping への応答は間隔内で期待されます。最終ノードから障害応答が返ってきた場合、または間隔内に応答がなかった場合は、失敗間隔としてカウントされます。構成可能な数の失敗間隔が連続して発生すると、パスはダウンしていると見なされます。優先順位のすべてのパスがダウンしている場合、優先順位はダウンしていると見なされます。

モニタされたパス

CLI がプロアクティブなモニタリングを使用してパスをモニタできる場合にのみ、OAM を使用してパスがモニタされます。ポリシーに関連付けられているパスのみがモニタされます。たとえば、セグメントリストが作成されポリシーに関連付けられていない場合、それはモニタされません。また、同じパスが複数のポリシーで使用されている場合、そのパスに対して作成されるモニタリングセッションは1つだけです。これは、パスがポリシーの基本設定に関連付けられたセグメントリストであるか、ヘッドエンドでパス補完を使用して計算されたものであるかに関係なく適用されます。

デフォルトでは、イメージがOAMモニタリング サポートのないバージョンからモニタリング サポートのあるバージョンにアップグレードされた場合、ポリシーのモニタリング方式は従来のファーストホップ方式になります。

MPLS OAM モニタリングは、すべての SRTE ポリシーに対してグローバルに有効にすることができます。グローバルに有効になっている場合、ポリシーごとに選択的に無効にすることができます。グローバルに有効化されていない場合は、個々のポリシーに対して選択的に有効化できます。

インデックス制限

index-limit X CLI は、パス全体ではなく、パスの最初のサブセットのみを ping するために使用されます。指定された index-limit 以下のセグメント リスト内のインデックスのみが、モニタするパスの一部です。たとえば、セグメント リストが次のようになっているとします。

```
index 100 mpls label 16001
index 200 mpls label 16002
index 300 mpls label 16003
```

次に、index-limit が指定されていない場合、ping されるパスは 16001、16002、16003 になります。index-limit が 250 の場合、ping されるパスは 16001、16002 になります。index-limit が 200 の場合、ping されるパスも 16001、16002 になります。

SRTE ポリシーの MPLS OAM モニタリングに関する注意事項と制限事項

SRTE ポリシーの MPLS OAM モニタリングには、次のガイドラインと制限事項があります。

- Cisco NX-OS リリース 10.1(2) 以降、MPLS OAM モニタリング（継続的かつ予防的なパス）が導入され、Cisco Nexus 9300 EX、9300-FX、9300-FX2、および 9300-GX プラットフォーム スイッチでサポートされています。
- SRTE ポリシーが構成されているヘッドエンド ノードでは、SRTE と MPLS OAM の両方を、それぞれ feature mpls segment-routing traffic-engineering および feature mpls oam の一部として個別に有効にする必要があります。そうでない場合、ユーザーは OAM を使用して SRTE ポリシーのモニタリングを構成できません。さらに、SR ファブリックの残りのノードでは、MPLS OAM モニタリングによって送信された ping に応答するために、feature mpls oam を使用して MPLS OAM を有効にする必要があります。
- SRTE は、モニタリング セッションの最大数を 1000 に制限します。
- ping の最小間隔は 1000 ミリ秒です。
- SRTE OAM モニタリング ポリシーがデバイスで実行されている場合、`feature mpls oam` を無効にすることはできません。すべての SRTE OAM モニタリング ポリシーが無効になっている場合にのみ、デバイスから `feature mpls oam` を無効にできます。それ以外の場合、次のエラー メッセージが表示されます。

  「SRTE MPLS 活性検出は、すべてのポリシーに対して有効になっているか、少なくとも 1 つのポリシーに対して有効になっているか、またはオンデマンド カラーに対して有効になっています。MPLS OAM を無効にする前に、活性検出が完全に無効になっていることを確認してください。」

- Cisco NX-OS リリース 10.1(2) では、SRTE OAM モニタリングは、スタティック ポリシーと、明示パスが構成されているオンデマンド カラーに対してサポートされています。
- OAM セッションは、PCEP を使用してダイナミック オプションで構成されたパスでは実行されません。

MPLS OAM モニタリングの構成

このセクションでは、ポリシーのプロアクティブなパス モニタリングを有効にするために必要な CLI について説明します。

- **グローバル設定**

  この構成により、構成されたすべてのポリシーの OAM パス モニタリングが有効になります。

- **ポリシー固有の構成**

  この構成により、特定のポリシーの OAM パス モニタリングが有効になります。

グローバル設定

始める前に

MPLS セグメント ルーティング トラフィック エンジニアリング機能が有効になっていることを確認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	**segment-routing**
	例：
	`switch(config)#segment-routing`
	`switch(config-sr)#`
		セグメント ルーティング構成モードを開始します。

ステップ 3	**traffic-engineering**
	例：
	`switch(config-sr)# traffic-engineering`
	`switch(config-sr-te)#`
		トラフィック エンジニアリング モードに入ります。

ステップ 4	[**liveness-detection**] 例：	活性検出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
	`switch(config-sr-te)# liveness-detection switch(config-sr-te-livedet)#`	
ステップ 5	**interval** *num* 例： `switch(config-sr-te-livedet)# interval 6000 switch(config-sr-te-livedet)#`	間隔はミリ秒です。デフォルトは3000 ms です。
ステップ 6	**multiplier** *num* 例： `switch(config-sr-te-livedet)# multiplier 5 switch(config-sr-te-livedet)#`	乗数は、乗数は、ダウンと見なされるためにアップしているパスの失敗する必要がある連続間隔数と、アップとみなされるためにダウンしているパスの連続間隔数を設定します。デフォルトは 3 です。
ステップ 7	**mpls** 例： `switch(config-sr-te-livedet)# mpls switch(config-sr-te-livedet-mpls)#`	mpl を介したセグメント ルーティングを有効にします。
ステップ 8	[**no**]**oam** 例： `switch(config-sr-te-livedet-mpls)# oam switch(config-sr-te-livedet-mpls)#`	すべての SRTE ポリシーに対して MPLS OAM モニタリングをグローバルに有効にします。 このコマンドの no 形式で、OAM モニタリングを無効にします。
ステップ 9	**segment-list name** *sidlist-name* 例： `switch(config-sr-te)# segment-list name blue index 10 mpls label 16004 index 10 mpls label 16005`	明示 SID リストを作成します。 （注） このコマンドは、sidlist-name の自動入力機能があります。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。
ステップ 10	**policy** *policy name* 例： `switch(config-sr-te)# policy 1 switch(config-sr-te-pol)`	ポリシーを設定します。
ステップ 11	**color** *numberIP-end-point* 例： `switch(config-sr-te-pol)# color 1 endpoint 5.5.5.5 switch(config-sr-te-pol)`	ポリシーのカラーとエンドポイントを設定します。

	コマンドまたはアクション	目的

ステップ 12	**candidate-paths**
	例：
	`switch(config-sr-te-pol)# candidate-paths`
	`switch(config-expcndpaths)#`
		ポリシーの候補パスを指定します。

ステップ 13	**preference** *preference-number*
	例：
	`switch(config-expcndpaths)# preference 100`
	`switch(cfg-pref)#`
		候補パスの優先順位を指定します。

ステップ 14	*sidlist-name*explicit segment-list
	例：
	`switch(cfg-pref)# explicit segment-list red`
	`switch(cfg-pref)#`
		明示リストを指定します。
		（注） このコマンドは、sidlist-name の自動入力機能があります。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。

ステップ 15	**on-demand color** *color_num*
	例：
	`switch(config-sr-te)# on-demand color 211`
	`switch(config-sr-te-color)#`
		オンデマンド色テンプレートモードを開始し、特定の色のオンデマンド色を構成します。

ステップ 16	**candidate-paths**
	例：
	`switch(config-sr-te-color)# candidate-paths`
	`switch(cfg-cndpath)#`
		ポリシーの候補パスを指定します。

ステップ 17	**preference** *preference-number*
	例：
	`switch(cfg-cndpath)# preference 100`
	`switch(cfg-pref)#`
		候補パスの優先順位を指定します。

ステップ 18	*sidlist-name*explicit segment-list
	例：
	`switch(cfg-pref)# explicit segment-list red`
	`switch(cfg-pref)#`
		明示リストを指定します。
		（注） このコマンドは、sidlist-name の自動入力機能があります。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。

ポリシー固有の構成

始める前に

MPLS セグメント ルーティング トラフィック エンジニアリング機能が有効になっていることを確認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	**segment-routing**
	例：
	`switch(config)#segment-routing`
	`switch(config-sr)#`
		セグメントルーティング構成モードを開始します。

ステップ 3	**traffic-engineering**
	例：
	`switch(config-sr)# traffic-engineering`
	`switch(config-sr-te)#`
		トラフィックエンジニアリングモードに入ります。

ステップ 4	[**liveness-detection**]
	例：
	`switch(config-sr-te)#`
	`liveness-detection`
	`switch(config-sr-te-livedet)#`
		活性検出構成モードを開始します。

ステップ 5	**interval** *num*
	例：
	`switch(config-sr-te-livedet)# interval 6000`
	`switch(config-sr-te-livedet)#`
		間隔はミリ秒です。デフォルトは3000 ms です。

ステップ 6	**multiplier** *num*
	例：
	`switch(config-sr-te-livedet)#`
	`multiplier 5`
	`switch(config-sr-te-livedet)#`
		乗数は、乗数は、ダウンと見なされるためにアップしているパスの失敗する必要がある連続間隔数と、アップとみなされるためにダウンしているパスの連続間隔数を設定します。デフォルトは 3 です。

ステップ 7	**segment-list name** *sidlist-name* 例：	明示 SID リストを作成します。

	コマンドまたはアクション	目的

	`switch(config-sr-te)# segment-list name blue`
	`index 10 mpls label 16004`
	`index 10 mpls label 16005`
		（注） このコマンドは、sidlist-name の自動入力機能があります。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。

ステップ 8	**policy** *policy name*
	例：
	`switch(config-sr-te)# policy 1`
	`switch(config-sr-te-pol)`
		ポリシーを設定します。

ステップ 9	**color** *numberIP-end-point*
	例：
	`switch(config-sr-te-pol)# color 1 endpoint 5.5.5.5`
	`switch(config-sr-te-pol)`
		ポリシーのカラーとエンドポイントを設定します。

ステップ 10	**candidate-paths**
	例：
	`switch(config-sr-te-pol)# candidate-paths`
	`switch(config-expcndpaths)#`
		ポリシーの候補パスを指定します。

ステップ 11	**preference** *preference-number*
	例：
	`switch(config-expcndpaths)# preference 100`
	`switch(cfg-pref)#`
		候補パスの優先順位を指定します。

ステップ 12	*sidlist-name*explicit segment-list
	例：
	`switch(cfg-pref)# explicit segment-list red`
	`switch(cfg-pref)#`
		明示リストを指定します。
		（注） このコマンドは、sidlist-name の自動入力機能があります。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。

ステップ 13	[**liveness-detection**]
	例：
	`switch(config-sr-te)# liveness-detection`
	`switch(config-sr-te-livedet)#`
		活性検出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 14	[**no**]**index-limit** *num*
	例：
	`switch(config-sr-te-livedet)# index-limit 20`
	`switch(config-sr-te-livedet)#`
		ユーザーが指定した数以下のインデックスを持つ SID のみをモニタします。

ステップ 15	[**no**]**shutdown**
	例：
	`switch(config-sr-te-livedet)# shutdown`
	`switch(config-sr-te-livedet)#`
		活性検出を無効にします。これは、関連するすべての構成を完全に削除せずに、活性検出を一時的に無効にする場合に便利です。
		このコマンドの no 形式で、OAM モニタリングを無効にします。

ステップ 16	**mpls**
	例：
	`switch(config-sr-te-livedet)# mpls`
	`switch(config-sr-te-livedet-mpls)#`
		mpl を介したセグメント ルーティングを有効にします。

ステップ 17	[**no**]**oam**
	例：
	`switch(config-sr-te-livedet-mpls)# oam`
	`switch(config-sr-te-livedet-mpls)#`
		すべてのSRTEポリシーに対してMPLS OAMモニタリングをグローバルに有効にします。
		このコマンドの no 形式で、OAM モニタリングを無効にします。

ステップ 18	**on-demand color** *color_num*
	例：
	`switch(config-sr-te)# on-demand color 211`
	`switch(config-sr-te-color)#`
		オンデマンド色テンプレートモードを開始し、特定の色のオンデマンド色を構成します。

ステップ 19	**candidate-paths**
	例：
	`switch(config-sr-te-color)# candidate-paths`
	`switch(cfg-cndpath)#`
		ポリシーの候補パスを指定します。

ステップ 20	**preference** *preference-number*
	例：
	`switch(cfg-cndpath)# preference 100`
	`switch(cfg-pref)#`
		候補パスの優先順位を指定します。

ステップ 21	*sidlist-name*explicit segment-list 例：	明示リストを指定します。

	コマンドまたはアクション	目的
	`switch(cfg-pref)# explicit segment-list red` `switch(cfg-pref)#`	（注） このコマンドは、sidlist-name の自動入力機能があります。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。

ステップ 22	[**liveness-detection**]
	例：
	`switch(config-sr-te-color)# liveness-detection`
	`switch(config-sr-te-color-livedet)#`
		活性検出構成モードを開始します。

ステップ 23	[**no**]**index-limit** *num*
	例：
	`switch(config-sr-te-color-livedet)# index-limit 20`
	`switch(config-sr-te-color-livedet)#`
		ユーザーが指定した数以下のインデックスを持つ SID のみをモニタします。

ステップ 24	[**no**]**shutdown**
	例：
	`switch(config-sr-te-color-livedet)# shutdown`
	`switch(config-sr-te-color-livedet)#`
		活性検出を無効にします。これは、関連するすべての構成を完全に削除せずに、活性検出を一時的に無効にする場合に便利です。
		このコマンドの no 形式で、OAM モニタリングを無効にします。

ステップ 25	**mpls**
	例：
	`switch(config-sr-te-color-livedet)# mpls`
	`switch(config-sr-te-color-livedet-mpls)#`
		mpl を介したセグメント ルーティングを有効にします。

ステップ 26	[**no**]**oam**
	例：
	`switch(config-sr-te-color-livedet-mpls)# oam`
	`switch(config-sr-te-color-livedet-mpls)#`
		すべての SRTE ポリシーに対して MPLS OAM モニタリングをグローバルに有効にします。
		このコマンドの no 形式で、OAM モニタリングを無効にします。

MPLS OAM モニタリングの構成の確認

MPLS OAM モニタリングの構成情報を表示するには、次のタスクのいずれかを実行します。

表 *11 : MPLS OAM* モニタリングの構成の確認

コマンド	目的
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。

コマンド	目的
show srte policy proactive-policy-monitoring	promon データベースに存在するすべてのアクティブなプロアクティブ ポリシー モニタリング セッションのリストを表示します。 （注） このコマンドの最後に疑問符オプションを使用して、次のオプションのいずれかを指定するか、Enter キーを押してすべてのセッションを表示できます。 • brief：セッションに関する簡単な情報を表示します • color：ポリシーのカラーに関連する promon セッションを示します • name：ポリシー名に関連する Promon セッションを表示します • セッション ID：セッション ID の Promon セッションを表示します
show srte policy proactive-policy-monitoring [**brief**]	セッション ID のリストとプロアクティブ ポリシー モニタリング セッションの状態のみを表示します。
show srte policy proactive-policy-monitoring [**session** <*session-id*>]	セッション ID を使用してフィルタリングし、そのセッションに関する情報を詳細に表示します。 （注） このコマンドには、セッション ID の自動入力機能があります。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。

コマンド	目的
show srte policy proactive-policy-monitoring color <*color*> **endpoint**<*endpoint*>	カラーとエンドポイントを使用してフィルタリングし、プロアクティブなポリシー モニタリング セッションを表示します。 （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。

MPLS OAM モニタリングの構成例

次に、MPLS OAM モニタリングの構成例を示します。

- ユーザー指定の乗数と間隔によるグローバル有効化の構成例：

```
segment-routing
  traffic-engineering
    liveness-detection
        interval 6000
        multiplier 5
      mpls
        oam
    segment-list name blue
      index 10 mpls label 16004
      index 20 mpls label 16005
    segment-list name green
      index 10 mpls label 16003
      index 20 mpls label 16006
    segment-list name red
      index 10 mpls label 16002
      index 20 mpls label 16004
      index 30 mpls label 16005
    policy customer-1
      color 1 endpoint 5.5.5.5
      candidate-paths
        preference 100
          explicit segment-list red
    on-demand color 211
      candidate-paths
        preference 100
          explicit segment-list green
```

- ユーザー指定の乗数、間隔、インデックス制限、およびシャットダウン オプションを使用したポリシー有効化の構成例：

```
segment-routing
  traffic-engineering
    liveness-detection
        interval 6000
        multiplier 5
    segment-list name blue
      index 10 mpls label 16004
      index 20 mpls label 16005
    segment-list name green
```

```
  index 10 mpls label 16003
  index 20 mpls label 16006
segment-list name red
  index 10 mpls label 16002
  index 20 mpls label 16004
  index 30 mpls label 16005
policy customer-1
  color 1 endpoint 5.5.5.5
  candidate-paths
    preference 100
      explicit segment-list red
  liveness-detection
    index-limit 20
    shutdown
    mpls
      oam
on-demand color 211
  candidate-paths
    preference 100
      explicit segment-list green
  liveness-detection
      index-limit 20
      shutdown
      mpls
        oam
```

SRTE 向け BFD の構成

このセクションでは、1 つまたは複数の SRTE ポリシーが構成されているスイッチで、SRTE ポリシーのアクティブパスまたはパスに障害が発生した場合にそれをプロアクティブに検出できるように、SRTE ポリシーに BFD モニタリングを構成する方法について説明します。

SRTE の BFD について

SRTE の BFD は、SRTE ポリシーの MPLS OAM モニタリングに似ています。SRTE 向けの BFD により、1 つ以上の SRTE ポリシーが構成されているスイッチで、SRTE ポリシーのアクティブパスに障害が発生したかどうかをプロアクティブに検出できます。現在アクティブな優先度の高いパスがすべて失敗した場合、SRTEはその優先度の高いパスがダウンしていると見なし、そのような優先順位があれば、ポリシーで次に高い優先順位をアクティブにします。そうでない場合は、ポリシーをダウンとしてマークします。

SRTE の BFD は、SRTE パスに沿って BFD プローブを継続的に送信することによって検出を実行します。各プローブは、SRTE ポリシーに従うトラフィックに適用されるのと同じラベルスタックを持つ MPLS にカプセル化され、プローブが同じパスをたどるようにします。さらに、プローブのラベル スタックの最も内側にもう 1 つのラベルが適用されます。これにより、ポリシーの最終ノードのデータプレーンに到達すると、プローブが送信者に返されます。これは、プローブが最終ノードによって受信され、コントロールプレーンで処理され、応答が返される SRTE ポリシーの MPLS OAM モニタリングとは異なります。

プローブは、各プローブ間の構成可能な間隔で送信され、プローブはその間隔内で送信者にループバックすることが期待されます。構成可能な数の失敗間隔が連続して発生すると、パス

はダウンしていると見なされます。優先順位のすべてのパスがダウンしている場合、優先順位はダウンしていると見なされます。

モニタされたパス

コマンドがプロアクティブなモニタリングを使用してパスをモニタできる場合にのみ、BFDを使用してパスがモニタされます。ポリシーに関連付けられているパスのみがモニタされます。たとえば、セグメントリストが作成されポリシーに関連付けられていない場合、それはモニタされません。また、同じパスが複数のポリシーで使用されている場合、そのパスに対して作成されるモニタリングセッションは1つだけです。これは、パスがポリシーの基本設定に関連付けられたセグメントリストであるか、ヘッドエンドでパス補完を使用して計算されたものであるかに関係なく適用されます。MPLS OAM モニタリングは、すべての SRTE ポリシーに対してグローバルに有効にすることができます。グローバルに有効になっている場合、ポリシーごとに選択的に無効にすることができます。グローバルに有効化されていない場合は、個々のポリシーに対して選択的に有効化できます。ポリシーがモニタされると、SRTE は実行可能な最も高い設定をプライマリ設定として選択し、次に高い設定をバックアップとして選択します。このプライマリとバックアップは転送プレーンにプログラムされているため、プライマリ パスの障害が BFD で検出された場合、転送レイヤはコントロール プレーンの SRTE からの介入を必要とせずにバックアップパスにすぐに切り替えることができます。これにより、障害回復に必要な時間が短縮されます。

インデックス制限

index-limit X コマンドは、パス全体ではなく、パスの最初のサブセットのみを検証するために使用されます。指定された index-limit 以下のセグメントリスト内のインデックスのみが、モニタするパスの一部です。たとえば、セグメント リストが次のようになっているとします。

- インデックス 100 mpls ラベル 16001
- インデックス 200 mpls ラベル 16002
- インデックス 300 mpls ラベル 16003

次に、index-limit が指定されていない場合、検証されるパスは 16001、16002、16003 になります。index-limit が 250 の場合、検証されるパスも 16001、16002 になります。index-limit が 200 の場合、検証されるパスも 16001、16002 になります。

SRTE 向け BFD に関する注意事項および制限事項

SRTEポリシー向けにBFDモニタリングを構成するための注意事項と制限事項は、以下のとおりです。

- Cisco NX-OS リリース 10.3(2)F 以降、SRTE ポリシーの BFD モニタリングが導入され、9300-FX、9300-FX2、9300-FX3、9300-GX、9300-GX2、N9K-C9364C、およびN9K- C9332C TOR プラットフォームのみでサポートされます。
- IPv4 アンダーレイを使用した SRTE MPLS のみが、BFD を使用したモニタリングでサポートされます。SRv6 ポリシーはサポートされていません。

- この形式のモニタリングを使用する場合、vPC はヘッドエンドでサポートされません。
- 一度に有効にできるのは、OAM または BFD モニタリングのいずれか 1 つだけです。つまり、OAM を使用して一部のポリシーをモニタし、BFD を使用して一部のポリシーをモニタすることはできません。
- IP リダイレクトは、到着したばかりのインターフェイスを終了する必要がある場合があるため、BFD プローブが送信者にループバックするノードの SR 対応コア インターフェイスで無効にする必要があります。
- SRTE がモニタリング パスに使用する最も内側のラベル（ヘッドエンド ラベル）は、エニーキャスト SID であってはなりません。同じエニーキャスト アドレスを共有する別のノードに応答が送信されないように、そのノードの一意の SID である必要があります。
- 転送するようにプログラムされている場合、特定のポリシーの ECMP メンバーの総数は 8 です。これには、プライマリ ECMP メンバーとバックアップ ECMP メンバーが含まれます。ポリシーのプライマリ設定とバックアップ設定の間に 8 を超える ECMP メンバーがある場合、8 のみが使用されます。
- SRTE ヘッドエンドノード（ポリシーが定義されている）で定義されている SRGB 範囲と、BFD活性検出によってモニタされるすべてのパスの最終ノードで定義されている SRGB 範囲は同じである必要があります。そして、SRGB の範囲はすべてのノードで同じにすることをお勧めします。BFD プローブパケットに追加された送信者への返信ラベルは、ローカルループバックインターフェイスのプレフィックスの connected-prefix-sid-map SR 構成から SRTE ヘッドエンドノードでローカルに学習されるため、そのラベルの値はパケットを返すノードで同じです。
- BFD モニタリングは、ダイナミック pcep オプションを使用したパス設定ではサポートされません。

SRTE 向け BFD の構成

このセクションでは、SRTE ポリシー向け BFD 保護を使用して、プロアクティブ パス モニタリングを有効にするために必要なコマンドを説明します。構成タスクは、すべてのポリシーまたは特定のポリシーのどちらに対して構成するかに基づいて、次の方法で実行できます。

- **グローバル構成**：この構成では、構成されているすべてのポリシーに対して BFD 保護が有効になります。
- **ポリシー固有の構成**：この構成では、特定のポリシーの BFD 保護を有効にします。

グローバル設定

始める前に

次の機能が有効になっていることを確認する必要があります。

- feature bfd

- feature mpls segment-routing
- feature mpls segment-routing traffic-engineering

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)#`
		セグメントルーティング構成モードを開始します。

ステップ 3	**traffic-engineering**
	例：
	`switch(config-sr)# traffic-engineering`
	`switch(config-sr-te)#`
		トラフィック エンジニアリング モードに入ります。

ステップ 4	[**no**] **liveness-detection**
	例：
	`switch(config-sr-te)#`
	`liveness-detection`
	`switch(config-sr-te-livedet)#`
		活性検出構成モードを開始します。

ステップ 5	**interval** *num*
	例：
	`switch(config-sr-te-livedet)# interval`
	`6000`
	`switch(config-sr-te-livedet)#`
		間隔はミリ秒です。デフォルトは3000 ms です。

ステップ 6	**multiplier** *num*
	例：
	`switch(config-sr-te-livedet)#`
	`multiplier 5`
	`switch(config-sr-te-livedet)#`
		乗数は、ダウンと見なされるためにアップしているパスの失敗する必要がある連続間隔数を設定します。BFD モニタリングが使用されている場合、プローブが成功すると、ダウンしているパスがアップと見なされます。デフォルトは 3 です。

ステップ 7	**mpls**
	例：
	`switch(config-sr-te-livedet)# mpls`
	`switch(config-sr-te-livedet-mpls)#`
		活性検出のため MPLS データ プレーン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 8	[**no**] **bfd**
	例：
	`switch(config-sr-te-livedet-mpls)# bfd`
	`switch(config-sr-te-livedet-mpls)#`
		すべての SRTE ポリシーに対して BFD 保護をグローバルに有効にします。
		このコマンドの no フォームは BFD 保護を無効にします。

ステップ 9	**segment-list name** *sidlist-name*
	例：
	`switch(config-sr-te)# segment-list name blue`
	`index 10 mpls label 16004`
	`index 10 mpls label 16005`
		明示 SID リストを作成します。
		（注） このコマンドは、sidlist-name の自動入力機能があります。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。

ステップ 10	**policy** *policy name*
	例：
	`switch(config-sr-te)# policy 1`
	`switch(config-sr-te-pol)`
		ポリシーを設定します。

ステップ 11	**color** *color* end-point *address*
	例：
	`switch(config-sr-te-pol)# color 1 endpoint 5.5.5.5`
	`switch(config-sr-te-pol)`
		ポリシーのカラーとエンドポイントを設定します。

ステップ 12	**candidate-paths**
	例：
	`switch(config-sr-te-pol)# candidate-paths`
	`switch(config-expcndpaths)#`
		ポリシーの候補パスを指定します。

ステップ 13	**preference** *preference-number*
	例：
	`switch(config-expcndpaths)# preference 100`
	`switch(cfg-pref)#`
		候補パスの優先順位を指定します。

ステップ 14	*sidlist-name*explicit segment-list
	例：
	`switch(cfg-pref)# explicit segment-list red`
	`switch(cfg-pref)#`
		明示リストを指定します。
		（注） このコマンドは、sidlist-name の自動入力機能があります。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。

	コマンドまたはアクション	目的

ステップ 15	**on-demand color** *color_num*
	例：
	`switch(config-sr-te)# on-demand color 211`
	`switch(config-sr-te-color)#`
		オンデマンド色テンプレートモードを開始し、特定の色のオンデマンド色を構成します。

ステップ 16	**candidate-paths**
	例：
	`switch(config-sr-te-color)# candidate-paths`
	`switch(cfg-cndpath)#`
		ポリシーの候補パスを指定します。

ステップ 17	**preference** *preference-number*
	例：
	`switch(cfg-cndpath)# preference 100`
	`switch(cfg-pref)#`
		候補パスの優先順位を指定します。

ステップ 18	*sidlist-name*explicit segment-list
	例：
	`switch(cfg-pref)# explicit segment-list red`
	`switch(cfg-pref)#`
		明示リストを指定します。
		（注） このコマンドは、sidlist-name の自動入力機能があります。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。

ポリシー固有の構成

始める前に

次の機能が有効になっていることを確認する必要があります。

- feature bfd
- feature mpls segment-routing
- feature mpls segment-routing traffic-engineering

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	**segment-routing**
	例：
	`switch(config)#segment-routing`
	`switch(config-sr)#`
		セグメントルーティング構成モードを開始します。

ステップ 3	**traffic-engineering**
	例：
	`switch(config-sr)# traffic-engineering`
	`switch(config-sr-te)#`
		トラフィックエンジニアリングモードに入ります。

ステップ 4	[**no**] **liveness-detection**
	例：
	`switch(config-sr-te)#`
	`liveness-detection`
	`switch(config-sr-te-livedet)#`
		活性検出構成モードを開始します。

ステップ 5	**interval** *num*
	例：
	`switch(config-sr-te-livedet)# interval 6000`
	`switch(config-sr-te-livedet)#`
		間隔はミリ秒です。デフォルトは3000 ms です。

ステップ 6	**multiplier** *num*
	例：
	`switch(config-sr-te-livedet)#`
	`multiplier 5`
	`switch(config-sr-te-livedet)#`
		乗数は、ダウンと見なされるためにアップしているパスの失敗が必要がある連続間隔数を設定します。BFD モニタリングが使用されている場合、プローブが成功すると、ダウンしているパスがアップと見なされます。デフォルトは 3 です。

ステップ 7	**segment-list name** *sidlist-name*
	例：
	`switch(config-sr-te)# segment-list name blue`
	`index 10 mpls label 16004`
	`index 10 mpls label 16005`
		明示 SID リストを作成します。

ステップ 8	**policy** *policy name*
	例：
	`switch(config-sr-te)# policy 1`
	`switch(config-sr-te-pol)`
		ポリシーを設定します。

ステップ 9	**color** *color* end-point *address*
	例：
	`switch(config-sr-te-pol)# color 1 endpoint 5.5.5.5`
	`switch(config-sr-te-pol)`
		ポリシーのカラーとエンドポイントを設定します。

	コマンドまたはアクション	目的

ステップ 10	**candidate-paths**
	例：
	`switch(config-sr-te-pol)# candidate-paths`
	`switch(config-expcndpaths)#`
		ポリシーの候補パスを指定します。

ステップ 11	**preference** *preference-number*
	例：
	`switch(config-expcndpaths)# preference 100`
	`switch(cfg-pref)#`
		候補パスの優先順位を指定します。

ステップ 12	*sidlist-name*explicit segment-list
	例：
	`switch(cfg-pref)# explicit segment-list red`
	`switch(cfg-pref)#`
		明示リストを指定します。

ステップ 13	[**no**] **liveness-detection**
	例：
	`switch(config-sr-te)# liveness-detection`
	`switch(config-sr-te-livedet)#`
		活性検出構成モードを開始します。

ステップ 14	[**no**]**index-limit** *num*
	例：
	`switch(config-sr-te-livedet)# index-limit 20`
	`switch(config-sr-te-livedet)#`
		ユーザーが指定した数以下のインデックスを持つ SID のみをモニタします。

ステップ 15	[**no**]**shutdown**
	例：
	`switch(config-sr-te-livedet)# shutdown`
	`switch(config-sr-te-livedet)#`
		活性検出を無効にします。これは、関連するすべての構成を完全に削除せずに、活性検出を一時的に無効にする場合に便利です。

ステップ 16	**mpls**
	例：
	`switch(config-sr-te-livedet)# mpls`
	`switch(config-sr-te-livedet-mpls)#`
		活性検出のため MPLS データ プレーン構成モードを開始します。

ステップ 17	[**no**] **bfd**
	例：
	`switch(config-sr-te-livedet-mpls)# oam`
	`switch(config-sr-te-livedet-mpls)#`
		構成されているポリシーの BFD 活性検出を有効にします。
		このコマンドの no 形式を使用すると、BFD 活性検出が構成されているポリシーの BFD 活性検出が無効になります。

	コマンドまたはアクション	目的

ステップ 18	**on-demand color** *color_num*
	例：
	`switch(config-sr-te)# on-demand color 211`
	`switch(config-sr-te-color)#`
		オンデマンド色テンプレートモードを開始し、特定の色のオンデマンド色を構成します。

ステップ 19	**candidate-paths**
	例：
	`switch(config-sr-te-color)# candidate-paths`
	`switch(cfg-cndpath)#`
		ポリシーの候補パスを指定します。

ステップ 20	**preference** *preference-number*
	例：
	`switch(cfg-cndpath)# preference 100`
	`switch(cfg-pref)#`
		候補パスの優先順位を指定します。

ステップ 21	*sidlist-name*explicit segment-list
	例：
	`switch(cfg-pref)# explicit segment-list red`
	`switch(cfg-pref)#`
		明示リストを指定します。

ステップ 22	[**no**] **liveness-detection**
	例：
	`switch(config-sr-te-color)# liveness-detection`
	`switch(config-sr-te-color-livedet)#`
		活性検出構成モードを開始します。

ステップ 23	[**no**] **index-limit** *num*
	例：
	`switch(config-sr-te-color-livedet)# index-limit 20`
	`switch(config-sr-te-color-livedet)#`
		ユーザーが指定した数以下のインデックスを持つ SID のみをモニタします。

ステップ 24	[**no**] **shutdown**
	例：
	`switch(config-sr-te-color-livedet)# shutdown`
	`switch(config-sr-te-color-livedet)#`
		活性検出を無効にします。これは、関連するすべての構成を完全に削除せずに、活性検出を一時的に無効にする場合に便利です。

ステップ 25	**mpls**
	例：
	`switch(config-sr-te-color-livedet)# mpls`
	`switch(config-sr-te-color-livedet-mpls)#`
		活性検出のため MPLS データ プレーン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 26	**[no] bfd**
	例：
	`switch(config-sr-te-color-livedet-mpls)# bfd`
	`switch(config-sr-te-color-livedet-mpls)#`
		構成されているオンデマンドカラーの BFD 活性検出を有効にします。
		このコマンドの no 形式を使用すると、構成されているオンデマンドカラーの BFD 活性検出が無効になります。

SRTE の BFD の構成例

次に、SRTE の BFD を設定する例を示します。

```
feature mpls segment-routing traffic-engineering segment-routing
traffic-engineering
liveness-detection
      multiplier NUM
      interval NUM
      mpls
        bfd
    segment-list name SEGLIST1
      index 100 mpls label 16001
      index 200 mpls label 16002
      index 300 mpls label 16003
    on-demand color 702
      explicit segment-list SEGLIST1
      liveness-detection
        mpls
          bfd
        index-limit 200
    policy name POL1
      color 20 endpoint 1.1.1.1
      liveness-detection
        mpls
          bfd
        index-limit 200
```

SRTE の BFD の構成の確認

SRTE ポリシー構成の BFD モニタリングを表示するには、次の作業のいずれかを行います。

表 *12 : MPLS OAM* モニタリングの構成の確認

コマンド	目的
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。

コマンド	目的
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、ポリシー名の自動入力機能があります。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自動入力機能があります。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 proactive-policy-monitoring	promon データベースに存在するすべてのアクティブなプロアクティブ ポリシー モニタリング セッションのリストを表示します。 （注） このコマンドの最後に疑問符オプションを使用して、次のオプションのいずれかを指定するか、Enter キーを押してすべてのセッションを表示できます。 • brief：セッションに関する簡単な情報を表示します • color：ポリシーのカラーに関連する promon セッションを示します • name：ポリシー名に関連する Promon セッションを表示します • セッション ID：セッション ID の Promon セッションを表示します
show srte policy proactive-policy-monitoring [**brief**]	セッション ID のリストとプロアクティブ ポリシー モニタリング セッションの状態のみを表示します。

コマンド	目的
show srte policy proactive-policy-monitoring [**session** <*session-id*>]	セッションIDを使用してフィルタリングし、そのセッションに関する情報を詳細に表示します。 （注） このコマンドには、セッションIDの自動入力機能があります。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 proactive-policy-monitoring color <*color*> **endpoint**<*endpoint*>	カラーとエンドポイントを使用してフィルタリングし、プロアクティブなポリシー モニタリング セッションを表示します。 （注） このコマンドには、カラーとエンドポイントの自動入力機能があります。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 mpls switching detail	このコマンドは、ユニキャストラベルデータベースを表示します。これは、SRTEポリシーFECの各NHLFEに使用されるモニタリングラベルを表示するために使用でき、SRTEモニタリングFEC自体を表示するために使用できます。
show bfd neighbors	BFDセッションの詳細を表示します。

セグメントルーティングでの出力ピアエンジニアリングの設定

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。

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。

セグメント ルーティングを使用した BGP 出力ピア エンジニアリングの概要

Cisco Nexus 9000 シリーズ スイッチは、多くの場合、大規模データセンター（MSDC）に導入されます。このような環境では、セグメントルーティング（SR）でBGP出力ピアエンジニアリング（EPE）をサポートすることが要件となります。

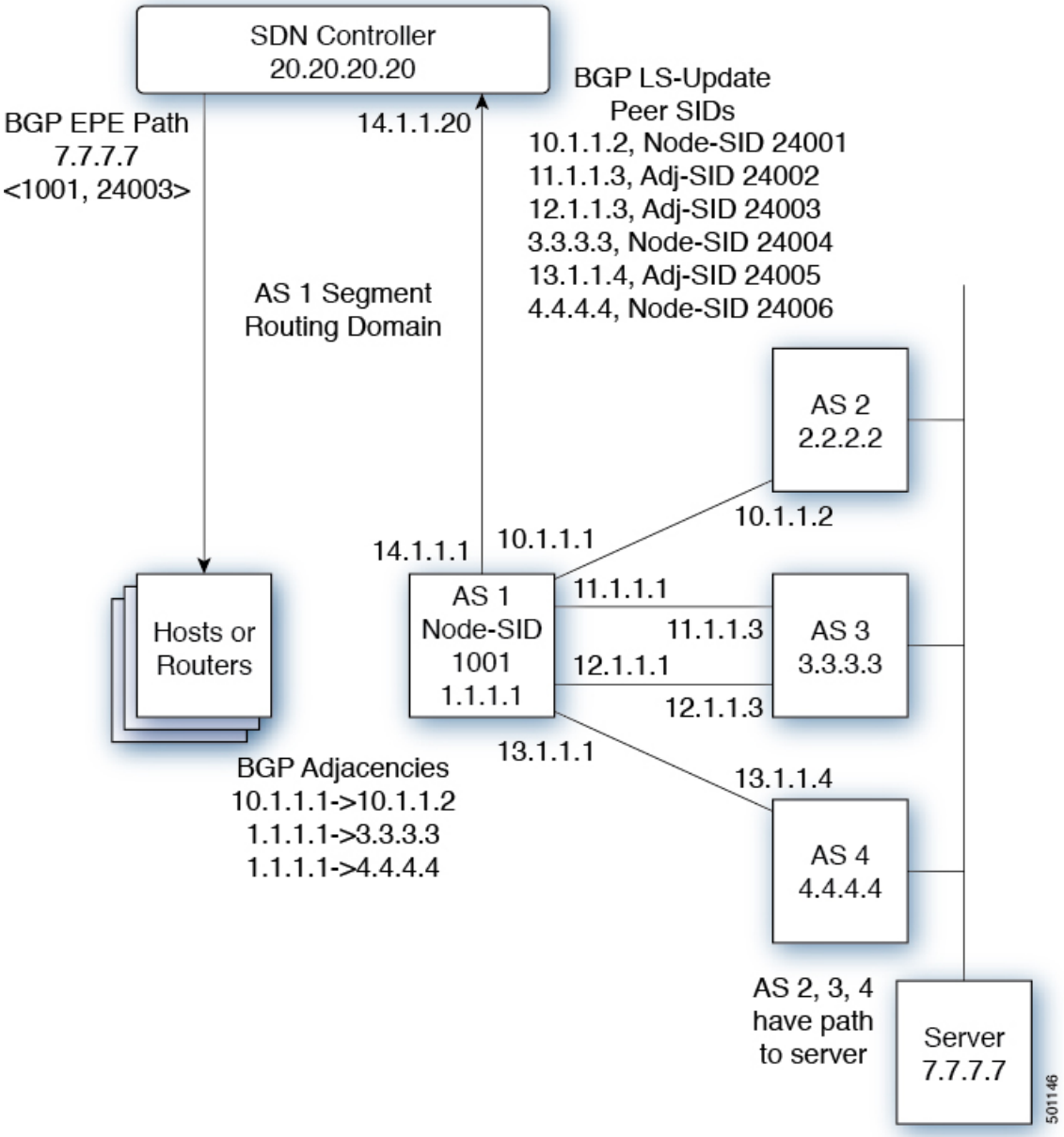
セグメント ルーティング（SR）はソース ルーティングを利用します。ノードは、制御された一連の命令（セグメント）によってパケットを操作するために、パケットの前に SR ヘッダーを付加します。セグメントは、トポロジまたはサービスベースの命令を表すことができます。SRでは、SR ドメインの入力ノードでのみフローごとの状態を維持しながら、トポロジパスまたはサービス チェーンを介してフローを操作できます。この機能の場合、セグメントルーティング アーキテクチャは、MPLS データ プレーンに直接適用されます。

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 プレフィックスは常に SR または 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プレフィックスは、BGP プレフィックス セグメントの識別子です。

SR ベースの出力ピア エンジニアリング（EPE）ソリューションにより、集中型（SDN）コントローラは、ドメイン内の入力境界ルータまたはホストで任意の出力ピア ポリシーをプログラムできます。

次の例では、3 つのルータすべてが iBGP を実行し、NRLI を相互にアドバタイズします。また、ルータはループバックをネクストホップとしてアドバタイズし、再帰的に解決します。これにより、図に示すように、ルータ間に ECMP が提供されます。

図 *13 :* 出力ピア エンジニアリングの例



SDN コントローラは、そのピアおよび隣接のそれぞれについて、出力ルータ 1.1.1.1 からのセグメント ID を受信します。次に、出口ポイントをコントローラのルーティング ドメイン内の他のルータおよびホストにインテリジェントにアドバタイズできます。図に示すように、BGP ネットワーク層到達可能性情報（NLRI）には、ルータ 1.1.1.1 へのノード SID と、7.7.7.7 へのトラフィックがリンク 12.1.1.1->12.1.1.3 を介して出力されることを示すピア隣接 SID 24003 の両方が含まれています。 。

BGP 出力ピア エンジニアリングのガイドラインと制限事項

BGP 出力ピア エンジニアリングには、次のガイドラインと制限事項があります。

- BGP 出力ピア エンジニアリングは、IPv4 BGP ピアでのみサポートされています。IPv6 BGP ピアはサポートされていません。
- BGP 出力ピア エンジニアリングは、デフォルトの VPN ルーティングおよび転送（VRF）インスタンスでのみサポートされます。
- 出力ピア エンジニアリング（EPE）ピア セットには、任意の数の EPG ピアを追加できます。ただし、インストールされている復元力のある CE ごとの FEC は 32 ピアに制限されています。
- 特定の BGP ネイバーは、単一のピア セットのメンバーにしかなれません。ピアセットが構成されています。複数のピアセットはサポートされていません。オプションの**ピアセット**名を指定して、ネイバーをピアセットに追加できます。対応する RPC FEC は、ピアセット内のすべてのピア間でトラフィックを負荷分散します。ピアセット名は、最長 63 文字の文字列です（64 NULL で終了）。この長さは、NX-OS ポリシー名の長さと一致します。ピアは、単一のピアセットのメンバーにしかなれません。
- 特定のピアの隣接関係は、異なるピアセットに個別に割り当てることはできません。
- Cisco NX-OS リリース 9.3(3) 以降、BGP 出力ピア エンジニアリングは Cisco Nexus 9300-GX プラットフォーム スイッチでサポートされます。

BGP を使用したネイバー出力ピア エンジニアリングの設定

RFC 7752 および draft-ietf-idr-bgpls-segment-routing-epe の導入により、出力園児に名リングを設定できます。この機能は、外部 BGP ネイバーに対してのみ有効であり、デフォルトでは設定されていません。出力エンジニアリングでは、RFC 7752 エンコーディングを使用します。

始める前に

- BGPを有効にする必要があります。
- リリース 7.0(3)I3(1) またはリリース 7.0(3)I4(1) からアップグレードした後、Cisco Nexus 9000 シリーズ スイッチで出力ピア エンジニアリング（EPE）を設定する前に、次のコマンドを使用して、TCAM リージョンを設定します。
  1. switch# **hardware access-list tcam region vpc-convergence 0**
  2. switch# **hardware access-list tcam region racl 0**
  3. switch# **hardware access-list tcam region mpls 256 double-wide**
- 設定を保存して、スイッチをリロードします。

詳細については、*Cisco Nexus 9000 Series NX-OS Security Configuration Guide* の「Using Templates to Configure ACL TCAM Region Sizes」および「Configuring ACL TCAM Region Sizes」のセクション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

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

ステップ 4	[**no**|**default**] **egress-engineering** [**peer-set** *peer-set-name*]
	例：
	`switch(config)# router bgp 1`
	`switch(config-router)# neighbor 4.4.4.4`
	`switch(config-router)#`
	`egress-engineering peer-set NewPeer`
		ピアノード SID がネイバーに割り当てられ、BGP リンク状態（BGP-LS）アドレス ファミリ リンク NLRI のインスタンスでアドバタイズされるかどうかを指定します。ネイバーがマルチホップ ネイバーである場合、BGP-LS リンク NLRI インスタンスもネイバーへの等コストマルチパス（ECMP）パスごとにアドバタイズされます。これには、一意の Peer-Adj-SID が含まれます。
		オプションで、ネイバーをピア セットに追加できます。ピアセット SID は、ピアノード SID と同じインスタンスの BGP-LS リンク NLRI でもアドバタイズされます。BGP リンクステート NLRI は、リンクステート アドレス ファミリが設定されているすべてのネイバーにアドバタイズされます。
		EPE の詳細については、RFC 7752 および draft-ietf-idr-bgpls-segment-routing-epe-05 を参照してください。

出力ピア エンジニアリングの設定例

BGP スピーカー 1.1.1.1 の出力ピア エンジニアリングのサンプル設定を参照してください。ネイバー 20.20.20.20 は SDN コントローラであることに注意してください。

```
 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

interface mgmt0
  ip address dhcp
  vrf member management


interface loopback1
  ip address 1.1.1.1/32
line console

line vty
ip route 2.2.2.2/32 10.1.1.2
ip route 3.3.3.3/32 11.1.1.3
ip route 3.3.3.3/32  12.1.1.3
ip route 4.4.4.4/32  13.1.1.4
ip route 20.20.20.20/32 14.1.1.20

router bgp 1
  address-family ipv4 unicast
  address-family link-state
 neighbor 10.1.1.2
    remote-as 2
    address-family ipv4
```

```
    egress-engineering
 neighbor 3.3.3.3
   remote-as 3
   address-family ipv4
   update-source loopback1
   ebgp-multihop 2
   egress-engineering
 neighbor 4.4.4.4
   remote-as 4
   address-family ipv4
   update-source loopback1
   ebgp-multihop 2
   egress-engineering
neighbor 20.20.20.20
   remote-as 1
   address-family link-state
   update-source loopback1
   ebgp-multihop 2
neighbor 124.11.50.5
    bfs
    remote-as 6
    update-source port-channel50.11
    egress-engineering peer-set pset2 <<<<<<<
    address-family ipv4 unicast
neighbor 124.11.101.2
    bfd
    remote-as 6
    update-source Vlan2401
    egress-engineering
    address-family ipv4 unicast
```

次に、**show bgp internal epe** コマンドの出力例を示します。

```
switch# show bgp internal epe
BGP Egress Peer Engineering (EPE) Information:
Link-State Server: Inactive
Link-State Client: Active
Configured EPE Peers: 26
Active EPE Peers: 3
EPE SID State:
RPC SID Peer or Set Assigned
ID Type Set Name ID Label Adj-Info, iod
1 Node 124.1.50.5 1 1600
2 Set pset1 2 1601
3 Node 6.6.6.6 3 1602
4 Node 124.11.50.5 4 1603
5 Set pset2 5 1604
6 Adj 6.6.6.6 6 1605 124.11.50.4->124.11.50.5/0x1600b031, 80
7 Adj 6.6.6.6 7 1606 124.1.50.4->124.1.50.5/0x16000031, 78
EPE Peer-Sets:
IPv4 Peer-Set: pset1, RPC-Set 2, Count 7, SID 1601
Peers: 124.11.116.2 124.11.111.2 124.11.106.2 124.11.101.2
124.11.49.5 124.1.50.5 124.1.49.5
IPv4 Peer-Set: pset2, RPC-Set 5, Count 5, SID 1604
Peers: 124.11.117.2 124.11.112.2 124.11.107.2 124.11.102.2
124.11.50.5
IPv4 Peer-Set: pset3, RPC-Set 0, Count 4, SID unspecified
Peers: 124.11.118.2 124.11.113.2 124.11.108.2 124.11.103.2
IPv4 Peer-Set: pset4, RPC-Set 0, Count 4, SID unspecified
Peers: 124.11.119.2 124.11.114.2 124.11.109.2 124.11.104.2
IPv4 Peer-Set: pset5, RPC-Set 0, Count 4, SID unspecified
Peers: 124.11.120.2 124.11.115.2 124.11.110.2 124.11.105.2
switch#
```

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

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。

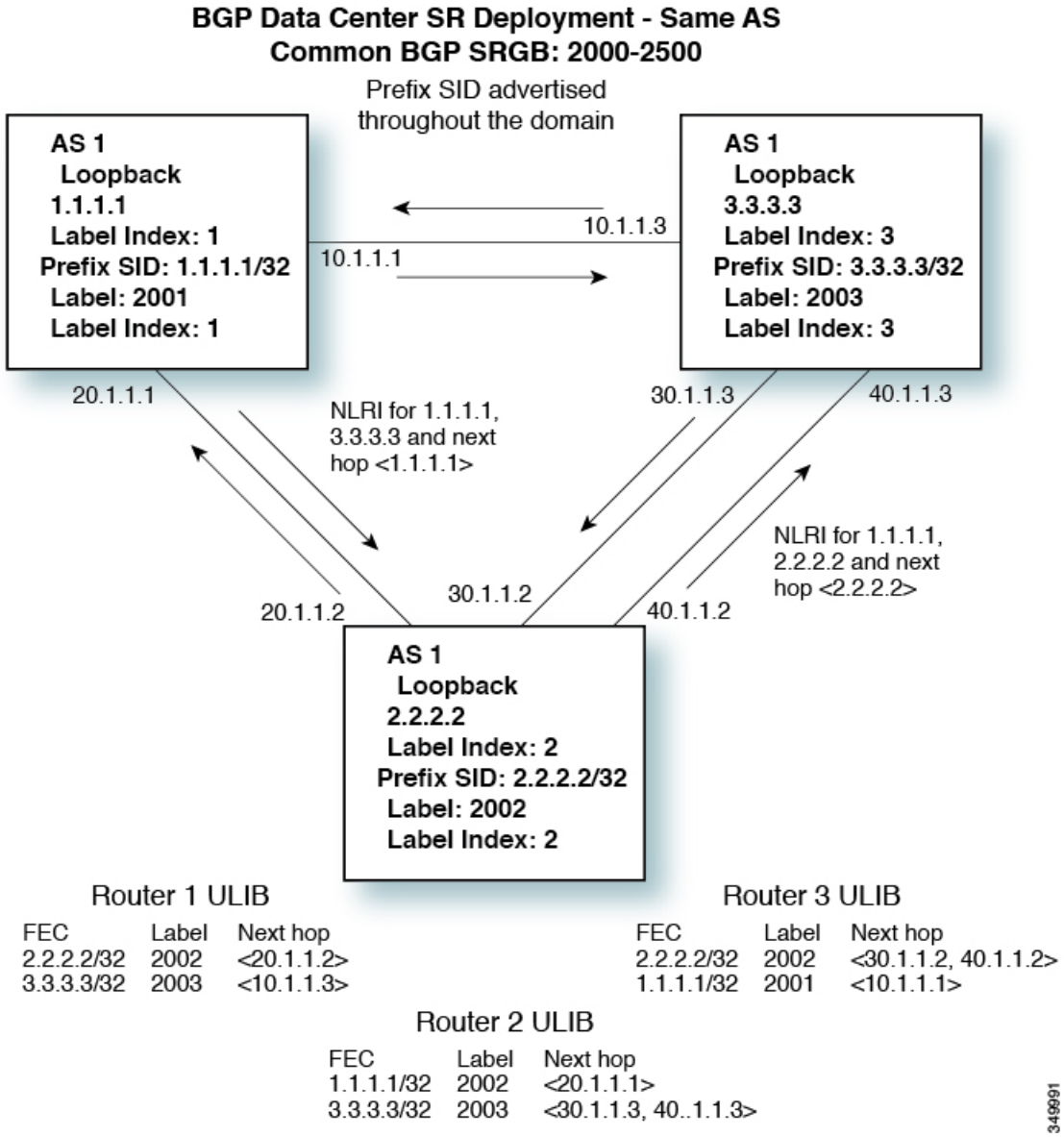
ステップ 5	[**no**] **address-family link-state**
	例：
	`switch(config)#router bgp 1`
	`switch(config-router)#address-family link-state`
	`switch(config-router)#neighbor 20.20.20.20`
	`switch(config-router)#address-family link-state`
		アドレスファミリ インターフェイス コンフィギュレーション モードを開始します。
		（注） このコマンドは、ネイバー アドレスファミリ 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

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 *14 : BGP* プレフィックス *SID* の簡単な例



セグメント ルーティング MPLS 上のレイヤ 2 EVPNの設定

レイヤ 2 EVPN について

イーサネット VPN（EVPN）は、MPLS ネットワークを介してイーサネット マルチポイント サービスを提供する次世代のソリューションです。EVPN は、コアでコントロールプレーン ベースの MAC ラーニングを可能にする既存の仮想プライベート LAN サービス（VPLS）とは対照的に動作します。EVPN では、EVPN インスタンスに参加している PE が MP-BGP プロトコルを使用してコントロールプレーン内でカスタマー MAC ルートを学習します。コントロールプレーン MAC 学習には数多くの利点があり、フローごとのロードバランシングによるマルチホーミングのサポートなどにより、VPLS の弱点に EVPN で対処できるようにします。

EVPN コントロール プレーンでは、データセンター ネットワークにおいて、次のものを提供します。

- データセンター ネットワークの物理トポロジに制限されない、柔軟なワークロード配置。そのため、データセンター ファブリック内の任意の場所に仮想マシン（VM）を配置できます。
- データセンター内部およびデータセンター間における最適なサーバー間 East-West トラフィック。サーバ/仮想マシン間の East-West トラフィックは、ファースト ホップ ルータでのほぼ特定されたルーティングで達成されます。ファースト ホップ ルーティングはアクセス レイヤで行われます。ホスト ルートの交換は、サーバまたはホストへの流入と送出に関するルーティングがほぼ特定されるようにする必要があります。VM モビリティは、新しい MAC アドレスまたは IP アドレスがローカル スイッチに直接接続されている場合に、新しいエンドポイント接続を検出することでサポートされます。ローカル スイッチは、新しい MAC または IP アドレスを検出すると、ネットワークの残りの部分に新しいロケーションを通知します。
- レイヤ 2 およびレイヤ 3 トラフィックのセグメンテーション。トラフィック セグメンテーションは MPLS カプセル化を使用して実現され、ラベル（BD ごとのラベルおよび VRF ごとのラベル）はセグメント識別子として機能します。

セグメント ルーティング MPLS 上のレイヤ 2 EVPN の注意事項と制限事項

セグメント ルーティング MPLS 上のレイヤ 2 EVPN には、次の注意事項と制限事項があります。

- セグメント ルーティング レイヤ 2 EVPN フラッディングは、入力レプリケーション メカニズムに基づいています。MPLS コアはマルチキャストをサポートしていません。
- ARP 抑制はサポートされていません。
- vPC での整合性チェックはサポートされていません。

- 同じレイヤ 2 EVI とレイヤ 3 EVI を一緒に設定することはできません。
- Cisco NX-OSリリース 9.3(1) 以降、レイヤ 2 EVPN は Cisco Nexus 9300-FX2 プラットフォーム スイッチでサポートされます。
- Cisco NX-OSリリース 9.3(5) 以降、セグメント ルーティング MPLS 上のレイヤ 2 EVPN は、Cisco Nexus 9300-GX および Cisco Nexus 9300-FX3 プラットフォーム スイッチでサポートされます。

セグメント ルーティング MPLS 上のレイヤ 2 EVPNの設定

始める前に

次の手順を実行します。

- **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。
- 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。
- **nv overlay** コマンドを使用して、nv オーバーレイ機能を有効にする必要があります。
- **nv overlay evpn** コマンドを使用して EVPN コントロール プレーン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	**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	**install feature-set mpls** 例： `switch(config)#install feature-set mpls`	MPLS 構成コマンドを有効にします。
ステップ 4	**feature-set mpls** 例： `switch(config)#install feature-set mpls`	MPLS 構成コマンドを有効にします。

	コマンドまたはアクション	目的
ステップ 5	**feature mpls segment-routing** 例： `switch(config)#feature mpls segment-routing`	セグメントルーティング構成コマンドを有効にします。
ステップ 6	**feature mpls evpn** 例： `switch(config)#feature mpls evpn`	EVPN over MPLS 構成コマンドを有効にします。このコマンドは **feature-nv** CLI コマンドとは相互に排他的です。
ステップ 7	**feature nv overlay** 例： `switch(config)#feature nv overlay`	セグメント ルーティング レイヤ 2 EVPN に使用される NVE 機能を有効にします。
ステップ 8	**nv overlay evpn** 例： `switch(config)#nv overlay evpn`	EVPN を有効にします。
ステップ 9	**interface loopback** *Interface_Number* 例： `switch(config)#interface loopback 1`	NVE のループバック インターフェイスを設定します。
ステップ 10	**ip address** *address* 例： `switch(config-if)#ip address 192.168.15.1`	IP アドレスを設定します。
ステップ 11	**exit** 例： `switch(config-if)#exit`	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 12	**evpn** 例： `switch(config)#evpn`	EVPN 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	**evi** *number*
	例：
	`switch(config-evpn)#evi 1000`
	`switch(config-evpn-sr)#`
		レイヤ 2 EVIを設定します。必要であれば、自動生成された EVI に基づいて RT を手動で構成できます。

ステップ 14	**encapsulation mpls** 例： `switch(config-evpn)#encapsulation mpls`	MPLSカプセル化と入力レプリケーションを有効にします。

	コマンドまたはアクション	目的
ステップ 15	**source-interface loopback** *Interface_Number* 例： `switch(config-evpn-nve-encap)#source-interface loopback 1`	NVE送信元インターフェイスを指定します。
ステップ 16	**exit** 例： `switch(config-evpn-nve-encap)#exit`	設定を終了します。
ステップ 17	**vrf context** *VRF_NAME* 例： `switch(config)#vrf context Tenant-A`	VRF を設定します。
ステップ 18	**evi** *EVI_ID* 例： `switch(config-vrf)#evi 30001`	L3 EVI を設定します。
ステップ 19	**exit** 例： `switch(config-vrf)#exit`	設定を終了します。
ステップ 20	**VLAN** *VLAN_ID* 例： `switch(config)#vlan 1001`	VLAN を設定します。
ステップ 21	**evi auto** 例： `switch(config-vlan)#evi auto`	L2 EVI を設定します。
ステップ 22	**exit** 例： `switch(config-vlan)#exit`	
ステップ 23	**router bgp** *autonomous-system-number* 例： `switch(config)#router bgp 1`	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 24	**address-family l2vpn evpn** 例： `switch(config-router)#address-family l2vpn evpn`	EVPN アドレス ファミリをグローバルに有効にします。

	コマンドまたはアクション	目的
ステップ 25	**neighbor address** *remote-as autonomous-system-number* 例： `switch(config-router)#neighbor 192.169.13.1 remote as 2`	BGP ネイバーを設定します。
ステップ 26	**address-family l2vpn evpn** 例： `switch(config-router-neighbor)#address-family l2vpn evpn`	ネイバーの EVPN アドレス ファミリを有効にします。
ステップ 27	**encapsulation mpls** 例： `switch(config-router-neighbor)#encapsulation mpls`	MPLS カプセル化を有効にします。
ステップ 28	**send-community extended** 例： `switch(config-router-neighbor)#send-community extended`	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 29	**vrf** *VRF_NAME* 例： `switch(config-router)#vrf Tenant-A`	BGP VRF を設定します。
ステップ 30	**exit** 例： `switch(config-router)#exit`	設定を終了します。

EVI 用の VLAN の設定

手順

	コマンドまたはアクション	目的
ステップ 1	**vlan** *number*	VLAN を設定します。
ステップ 2	**evi** *[auto]*	VLAN の BD ラベルを作成します。このラベルは、セグメント ルーティング レイヤ 2 EVPN 全体で VLAN の識別子として使用されます。

NVE インターフェイスの設定

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`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 loopback** *loopback_number* 例： `switch(config)# interface loopback 1`	IP アドレスをこのループバック インターフェイスに関連付け、この IP アドレスをセグメント ルーティング設定に使用します。
ステップ 3	**ip address** 例： `switch(config-if)#ip address 192.169.15.1/32`	IPv4 アドレス ファミリを指定し、ルータ 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	**evpn** 例： `switch(config)#evpn`	EVPN 設定モードを開始します。
ステップ 5	**encapsulation mpls** 例： `switch(config-evpn)# encapsulation mpls`	MPLS カプセル化と入力レプリケーションを有効にします。
ステップ 6	**source-interface** *loopback_number* 例： `switch(config-evpn-nve-encap)#source-interface loopback 1`	NVE 送信元インターフェイスを指定します。
ステップ 7	**exit** 例： `switch(config)# 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。

VRF 下での EVI の設定

手順

	コマンドまたはアクション	目的
ステップ 1	**vrf context** テナント	VRF テナントを作成します。

	コマンドまたはアクション	目的
ステップ 2	**evi** *number*	VRF 下でレイヤ 3 EVI を設定します。

エニーキャスト ゲートウェイの設定

ファブリック転送の設定は、SVIがエニーキャストモードで設定されている場合にのみ必要です。

手順

	コマンドまたはアクション	目的
ステップ 1	**fabric forwarding anycast-gateway-mac 0000.aabb.ccdd**	分散ゲートウェイの仮想MACアドレスを設定します。
ステップ 2	**fabric forwarding mode anycast-gateway**	インターフェイス コンフィギュレーション モードで SVI をエニーキャスト ゲートウェイと関連付けます。

ループバック インターフェイスのラベル付きパスのアドバタイズ

レイヤ2 EVPNエンドポイントとしてアドバタイズされるループバック インターフェイスは、ラベル インデックスにマッピングする必要があります。これにより、BGP は、同じものに対応する MPLS ラベル付きパスをアドバタイズします。

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**]**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。

	コマンドまたはアクション	目的
ステップ 4	**connected-prefix-sid-map** 例： `switch(config-sr-mpls)# connected-prefix-sid-map`	ローカル プレフィックスと SID のアドレス ファミリ固有のマッピングを設定できるサブモードを開始します。
ステップ 5	**address-family ipv4** 例： `switch(config-sr-mpls-conn)# address-family ipv4`	IPv4 アドレス プレフィックスを指定します。
ステップ 6	**1.1.1.1/32 index 100** 例： `switch(config-sr-mpls-conn-af)# 1.1.1.1/32 100`	SID 100 にアドレス 1.1.1.1/32 を関連付けます。
ステップ 7	**exit-address-family** 例： `switch(config-sr-mpls-conn-af)# exit-address-family`	アドレス ファミリを終了します。

SRv6 静的プレフィックス単位 TE について

SRv6 静的プレフィックス単位 TE 機能を使用すると、デフォルト以外の VRF にマッピングされたプレフィックスをマッピングおよびアドバタイズできます。この機能により、一致する VRF ルート ターゲットを使用して単一のインスタンスで複数のプレフィックスをアドバタイズでき、各プレフィックスを手動で入力する必要がなくなります。

Cisco NX-OS リリース 9.3(5) では、1 つの VNF だけが VM にサービスを提供できます。

SRv6 の静的なプレフィックスごとの TE の設定

始める前に

次の手順を実行します。

- **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。
- 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	**vrf context** *VRF_Name* 例： `switch(config)# vrf context vrf_2_7_8`	VRF 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。
ステップ 3	**rd** *rd_format* 例： `switch(config-vrf)# rd 2.2.2.0:2`	RD を VRF に割り当てます。
ステップ 4	**address-family** {**ipv4** | **ipv6** } 例： `switch(config-vrf)# address-family ipv4 unicast`	VRF インスタンス用に 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 5	**route-target import** *route-target-id* 例： `switch(config-vrf)# route-target import 1:2`	VRF へのルートのインポートを設定します。
ステップ 6	**route-target import** *route-target-id* **evpn** 例： `switch(config-vrf)# route-target import 1:2 evpn`	一致するルート ターゲット値を持つ、レイヤ3 EVPN から VRF へのルートのインポートを設定します。
ステップ 7	**route-target export** *route-target-id* 例： `switch(config-vrf)# route-target export 1:2`	VRF からのルートのエクスポートを設定します。
ステップ 8	**route-target export** *route-target-id* **evpn** 例： `switch(config-vrf)# route-target export 1:2 evpn`	一致するルート ターゲット値を持つ、VPN から レイヤ3 EVPN からへのルートのエクスポートを設定します。
ステップ 9	**router bgp** *autonomous-system-number* 例： `switch(config)# router bgp 65000`	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。

	コマンドまたはアクション	目的
ステップ 10	**router-id** *id* 例： `switch(config-router)# router-id 2.2.2.0`	ルータ ID を設定します。
ステップ 11	**address-family l2vpn evpn** 例： `switch(config-router-af)# address-family l2vpn evpn`	レイヤ 2 VPN EVPN のグローバル 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 12	**neighbor** *ipv4-address* **remote-as**
	例：
	`switch(config-router)# neighbor 7.7.7.0 remote-as 65000`
	`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 13	**update-source loopback** *number* 例： `switch(config-router-neighbor)# update-source loopback0`	ループバック番号を指定します
ステップ 14	**address-family l2vpn evpn** 例： `switch(config-router-neighbor)#address-family l2vpn evpn`	ネイバーの EVPN アドレス ファミリを有効にします。
ステップ 15	**send-community extended** 例： `switch(config-router-neighbor)#send-community extended`	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 16	**encapsulation mpls** 例： `switch(config-router-neighbor)#encapsulation mpls`	MPLS カプセル化を有効にします。
ステップ 17	**exit** 例： `switch(config-router-neighbor)#exit`	設定を終了します。

例

次の例は、VRF VT を定義するために RPM 構成を設定する方法を示しています。

```
rf context vrf_2_7_8
          rd 2.2.2.0:2
```

```
          address-family ipv4 unicast
            route-target import 0.0.1.1:2
            route-target import 0.0.1.1:2 evpn
            route-target export 0.0.1.1:2
            route-target export 0.0.1.1:2 evpn
ip extcommunity-list standard vrf_2_7_8-test permit rt 0.0.1.1:2
            route-map Node-2 permit 4
            match extcommunity vrf_2_7_8-test
            set extcommunity color 204
```

RD Auto について

自動派生ルート識別子（rd auto）は、IETF RFC 4364 セクション 4.2 で説明されているタイプ 1 エンコーディング形式に基づいています。https://tools.ietf.org/html/rfc4364#section-4.2タイプ 1 エンコーディングでは、4 バイトの管理フィールドと 2 バイトの番号フィールドを使用できます。Cisco NX-OS 内では、自動導出 RD は、4 バイトの管理フィールド（RID）としての BGP ルータ ID の IP アドレスと、2 バイトの番号フィールド（VRF ID）の内部 VRF ID を使用して構築されます。

2 バイトの番号付けフィールドは常に VRF から取得されますが、IP-VRF または MAC-VRF での使用に応じて異なる番号付け方式になります。

- IP-VRF の 2 バイトの番号付けフィールドは、1 から始まる内部 VRF ID を使用します。VRF ID 1 および 2 は、それぞれデフォルト VRF および管理 VRF 用に予約されています。最初のカスタム定義 IP VRF は VRF ID 3 を使用します。
- MAC-VRF の 2 バイトの番号付けフィールドは、VLAN ID + 32767 を使用します。その結果、VLAN ID 1 は 32768 になります。

例：自動取得ルート識別子（RD）

- BGP ルータ ID 192.0.2.1 および VRF ID 6-RD 192.0.2.1:6 の IP-VRF
- BGP ルータ ID 192.0.2.1 および VLAN 20-RD 192.0.2.1:32787 の MAC-VRF

Route-Target Auto について

自動派生Route-Target（route-target import/export/both auto）は、IETF RFC 4364 セクション 4.2（https://tools.ietf.org/html/rfc4364#section-4.2）で説明されているタイプ 0 エンコーディング形式に基づいています。IETF RFC 4364 セクション 4.2 ではルート識別子形式について説明し、IETF RFC 4364 セクション 4.3.1では、Route-Target に同様の形式を使用することが望ましいとしています。タイプ 0 エンコーディングでは、2 バイトの管理フィールドと 4 バイトの番号フィールドを使用できます。Cisco NX-OS 内では、自動派生 Route-Target は、2 バイトの管理フィールドとして自律システム番号（ASN）、4 バイトの番号フィールドのサービス識別子（EVI）で構成されます。

2 バイト ASN

タイプ 0 エンコーディングでは、2 バイトの管理フィールドと 4 バイトの番号フィールドを使用できます。Cisco NX-OS 内では、自動派生 Route-Target は、2 バイトの管理フィールドとし

ての自律システム番号（ASN）と、4 バイトの番号フィールドのサービス識別子（EVI）で構成されます。

自動派生 Route-Target（RT）の例：

- ASN 65001 と L3EVI 50001 内の IP-VRF： Route-Target 65001:50001
- ASN 65001 と L2VNI 30001 内の MAC-VRF：Route-Target 65001:30001

Multi-AS 環境では、Route-Target を静的に定義するか、Route-Target の ASN 部分と一致するように書き換える必要があります。

（注） 4 バイト ASN の自動派生 Route-Target はサポートされていません。

4 バイト ASN

タイプ 0 エンコーディングでは、2 バイトの管理フィールドと 4 バイトの番号フィールドを使用できます。Cisco NX-OS 内では、自動派生 Route-Target は、2 バイトの管理フィールドとしての自律システム番号（ASN）と、4 バイトの番号フィールドのサービス識別子（EVI）で構成されます。4 バイト長の ASN 要求と 24 ビット（3 バイト）を必要とする EVI では、拡張コミュニティ内のサブフィールド長が使い果たされます（2 バイト タイプと 6 バイト サブフィールド）。長さと形式の制約、およびサービス識別子（EVI）の一意性の重要性の結果、4 バイトの ASN は、IETF RFC 6793 セクション 9（https://tools.ietf.org/html/rfc6793#section-9）で説明されているように、AS_TRANS という名前の 2 バイトの ASN で表されます。2 バイトの ASN 23456 は、4 バイトの ASN をエイリアスする特別な目的の AS 番号であるAS_TRANSとして IANA（https://www.iana.org/assignments/iana-as-numbers-special-registry/iana-as-numbers-special-registry.xhtml）によって登録されます。

4 バイトの ASN（AS_TRANS）を使用した自動派生 Route-Target（RT）の例：

- ASN 65656 と L3VNI 50001 内の IP-VR：Route-Target 23456:50001
- ASN 65656 とL2VNI 30001 内の MAC-VRF：Route-Target 23456:30001

BD 用の RD およびルート ターゲットの設定

VLAN で evi auto を設定すると、ブリッジ ドメイン (BD) RD およびルート ターゲットが自動的に生成されます。BD RD およびルート ターゲットを手動で設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例：	グローバル 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	**evpn** 例： `switch(config)# evpn`	EVPN 設定モードを開始します。
ステップ 3	**evi** *VLAN_ID* 例： `switch(config-evpn)# evi 1001`	RD/ルート ターゲットを設定するための L2 EVI を指定します。
ステップ 4	**rd** *rd_format* 例： `switch(config-evpn-evi-sr)# rd 192.1.1.1:33768`	RD を設定します。
ステップ 5	**route-target both** *rt_format* 例： `switch(config-evpn-evi-sr)# route-target both 1:20001`	ルートターゲットを設定します。

VRF用 の RD およびルート ターゲットの設定

VRF で **evi** *evi_ID* を設定すると、VRF RD およびルート ターゲットが自動的に生成されます。VRF RD およびルート ターゲットを手動で設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`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 A`	VRF を設定します。
ステップ 3	**rd auto** または **rd_format** 例： `switch(config-vrf)# rd auto`	RD を設定します。
ステップ 4	**address-family ipv4 unicast** 例：	IPv4アドレスファミリを有効にします。

	コマンドまたはアクション	目的
	switch(config-vrf)# address-family ipv4 unicast	
ステップ 5	**route-target both** *rt_format* **evpn** 例： switch(config-vrf-af-ipv4)# route-target both 1:30001 evpn	ルートターゲットを設定します。

セグメント ルーティング MPLS 上のレイヤ 2 EVPN の設定例

次の例は、セグメント ルーティング MPLS を介したレイヤ 2 EVPN の設定を示しています。

```
install feature-set mpls
feature-set mpls
nv overlay evpn
feature bgp
feature mpls segment-routing
feature mpls evpn
feature interface-vlan
feature nv overlay

fabric forwarding anycast-gateway-mac 0000.1111.2222

vlan 1001
  evi auto

vrf context Tenant-A
  evi 30001

interface loopback 1
  ip address 192.168.15.1/32

interface vlan 1001
  no shutdown
  vrf member Tenant-A
  ip address 111.1.0.1/16
  fabric forwarding mode anycast-gateway

router bgp 1
  address-family l2vpn evpn
    neighbor 192.169.13.1
      remote-as 2
      address-family l2vpn evpn
        send-community extended
        encapsulation mpls
    vrf Tenant-A

evpn
  encapsulation mpls
    source-interface loopback 1
```

繰り返しの VPN ルートの SRTE の構成

デフォルト以外の VRF 内のルートが、デフォルト VRF 内のルート上で再帰する前に、同じ VRF 内の他のルート上で再帰するユースケースを想定します。さらに、これらのルートは EVPN タイプ 5 ルートとして BGP 経由でシグナリングされ、ルートのゲートウェイ IP フィールド（GW-IP）がネクストホップを指定します。これらのタイプのルートの SR トラフィック エンジニアリングをサポートするために、再帰 VPN ルートの SRTE 機能を使用すると、BGP はルートを再帰的に解決し、現在のルートのネクストホップを解決する次のルートを反復的に検索します。ネクスト ホップはデフォルト VRF にあります。このルートには、ルーティングに必要な VPN ラベルが必要であり、デフォルト VRF にあるネクストホップを使用して、SRTE ポリシーのエンドポイントを選択してトラフィックを誘導できるようになりました。

したがって、再帰 VPN ルートの SRTE 機能により、BGP はエンドポイントとして GW-IP を使用して SRTE からポリシーを要求できます。SRTE は一致するポリシーの BSID を返します。ただし、デフォルト VRF では、CO ポリシーがより適切な一致に置き換えられると、BSID が後で変更される可能性があります。

繰り返しの VPN ルートの SRTE の構成に関する注意事項および制限事項

Cisco NX-OS リリース 10.3(2)F 以降では、再帰 VPN ルート機能の SRTE がサポートされています。

次に、この機能に関するガイドラインおよび制限事項を示します。

- この機能は、Cisco Nexus 9300-EX、9300-FX、9300-FX2、9300-GX、および N9K-C9332D-GX2B プラットフォーム スイッチでサポートされています。
- この機能は、ネクストホップとしてゲートウェイ IP を持つタイプ 5 EVPN ルートでのみサポートされます。デフォルト VRF の再帰ルートではサポートされません。
- IPv4 ルートのみがサポートされます。
- ネクストホップが同じ VRF 内の別のルートである VRF 内のルートのプレフィックス長は 32 ビット（ホスト ルート）である必要があります。
- 複数の IPv4 ユニキャスト非デフォルト VRF への EVPN 再帰 VPN ルートのルートリークまたはインポートは許可されません。
- カラーのみのルートはサポートされていません。
- ルート インジェクタをネットワーク内のトラフィック ベアリング リーフの 1 つに統合することは推奨されません。

繰り返しの VPN ルートの SRTE 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	**router bgp** *number*
	例：
	`switch(config)# router bgp 100`
	`switch(config-router)#`
		BGP を設定します。

ステップ 3	**vrf** *VRF_Name*
	例：
	`switch(config-router)# vrf vrf3`
	`switch(config-router)#`
		ルートマップを vrf コンテキストに適用します。

ステップ 4	**address-family ipv4 unicast**
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router)#`
		IPv4 のアドレスファミリを設定します。

ステップ 5	**export-gateway-ip**
	例：
	`switch(config-router)# export-gateway-ip`
	`switch(config-router)#`
		gateway-ip をエクスポートしてアドバタイズして、EVPN タイプ 5 ルートを再接続します。
		（注） gateway-ip のエクスポートと EVPN ゲートウェイ構成の設定は同時に実行できます。同時に設定すると、すべてのプレフィックスがゲートウェイ IP とともにエクスポートされます。

ステップ 6	**address-family l2vpn evpn**
	例：
	`switch(config-router)# address-family l2vpn evpn`
	`switch(config-router)#`
		L2VPN EVPN のアドレス ファミリを構成します。

ステップ 7	**route-map** *map-name* **out** 例：	発信ルートに設定された BGP ポリシーを適用します。

	コマンドまたはアクション	目的
	`switch(config-router)# route-map setrrrnh out` `switch(config-route-map)#`	

ステップ 8	**route-map** *map-name* [**permit** | **deny**] [*seq*]
	例：
	`switch(config-route-map)# route-map ABC permit 10`
	`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 マップ設定モードを開始します。

ステップ 9	**set extcommunity color** *color-num*
	例：
	`switch(config-route-map)# set extcommunity color 20`
	`switch(config-route-map)`
		カラー拡張コミュニティの BGP 外部コミュニティ属性を設定します。

繰り返しの VPN ルートの SRTE の構成例

```
switch# configure terminal
switch(config)# router bgp 100
switch(config-route-map)# vrf vrf3
switch(config-router)# address-family ipv4 unicast
switch(config-router)# export-gateway-ip
switch(config-router)# l2vpn evpn
switch(config-router)# route-map setrrnh out
switch(config-router)# route-map ABC permit 10
switch(config-route-map)# set extcommunity color 20
```

繰り返しの VPN ルートの SRTE の構成確認

繰り返しの VPN ルートの SRTE 構成に関する情報を表示するには、以下のタスクのいずれかを実行します：

表 *13 :* 繰り返しの *VPN* ルートの *SRTE* の構成確認

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。

コマンド	目的
show running-config rpm	ルート ポリシー マネージャ（RPM）についての情報を表示します。
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 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、ポリシー名の自動入力機能があります。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自動入力機能があります。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 ip route vrf <*vrf-name*>	VRF のルーティング情報を表示します。

セグメントルーティングのVNFの比例マルチパスの設定

セグメント ルーティングの VNF の比例マルチパスについて

ネットワーク機能仮想化インフラストラクチャ（NFVi）では、サービス ネットワーク（ポータブル IP）が仮想ネットワーク機能（VNF）によりアドバタイズされます。VNF は、ポータブル IP ゲートウェイ (PIP-GW) とも呼ばれ、VNF 内の VM 間でデータ パケットをルーティングします。セグメント ルーティング機能の VNF の比例マルチパスにより、EVPN アドレス ファミリでサービス ネットワーク（PIP）の VNF をアドバタイズできます。VNF の IP アドレスは、サービス ネットワークの EVPN IP プレフィックス ルート NLRI アドバタイズメントの「ゲートウェイ IP アドレス」フィールドでエンコードされます。

VNF の IP アドレスをアドバタイズすることにより、EVPN ファブリックの入力ノードは、VNF IP アドレスを VNF に接続されたリーフに再帰的に解決します。リーフは、サービス ネットワーク（PIP）をアドバタイズするのと同じノードである可能性があります。

ルートインジェクタは、IPv4 または IPv6 AF にルートを挿入する BGP プロトコルです。この場合、ルートインジェクタは、ネクスト ホップが VNF として設定されている VM にルートを挿入します。

ルート インジェクタとは異なり、VNF はルーティング プロトコルに参加して、VM の到達可能性をアドバタイズできます。サポートされているプロトコルは、eBGP、IS-IS、および OSPF です。

セグメント ルーティングの VNF の比例マルチパスの有効化

セグメント ルーティング機能の VNF の比例マルチパスを有効にして、ネクストホップ パスを保持することにより、IGP または静的ルートのルートを再配布できます。その後、再構築された EVPN タイプ 5 ルートのゲートウェイ IP をエクスポートしてアドバタイズできます。

Cisco NX-OS リリース 9.3(5) では、1 つの VNF だけが VM にサービスを提供できます。

始める前に

次の手順を実行します。

- **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します。
- MPLS セグメント ルーティング機能を有効化します。

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 モードに入ります。

ステップ 2	**route-map** *export-l2evpn-rtmap* **permit** *10* 例： `switch(config)# route-map export-l2evpn-rtmap permit 10`	<<説明が必要>>
ステップ 3	**match ip address prefix-list** *pip-pfx-list* 例： `switch(config-route-map)# match ip prefix-list vm-pfx-list`	PIP-GW をゲートウェイとしてアドバタイズする必要があるプレフィックスを定義します。
ステップ 4	**set evpn gateway-ip use-nexthop** 例： `switch(config-route-map)# set evpn gateway-ip use-nexthop`	gateway-ip をアドバタイズするための特定のルートを定義します。

ステップ 5	**vrf context** *VRF_Name*
	例：
	`switch(config-route-map)# vrf context vrf`
	`switch(config-route-map)# address-family ipv4 unicast`
	`switch(config-route-map)# export map export-l2evpn-rtmap`
		ルート マップを vrf コンテキストに適用します。

ステップ 6	**address-family ipv4 unicast**
	例：
	`switch(config-route-map)# address-family ipv4 unicast`
	`switch(config-route-map)# export map export-l2evpn-rtmap`
		ルート マップを vrf コンテキストに適用します。

ステップ 7	**export map** *export-l2evpn-rtmap* 例： `switch(config-route-map)# export map export-l2evpn-rtmap`	ルート マップを vrf コンテキストに適用します。
ステップ 8	**router bgp** *number* 例：	BGP を設定します。

	コマンドまたはアクション	目的
	`switch(config)# router bgp 100`	
ステップ 9	**vrf** *VRF_Name* 例： `switch(config-route-map)# vrf vrf3`	ルート マップを vrf コンテキストに適用します。
ステップ 10	**address-family ipv4 unicast** 例： `switch(config-router)# address-family ipv4 unicast`	IPv4 のアドレス ファミリを設定します。
ステップ 11	**export-gateway-ip** 例： `switch(config-route-map)# export-gateway-ip`	gateway-ip をエクスポートしてアドバタイズして、EVPN タイプ 5 ルートを再接続します。 （注） gateway-ip のエクスポートと EVPN ゲートウェイ構成の設定は同時に実行できます。同時に設定すると、すべてのプレフィックスがゲートウェイ IP とともにエクスポートされます。

vPC マルチホーミング

マルチホーミングについて

Cisco Nexus プラットフォーム スイッチは、vPC ベースのマルチホーミングをサポートします。このマルチホーミングでは、スイッチのペアが冗長性のために単一のデバイスとして機能し、両方のスイッチがアクティブ モードで機能します。EVPN 環境の Cisco Nexus プラットフォーム スイッチでは、レイヤ 2 マルチホーミングをサポートする 2 つのソリューションがあります。これらのソリューションは、MCT リンクが必要な従来の vPC（エミュレートまたは仮想 IP アドレス）と BGP EVPN 技術に基づいています。

BGP EVPN コントロール プレーンを使用している間、各 vPC ペアは共通の仮想 IP（VIP）を使用して、アクティブ/アクティブの冗長性を提供します。さらに、BGP EVPN ベースのマルチホーミングは、特定の障害シナリオで高速コンバージェンスを提供します。

vPC ピア上の BD ごとのラベル

vPC ピアが同じ BD ごとのラベルを持つようにするには、BD ごとのラベルに次の値を指定する必要があります。

```
Label value = Label_base + VLAN_ID
```

ラベル ベースは、同じ vPC ピアで設定されます。現在、VLAN 設定は両方の vPC ピアで同一であるため、両方の vPC ピアに同じラベルが付けられます。

Cisco NX-OS リリース 9.3(1) では、BD ごとのラベルの設定はサポートされていません。このリリースでは、evi auto のみがサポートされています。

vPC ピア上の VRF ごとのラベル

vPC ピアが同じ VRF ごとのラベルを持つようにするには、VRF ごとのラベルに次の値を指定する必要があります。

```
Label value = Label_base + vrf_allocate_index
```

vPC ピアの割り当てインデックスを設定するには、次の手順を実行します。

```
Router bgp 1
  vrf Tenant_A
    allocate-index 11
```

バックアップ リンクの設定

バックアップ リンクは、vPC ピア間で設定する必要があります。このリンクとしては、MCT に並列な任意のレイヤ 3 リンクが可能です。

例

```
interface vlan 100
    ip add 10.1.1.1/24
        mpls ip forwarding

< enable underlay protocol >
```

vPC マルチホーミング ピアリングの注意事項と制約事項

vPC マルチホーミング ピアリングには、次の注意事項と制約事項があります。

- ESI ベースのマルチホーミングはサポートされていません。
- 物理および仮想セカンダリ IP アドレスは、両方とも MPLS ラベル付きパスを介してアドバタイズされる必要があります。
- vPC 整合性チェックは、BD ごとのラベル設定ではサポートされていません。

vPC マルチホーミングの設定例

次の例は、vPC マルチホーミングの設定を示しています。

- vPC プライマリ

```
interface loopback1
  ip address 192.169.15.1/32
  ip address 192.169.15.15/32 secondary

evpn
```

```
  encapsulation mpls
    source-interface loopback1

vlan 101
  evi auto

vrf context A
  evi 301

router bgp 1
  vrf A
    allocate-index 1001
```

- vPC セカンダリ

```
interface loopback1
  ip address 192.169.15.2/32
  ip address 192.169.15.15/32 secondary

evpn
  encapsulation mpls
    source-interface loopback1

vlan 101
  evi auto

vrf context A
  evi 301

router bgp 1
  vrf A
    allocate-index 1001
```

セグメント ルーティング MPLS を介したレイヤ 3 EVPN およびレイヤ 3 VPN の構成

このセクションでは、レイヤ 3 EVPN を設定するタスクと、L3 EVPN および L3VPN ルータのスティッチングについて説明します。構成を完了するには、次の作業を実行します。

インポートおよびエクスポート ルール用の VRF およびルート ターゲットの設定

手順

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

	コマンドまたはアクション	目的
ステップ 2	**vrf** *vrf-name*	VPN ルーティングおよび転送（VRF）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。
ステップ 3	**rd auto**	一意のルート識別子（RD）をVRFに自動的に割り当てます。
ステップ 4	**address-family** { **ipv4 | ipv6** } **unicast**	VRF インスタンス用に 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 サブモードを開始します。
ステップ 5	**route-target import** *route-target-id*	一致するルートターゲット値を持つ、L3 VPN BGP NLRI から VRF へのルートのインポートを設定します。
ステップ 6	**route-target export** *route-target-id*	VRF から L3VPN BGP NLRI へのルートのエクスポートを設定し、指定されたルートターゲット識別子を L3VPN BGP NLRI に割り当てます。
ステップ 7	**route-target import** *route-target-id* **evpn**	一致するルートターゲット値を持つ L3 EVPN BGP NLRI からのルートのインポートを設定します。
ステップ 8	**route-target export** *route-target-id* **evpn**	VRF から L3 EVPN BGP NLRI へのルートのエクスポートを設定し、指定されたルートターゲット識別子を BGP EVPN NLRI に割り当てます。

BGP EVPN およびラベル割り当てモードの設定

encapsulation mpls コマンドを使用して MPLS トンネル カプセル化を使用できます。EVPN アドレス ファミリのラベル割り当てモードを設定できます。NX-OS の IP ルート タイプの EVPN でのデフォルトのトンネル カプセル化は VXLAN です。

BGP EVPN を介した Cisco Nexus 9000 シリーズ スイッチからの（IP またはラベル）バインディングのアドバタイズにより、リモート スイッチはルーティングされたトラフィックをその IP に送信できます。その際、MPLS を介して IP をアドバタイズしたスイッチへの IP のラベルを使用します。

IP プレフィックス ルート（タイプ 5）は次のとおりです。

- MPLS カプセル化によるタイプ 5 ルート

```
RT-5 Route - IP Prefix
```

```
RD:  L3 RD
IP Length:  prefix length
IP address:  IP (4 bytes)
Label1:  BGP MPLS Label
Route Target
RT for IP-VRF
```

デフォルトのラベル割り当てモードは、MPLS 上のレイヤ 3 EVPN の VRF 単位です。

BGP EVPN とラベル割り当てモードを設定するには、次の手順を実行します。

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。

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**	グローバル 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。
		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 l2vpn evpn**
	例：
	`switch(config-router)# address-family l2vpn evpn`
	`switch(config-router-af)#`
		レイヤ 2 VPN EVPN のグローバル 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	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 5	**neighbor** *ipv4-address* **remote-as** *autonomous-system-number* **例**：	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。

	コマンドまたはアクション	目的
	`switch(config-router)# neighbor 10.1.1.1 remote-as 64497` `switch(config-router-neighbor)#`	

ステップ 6	**address-family l2vpn evpn**
	例：
	`switch(config-router-neighbor)# address-family l2vpn evpn`
	`switch(config-router-neighbor-af)#`
		ラベル付きのレイヤ 2 VPN EVPN をアドバタイズします。

ステップ 7	**encapsulation mpls**
	例：
	`router bgp 100`
	`address-family l2vpn evpn`
	`neighbor NVE2 remote-as 100`
	`address-family l2vpn evpn`
	`send-community extended`
	`encapsulation mpls`
	`vrf foo`
	`address-family ipv4 unicast`
	`advertise l2vpn evpn`
	BGP セグメント ルーティング設定：
	`router bgp 100`
	`address-family ipv4 unicast`
	`network 200.0.0.1/32 route-map label_index_pol_100`
	`network 192.168.5.1/32 route-map label_index_pol_101`
	`network 101.0.0.0/24 route-map label_index_pol_103`
	`allocate-label all`
	`neighbor 192.168.5.6 remote-as 20`
	`address-family ipv4 labeled-unicast`
	`send-community extended`
		BGP EVPN アドレス ファミリを有効にし、EVPN タイプ 5 ルート アップデートをネイバーに送信します。
		（注） NX-OS の IP ルート タイプの EVPN でのデフォルトのトンネル カプセル化は VXLAN です。これをオーバーライドするために、MPLS トンネルのカプセル化を示す新しい CLI が導入されています。

ステップ 8	**vrf** <*customer_name*>	VRF を設定します。
ステップ 9	**address-family ipv4 unicast**	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 10	**advertise l2vpn evpn**	レイヤ 2 VPN EVPN をアドバタイズします。
ステップ 11	**redistribute direct route-map DIRECT_TO_BGP**	直接接続されたルートを BGP-EVPN に再配布します。
ステップ 12	**label-allocation-mode per-vrf**	ラベル割り当てモードを VRF 単位に設定します。プレフィックス単位のラベ

	コマンドまたはアクション	目的
		ル モードを設定する場合は、**no label-allocation-mode per-vrf** CLI コマンドを使用します。 EVPN アドレス ファミリの場合、デフォルトのラベル割り当ては VRF 単位です。一方、ラベル割り当て CLI がサポートされている他のアドレス ファミリではプレフィックス単位モードです。実行コンフィギュレーションでは、CLI の no 形式は表示されません。

例

プレフィックス単位のラベル割り当ての設定については、次の例を参照してください。

```
router bgp 65000
    [address-family l2vpn evpn]
    neighbor 10.1.1.1
        remote-as 100
        address-family l2vpn evpn
        send-community extended
    neighbor 20.1.1.1
        remote-as 65000
        address-family l2vpn evpn
        encapsulation mpls
        send-community extended
    vrf customer1
        address-family ipv4 unicast
            advertise l2vpn evpn
            redistribute direct route-map DIRECT_TO_BGP
            no label-allocation-mode per-vrf
```

BGP レイヤ 3 EVPN およびレイヤ 3 VPN スティッチングの構成

同じルーターでスティッチングを構成するには、レイヤー 3 VPN ネイバー関係とルーター アドバタイズメントを構成します。

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

	コマンドまたはアクション	目的

ステップ 2	[**no**] **router bgp** *autonomous-system-number*
	例：
	`switch# configure terminal`
	`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。
		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	**address-family** {**vpnv4** | **vpnv6**} **unicast**
	例：
	`switch(config-router)# address-family vpnv4 unicast`
	`switch(config-router-af)# address-family vpnv6 unicast`
	`switch(config-router-af)#`
		レイヤ 3 VPNv4 または VPNv6 に対するグローバル 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	**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 5	**neighbor** *ipv4-address* **remote-as** *autonomous-system-number* 例： `switch(config-router)# neighbor 20.1.1.1 remote-as 64498`	リモート BGP L3VPN ピアの IPv4 アドレスおよび AS 番号を設定します。

ステップ 6	**address-family** {**vpnv4** | **vpnv6**} **unicast**
	例：
	`switch(config-router)# address-family vpnv4 unicast`
	`switch(config-router-af)# address-family vpnv6 unicast`
	`switch(config-router-af)#`
		VPNv4 または VPNv6 のアドレスファミリのネイバーを設定します。

ステップ 7	**send-community extended**	BGP VPN アドレス ファミリを有効にします
ステップ 8	**import l2vpn evpn reoriginate**	標準のルート ターゲット識別子と一致するルート ターゲット識別子を持つレイヤ 3 BGP EVPN NLRI からのルーティング情報のインポートを設定し、このルーティング情報を、スティッチング ルート ターゲット識別子に割り当てる

	コマンドまたはアクション	目的
		再発信の後に、BGP EVPN ネイバーへエクスポートします。

ステップ 9	**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 レイヤ 3 EVPN 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 10	**address-family** {**l2vpn** | **evpn**
	例：
	`switch(config-router-neighbor)# address-family l2vpn evpn`
	`switch(config-router-neighbor-af)#`
		レイヤ 3 EVPN のネイバー アドレス ファミリを設定します。

ステップ 11	**import vpn unicast reoriginate**	スティッチング ルート ターゲット識別子と一致するルートターゲット識別子を持つ BGP EVPN NLRI からのルーティング情報のインポートを有効にし、この再発信後のルーティング情報をレイヤ 3 VPN BGP ネイバーにエクスポートします。
ステップ 12	**vrf** <*customer_name*>	VRF を設定します。
ステップ 13	**address-family ipv4 unicast**	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 14	**advertise l2vpn evpn**	レイヤ 2 VPN EVPN をアドバタイズします。

例

```
vrf context Customer1
    rd auto
    address-family ipv4 unicast
        route-target import 100:100
        route-target export  100:100
        route-target import 100:100 evpn
        route-target export 100:100 evpn

segment-routing
    mpls
      global-block 11000 20000
      connected-prefix-sid
         address-family ipv4 unicast
         200.0.0.1 index 101
!
```

```
int lo1
  ip address 200.0.0.1/32
!
interface e1/13
  description “MPLS interface towards Core”
  ip address 192.168.5.1/24
  mpls ip forwarding
  no shut

router bgp 100
address-family ipv4 unicast
allocate-label all
address-family ipv6 unicast
address-family l2vpn evpn
address-family vpnv4 unicast
address-family vpnv6 unicast
 neighbor 10.0.0.1 remote-as 200
    update-source loopback1
    address-family vpnv4 unicast
      send-community extended
      import l2vpn evpn reoriginate
    address-family vpnv6 unicast
      import l2vpn evpn reoriginate
      send-community extended
  neighbor 20.0.0.1 remote-as 300
    address-family l2vpn evpn
      send-community extended
      import vpn unicast reoriginate
      encapsulation mpls
  neighbor 192.168.5.6 remote-as 300
      address-family ipv4 labeled-unicast
  vrf Customer1
    address-family ipv4 unicast
      advertise l2vpn evpn
    address-family ipv6 unicast
      advertise l2vpn evpn
```

レイヤー 3 EVPN およびレイヤー 3 VPN を有効にする機能の設定

始める前に

VPN ファブリック ライセンスをインストールします。

feature interface-vlan コマンドが有効になってい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	**feature bgp**	BGP 機能と構成を有効にします。
ステップ 2	**install feature-set mpls**	MPLS 構成コマンドを有効にします。
ステップ 3	**feature-set mpls**	MPLS 構成コマンドを有効にします。

	コマンドまたはアクション	目的
ステップ 4	**feature mpls segment-routing**	セグメント ルーティング構成コマンドを有効にします。
ステップ 5	**feature mpls evpn**	EVPN over MPLS 構成コマンドを有効にします。このコマンドは **feature-nv** CLI コマンドとは相互に排他的です。
ステップ 6	**feature mpls l3vpn**	EVPN over MPLS 構成コマンドを有効にします。このコマンドは **feature-nv** CLI コマンドとは相互に排他的です。

セグメント ルーティングを介した BGP L3 VPN の構成

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。

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。

feature mpls l3vpn コマンドを使用して、MPLS L3 VPN 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。
		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	**address-family** {**vpnv4** | **vpnv6**} **unicast** 例： `switch(config-router)# address-family vpnv4 unicast`	レイヤ 3 VPNv4 または VPNv6 に対するグローバル 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
	`switch(config-router-af)# address-family vpnv6 unicast switch(config-router-af)#`	
ステップ 4	[**no**] **allocate-label option-b**	AS 間オプション b を無効にします
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 20.1.1.1 remote-as 64498 switch(config-router-neighbor)#`	リモート BGP L3VPN 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** {**vpnv4** | **vpnv6** } **unicast** 例： `switch(config-router-neighbor)# address-family vpnv4 unicast switch(config-router-neighbor-af)#`	VPNv4 または VPNv6 のアドレスファミリのネイバーを設定します。
ステップ 8	**send-community extended**	BGP VPN アドレス ファミリを有効にします。
ステップ 9	**vrf** <*customer_name*>	VRF を設定します。
ステップ 10	**allocate-index x**	割り当てインデックスを設定します。
ステップ 11	**address-family ipv4 unicast**	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 12	**redistribute direct route-map DIRECT_TO_BGP**	直接接続されたルートを BGP-L3VPN に再配布します。

SRTE 経由 BGP レイヤ 3 VPN

この機能により、データセンター相互接続（DCI）/WAN エッジ展開のセグメント ルーティング コアに対するトラフィック エンジニアリング機能が有効になります。DCI ハンドオフ（SR に基づき VxLAN から L3VPN へ、またはその逆）を可能にし、SR コアで SRTE 機能を使用できるため、さまざまなトラフィック クラスによって SLA を達成できます。SRTE 機能は、L3VPN プレフィックスに SR-Policy を適用することにより、DCI またはエッジ ルータに適用できます。L3VPN プレフィックスは、拡張コミュニティ カラーを設定した後（DCI またはエッジ ノードによって）アドバタイズでき、BGP L3VPN ネイバーは、そのカラーに基づいて SR

ポリシーを適用して SRTE を作成できます。以下に、L3VPN プレフィックスで拡張コミュニティ カラーを構成するための構成を示します。

SRTE を介したレイヤ 3 VPN の構成に関する注意事項と制限事項

Cisco NX-OS リリース 10.1(2) 以降、セグメント ルーティング トラフィック エンジニアリングは、Cisco Nexus 9300-FX3、N9K-C9316D-GX、N9K-C93180YC-FX、N9K-C93240YC-FX2、および N9K- C9364C プラットフォーム スイッチ上でレイヤ 4 VPN を介してサポートされます。

この機能の制限は次のとおりです。

- アンダーレイ IPv6 はサポートされません。SRv6 は代替です。
- BGP の専用ファブリックにおける PCE の欠点のため、BGP アンダーレイを使用した PCE はサポートされていません。
- NXOS が BGP-LS で LSA をアドバタイズできないため、PCE を使用した OSPF-SRTE はサポートされていません。
- 合計 1000 の SRTE ポリシー スケール、BGP VPNv4 32K ルート、BGP VPNV6 32k ルート、および 1000 のアンダーレイ SR プレフィックスをサポートします。

Cisco NX-OS リリース 10.2(3)F 以降、カラー専用（CO）ビットのオプションがルート マップに追加されています。SRTE ポリシーを使用している特定のプレフィックスの CO ビットの値が変更された場合、BGP は古いポリシーを削除し、新しいポリシーを追加します。

拡張コミュニティ カラーの構成

このセクションは、次のトピックで構成されています。

入力ノードにおける拡張コミュニティ カラーの構成

SRTE ポリシーがインスタンス化される入力ノードによってプレフィックスが通知されるときに、入力ノードで拡張コミュニティ カラーを構成するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的

ステップ 1	**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 ABC`
	`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。

	コマンドまたはアクション	目的

ステップ 3	**set extcommunity color** *color-num*
	例：
	`switch(config-route-map)# set extcommunity color 20`
	`switch(config-route-map)#`
		カラー拡張コミュニティの BGP 外部コミュニティ属性を設定します。

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	[**no**] **router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp1`
	`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。
		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 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 vpnv4/vpnv6 unicast**
	例：
	`switch(config-router-neighbor)# address-family vpnv4/vpnv6 unicast`
	`switch(config-router-neighbor-af)#`
		vpnv4/vpnv6 アドレス ファミリ タイプのルータ アドレスファミリ構成モードを開始します。

ステップ 8	**route-map** *map-name* **in**
	例：
	`switch(config-router-neighbor-af)# route-map ABC in`
	`switch(config-router-neighbor-af)#`
		構成された BGP ポリシーを受信ルートに適用します。
		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。

出力ノードでの拡張コミュニティ カラーの構成

プレフィックスが出力ノードによって通知されるときに、出力ノードで拡張コミュニティ カラーを構成するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的

ステップ 1	**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 ABC`
	`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	**set extcommunity color** *color-num*
	例：
	`switch(config-route-map)# set extcommunity color 20`
	`switch(config-route-map)#`
		カラー拡張コミュニティの BGP 外部コミュニティ属性を設定します。

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	[**no**] **router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp1`
	`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。
		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 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 vpnv4/vpnv6 unicast**
	例：
	`switch(config-router-neighbor)# address-family vpnv4/vpnv6 unicast`
	`switch(config-router-neighbor-af)#`
		vpnv4/vpnv6 アドレス ファミリ タイプのルータ アドレスファミリ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 8	**route-map** *map-name* **out**
	例：
	`switch(config-router-neighbor-af)# route-map ABC out`
	`switch(config-router-neighbor-af)#`
		発信ルートに設定された BGP ポリシーを適用します。
		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。

出力ノードでのネットワーク/再配布コマンドの拡張コミュニティカラー構成

プレフィックスが出力ノードによって通知されるときに、出力ノードで network/redistribute コマンドの拡張コミュニティ カラーを構成するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的

ステップ 1	**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 ABC`
	`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	**set extcommunity color** *color-num*
	例：
	`switch(config-route-map)# set extcommunity color 20`
	`switch(config-route-map)#`
		カラー拡張コミュニティの BGP 外部コミュニティ属性を設定します。

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	[**no**] **router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp1;`
	`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。
		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。

	コマンドまたはアクション	目的
ステップ 6	**vrf** *<customer_name>*	VRF を設定します。
ステップ 7	**address-family ipv4 unicast** 例： `switch(config-router-vrf)# address-family ipv4 unicast switch(config-router-af)#`	VRF インスタンスの IPv4 アドレス ファミリを指定し、アドレス ファミリ構成モードを開始します。
ステップ 8	**redistribute static route-map** *map-name* **out** 例： `switch(config-router-vrf-af)# redistribute static route-map ABC switch(config-router-af)#`	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 9	**network** *ip-prefix* [**route-map** *map-name*] 例： `switch(config-router-vrf-af)# network 1.1.1.1/32 route-map ABC switch(config-router-af-network)#`	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。

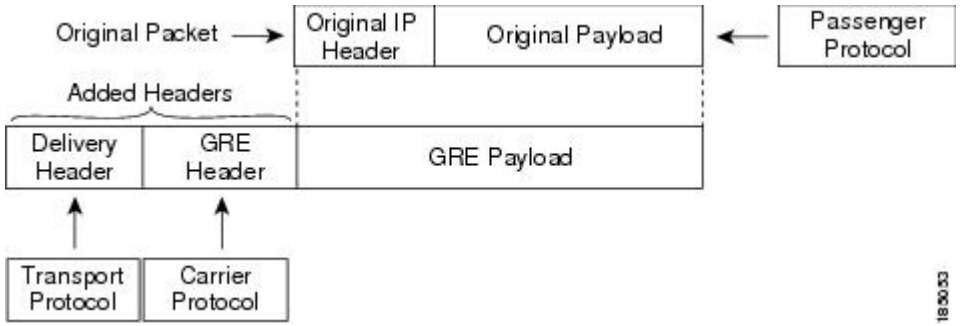
セグメント ルーティング MPLS および GRE トンネルの設定

GRE トンネル

Generic Routing Encapsulation（GRE）をさまざまなパッセンジャ プロトコルのキャリア プロトコルとして使用できます。

この次図は、GRE トンネルの IP トンネルのコンポーネントを示しています。オリジナルのパッセンジャ プロトコル パケットは GRE ペイロードとなり、デバイスはパケットに GRE ヘッダーを追加します。次にデバイスはトランスポート プロトコル ヘッダーをパケットに追加して送信します。

図 *15 : GRE PDU*



セグメント ルーティング MPLS および GRE

Cisco NX-OS リリース 9.3(1) 以降、Cisco Nexus デバイスではセグメント ルーティング MPLS とジェネリック ルーティング カプセル化(GRE)の両方を設定できます。これらのテクノロジーは両方ともシームレスに動作します。MPLS トンネルの終了後には、すべての MPLS トラフィックを GRE トンネルに転送できます。同様に、GRE の終了後には、GRE トンネルからのすべてのトラフィックを MPLS クラウドに転送できます。

すべての PE ルータは、別の GRE クラウドとの間で GRE トラフィックを開始、転送、または終了できます。同様に、すべてのトンネル通過ノードまたはトンネルエンド ノードは、MPLS トンネル カプセル化を設定できます。

Cisco Nexus 9000 スイッチでトンネルとセグメント ルーティングの両方が有効になっている場合、それぞれのフローの TTL 動作は次のとおりです。

- 着信 IP トラフィック、GRE ヘッダー付きの出力では、GRE ヘッダーの TTL 値は、着信 IP パケットの TTL 値より 1 少ない値です。
- 着信 IP トラフィック、MPLS ヘッダー付きの出力では、MPLS ヘッダーの TTL 値は、着信 IP パケットの TTL 値より 1 少ない値です。
- 着信 GRE トラフィック、MPLS ヘッダー付きの出力、MPLS ヘッダーの TTL 値はデフォルト (255) です。
- 着信 MPLS トラフィック、GRE ヘッダー付きの出力、GRE ヘッダーの TTL 値はデフォルト (255) です。

セグメント ルーティング MPLS および GRE の注意事項と制限事項

セグメント ルーティング MPLS および GRE には、次の注意事項と制限事項があります。

- トンネル パケットの入力統計はサポートされていません。
- template-mpls-heavy テンプレートのみがサポートされています。
- MPLS セグメント ルーティングは、トンネル インターフェイスではサポートされていません。

- モジュラ スイッチのハードウェア制限により、トンネルの宛先IPアドレスの出力インターフェイスが Cisco Nexus 9300-FX/FX2 プラットフォーム スイッチを越える場合、トンネル Tx トラフィックはサポートされません。
- 最大 4 つの GRE トンネルがサポートされます。
- Cisco NX-OS リリース 9.3(3) 以降、Cisco Nexus 9300-GX プラットフォーム スイッチ上ではセグメント ルーティング MPLS と GRE の両方を設定できます。
- セグメント ルーティング MPLS と GRE の両方が共存している場合、トンネル Rx パケット カウンタは機能しません。

セグメント ルーティング MPLS および GRE の設定

静的 MPLS などの相互に排他的な MPLS 機能がイネーブルになっていない限り、MPLS セグメント ルーティングをイネーブルにできます。

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。

feature tunnel コマンドを使用して、トンネリ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)#`
		グローバル設定モードを開始します。

ステップ 2	[**no**] **feature segment-routing** 例： `switch(config)# feature segment-routing`	MPLS セグメント ルーティング機能を有効化します。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	（任意） **show running-config | inc 'feature segment-routing'** 例： `switch(config)# 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。
ステップ 4	（任意） **copy running-config startup-config** 例：	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

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

ステップ 5	**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 6	**feature tunnel**
	例：
	`switch(config)# feature tunnel`
	`switch(config-if)#`
		新しいトンネルインターフェイスを作成できます。
		トンネルインターフェイス機能を無効にするには、このコマンドの **no** 形式を使用します。

ステップ 7	switch(config)# **interface tunnel** *number*	トンネル 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	switch(config-if)# **tunnel mode** {**gre ip** }	このトンネル モードを GRE に設定します。 IP での GRE カプセル化の使用を指定するには、**gre** キーワードおよび **ip** キーワードを指定します。
ステップ 9	**tunnel source** {*ip-address* | *interface-name*} 例： `switch(config-if)# tunnel source ethernet 1/2`	この IP トンネルの送信元アドレスを設定します。送信元は、IP アドレスまたは論理インターフェイス名によって指定できます。
ステップ 10	**tunnel destination** *ip{address | hostname}* 例： `switch(config-if)# tunnel destination 192.0.2.1`	この IP トンネルの宛先アドレスを設定します。宛先は、IP アドレスまたは論理ホスト名によって指定できます。
ステップ 11	**tunnel use-vrf** *vrf-name* 例： `switch(config-if)# tunnel use-vrf blue`	
ステップ 12	**ipv6 address** *IPv6* アドレス	`switch(config-if)# 10.1.1.1` IPv6 アドレス を設定します。 （注） トンネルの送信元アドレスと宛先アドレスは同じままです（IPv4アドレス）。

	コマンドまたはアクション	目的
ステップ 13	（任意） switch(config-if)# **show interface tunnel** *number*	トンネルインターフェイスの統計情報を表示します。
ステップ 14	switch(config-if)# **mtu** *value*	インターフェイスで送信される IP パケットの Maximum Transmission Unit（MTU;最大伝送単位）を設定します。
ステップ 15	（任意） switch(config-if)# **copy running-config startup-config**	リブートおよびリスタート時に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て、変更を継続的に保存します。

セグメント ルーティング MPLS および GRE の設定の確認

スタティック ルーティング MPLS および GRE 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 segment-routing mpls	セグメント ルーティング MPLS 情報を表示します

レイヤ 3 EVPN の SR-TE の確認

ODN の検証は、L3VPN VRF プレフィックスに基づいています。

1. R1（ヘッドエンドと PCE サーバー）間の PCEP セッションが確立されていることを確認します。

```
R1# show srte pce ipv4 peer

PCC's peer database:
--------------------
Remote PCEP conn IPv4 addr: 58.8.8.8
Local PCEP conn IPv4 addr: 51.1.1.1
Precedence: 0
State: up
```

2. 次のコマンドを使用して、R1、R3、および R6 の BGP LS および BGP EVPN セッションを確認します。
   - Show bgp l2vpn evpn summary
   - Show bgp link-state summary

3. R1（ヘッドエンド）に、R6 ループバック アドレスへの可視性がないことを確認します。

```
R1# show ip route 56.6.6.6
IP Route Table for VRF "default"
```

```
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

56.6.6.6/32, ubest/mbest: 1/0
    *via Null0, [1/0], 1d02h, static
```

4. VRF プレフィックスが MP-BGP によって R1 VRF SR ルーティング テーブルにインジェクトされることを確認します。

```
R1# show ip route vrf sr
106.107.4.1/32, ubest/mbest: 1/0
    *via binding label 100534%default, [20/0], 1d01h, bgp-6503, external, tag 6500
(mpls-vpn)
```

5. SR-TE トンネルを確認します。

```
R1# show srte policy
Policy name: 51.1.1.1|1001
    Source: 51.1.1.1
    End-point: 56.6.6.6
    Created by: bgp
    State: UP
    Color: 1001
    Insert: FALSE
    Re-opt timer: 0
    Binding-sid Label: 100534
    Policy-Id: 2
    Flags:
    Path type = MPLS           Path options count: 1
     Path-option Preference:100 ECMP path count: 1
      1.      PCE         Weighted: No
        Delegated PCE: 58.8.8.8
                Index: 1                Label: 101104
                Index: 2                Label: 201102
                Index: 3                Label: 201103
```

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。

コマンド	目的
show running-config rpm	ルート ポリシー マネージャ（RPM）についての情報を表示します。
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。
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。

コマンド	目的
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
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。
show nve evi	EVI のステータスを表示します。
show nve peer mpls	セグメントルーティング ピアのステータスを表示します。
show nve adjacency mpls	ピア隣接のステータスを表示します。

SRTE 明示パス エンドポイント置換の構成

この章には、SRTE 明示パス エンドポイント置換機能を構成する方法に関する情報が含まれています。

SRTE 明示パス エンドポイント置換

SRTE 明示パス エンドポイント置換機能を使用すると、ユーザーは明示パスを一連の MPLS ラベル（通常の明示パスと同様）として定義できますが、ポリシー エンドポイント ラベルを表す一連のプレースホルダーを追加できます。プレースホルダーは、**policy-endpoint** キーワードで表されます。ポリシーエンドポイントプレースホルダーが表示されるパス内の位置は、SRTE によって、ポリシーのエンドポイント IP アドレスのノード SID を表すセグメント ルーティング ラベルに内部的に解決されます。

これは、定義する必要があるポリシーの総数を減らすため、オンデマンドのカラーテンプレートと組み合わせて使用すると役立ちます。カラーとエンドポイントの組み合わせごとに個別のパスを定義する代わりに、ユーザーは、その色のすべてのエンドポイントのポリシーを定義するためのエンドポイント置換を含む明示的なパスを含むオンデマンド カラー テンプレートを定義できます。

SRTE 明示パス エンドポイント置換の注意事項と制限事項

SRTE 明示パス エンドポイントの置換には、次の注意事項と制限事項があります。

- Cisco NX-OS Release 10.1(1) 以降、SRTE 明示パス エンドポイント置換は、Cisco Nexus 9300-FX、9300-FX2、9300-FX3、および 9300-GX プラットフォーム スイッチでサポートされています。
- 部分パスが解決されたエンドポイント ラベルと同じラベルで終わる場合、余分な（重複した）トランスポート ラベルを追加しないでください。
- SRGB はすべてのノードで同じでなければなりません。そうでない場合、各中間ノードのセグメント構成によっては、機能が動作しない場合があります。
- セグメント リストには、ポリシー エンドポイント エントリを 1 つだけ含めることができます。

SRTE 明示的パス エンドポイント置換の構成

エンドポイント置換を使用するポリシーを作成するには、最初にセグメント リスト モードを使用してパスを定義します。次に、その名前を使用してパスをオンデマンドの色に関連付けます。

始める前に

MPLS セグメント ルーティング トラフィック エンジニアリング機能が有効になっていることを確認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	**segment-routing**
	例：
	`switch(config)#segment-routing`
	`switch(config-sr)#`
		セグメント ルーティング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 3	**traffic-engineering**
	例：
	`switch(config-sr)# traffic-engineering`
	`switch(config-sr-te)#`
		トラフィック エンジニアリング モードに入ります。

ステップ 4	**segment-list name** *path*
	例：
	`switch(config-sr-te)# segment-list name path`
	`switch(config-sr-te-exp-seg-list)#`
		明示的なセグメント リストを構成します。

ステップ 5	**index 1 mpls label** *label-ID*
	例：
	`switch(config-sr-te-exp-seg-list)# index 1 mpls label 16201`
	`switch(config-sr-te-exp-seg-list)#`
		セグメント リストに MPLS ラベルを構成します。

ステップ 6	**index 2 policy-endpoint**
	例：
	`switch(config-sr-te-exp-seg-list)# index 2 policy-endpoint`
	`switch(config-sr-te-exp-seg-list)#`
		ポリシーのエンドポイント解決を構成します。

ステップ 7	**exit**
	例：
	`switch(config-sr-te-exp-seg-list)# exit`
	`switch(config-sr-te)#`
		セグメント リスト モードを終了し、SRTE モードに戻ります。

ステップ 8	**on-demand color** *color_num*
	例：
	`switch(config-sr-te)# on-demand color 201`
	`switch(config-sr-te-color)#`
		オンデマンド色テンプレート モードを開始し、特定の色のオンデマンド色を構成します。

ステップ 9	**candidate-paths** 例： `switch(config-sr-te-color)# candidate-paths`	SR-TE カラー ポリシーの候補パスを指定します。
ステップ 10	**preference** *preference-number* 例： `switch(cfg-cndpath)# preference 100`	候補パスの優先順位を指定します。
ステップ 11	**explicit segment-list** *path* 例：	明示セグメント リストを指定します。

	コマンドまたはアクション	目的
	`switch(cfg-pref)# explicit segment-list path`	

SRTE 明示パス エンドポイントの置換構成例

この例は、SRTE 明示パス エンドポイントの置換構成を示しています。

```
switch(config)# segment-routing
switch(config-sr)# traffic-engineering
switch(config-sr-te)# segment-list name path
switch(config-sr-te-exp-seg-list)# index 1 mpls label 16201
switch(config-sr-te-exp-seg-list)# index 2 policy-endpoint
switch(config-sr-te-exp-seg-list)# exit
switch(config-sr-te)# on-demand color 201
switch(config-sr-te-color)# candidate-paths
switch(cfg-cndpath)# preference 100
switch(cfg-pref)# explicit segment-list path
```

SRTE 明示パス エンドポイント置換の構成の確認

SRTE 明示パス エンドポイント置換構成に関する必要な詳細を表示するには、次のいずれかのタスクを実行します。

表 *14 : SRTE* 明示パス エンドポイントの置換構成の確認

コマンド	目的
show srte policy	許可されたポリシーのみを表示します。 （注） エンドポイント ラベルが解決され、最初のホップに到達できる場合、状態は UP と表示されます。エンドポイント ラベルが解決されていない場合、または最初のホップに到達できない場合、状態は DOWN と表示されます。
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。 （注） エンドポイント ラベルが解決され、最初のホップに到達できる場合、状態は UP と表示されます。エンドポイント ラベルが解決されていない場合、または最初のホップに到達できない場合、状態は DOWN と表示されます。

コマンド	目的
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。 （注） エンドポイント ラベルが解決され、最初のホップに到達できる場合、状態は UP と表示されます。エンドポイント ラベルが解決されていない場合、または最初のホップに到達できない場合、状態は DOWN と表示されます。
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	既存の最初のホップとポリシー エンドポイントの状態を表示します。

デフォルト VRF を介した SRTE の構成

デフォルト VRF を介した SRTE について

デフォルト VRF を介した SRTE 機能を使用すると、セグメント ルーティング トラフィック エンジニアリングを組み込んで、ネットワークでトラフィック ステアリングの利点を実現できます。SRTE は、大規模なデータセンター（DC）でのルーティングに BGP を使用しながら、スケーラビリティを向上させます。

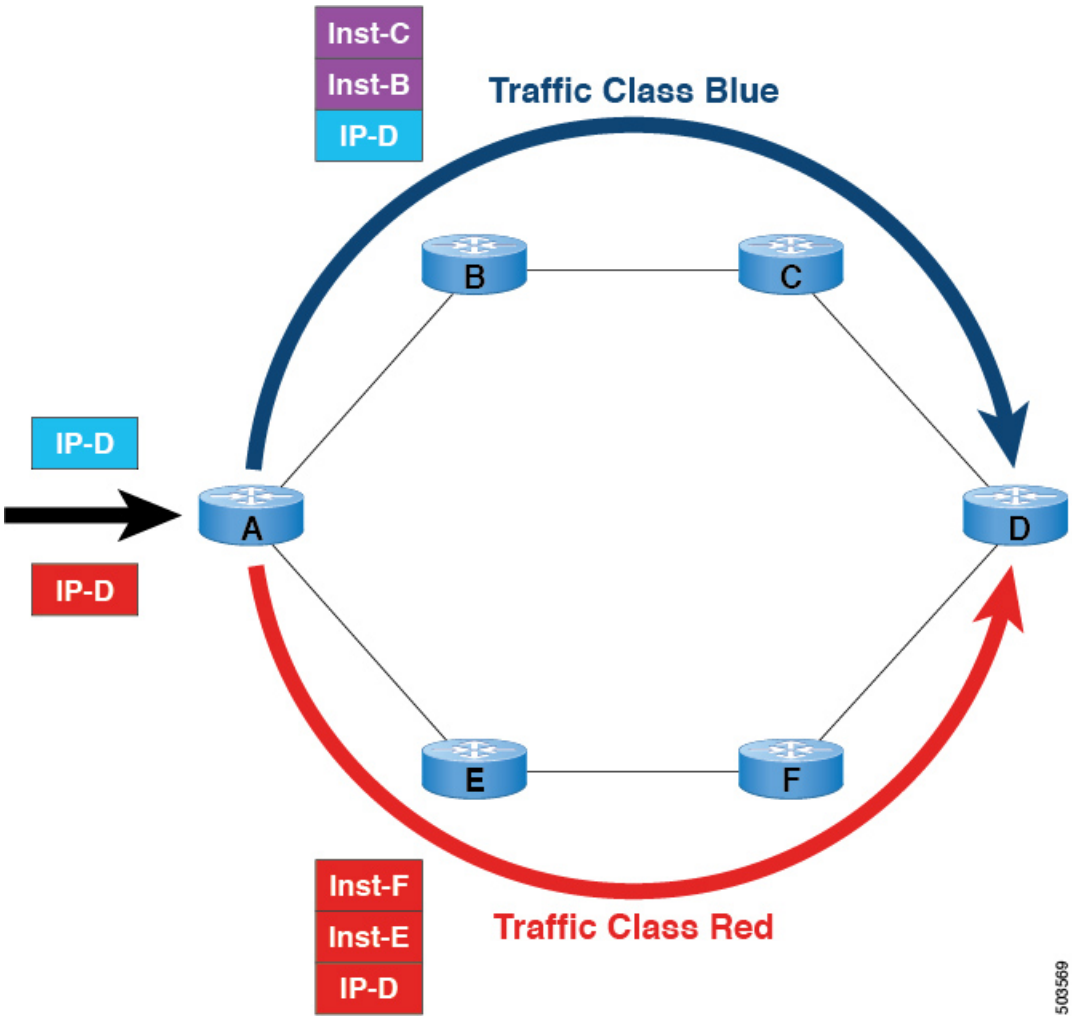
デフォルト VRF を介した SRTE 機能は、拡張コミュニティ属性として存在し、トラフィックステアリングのベースとして番号で表されるルートカラーを使用します。カラーに基づいてプレーン分離が実現され、トラフィックを伝送するための SR ポリシーが作成されます。さらにカラーに基づいて、DCはさまざまなプレーンに分割されます。アプリケーションは、各プレーンを使用して特定のプレーンのみをルーティングし、トラフィックを適切な宛先に誘導するように構成されています。

平面分離には次の利点があります。

- 1 つのフローが他のフローに影響を与えることはありません。
- 大小のフローは、異なる平面に分離されます。
- デバッグを容易にするための障害分離：1 つのプレーンの障害が他のプレーンに影響を与えることはありません。たとえば、1 つのプレーンでネットワーク障害が発生した場合、そのプレーンのアプリケーションのみが影響を受けますが、残りのプレーンのアプリケーションは影響を受けません。さらに、障害を分離し、分離してトラブルシューティングを行うことができます。

次の例では、図を使用してデフォルト VRF を介した SRTE 機能を説明しています。

図 *16 :* デフォルト *VRF* を介した *SRTE* の例



- BGP の場合、ノード A は入力ルータであり、ノード D は出力ルータです。D はネクストホップでもあります。
- SRTE の場合、ノード A は SRTE ヘッドエンドであり、ノード D はポリシーのエンドポイントです。
- ルート プレフィックス 1 はブルー プレーンを使用するように構成され、ルート 2 はレッド プレーンを使用するように構成されています。

青のトラフィックには、ノード B とノード C を介してトラフィックを誘導する命令が追加され、赤のトラフィックには、ノード E とノード F を経由してトラフィックを誘導する命令が追加されます。要約すると、トラフィックはアドバタイズメントのカラーに基づいて処理されます。これは、以前にアドバタイズされたプレフィックスです。

デフォルト VRF 経由の SRTE を構成する場合の注意事項と制限事項

- Cisco NX-OS リリース 10.1(1) 以降、セグメント ルーティング トラフィック エンジニアリングは、Cisco Nexus 9300-FX3、N9K-C9316D-GX、N9K-C93180YC-FX、N9K-C93240YC-FX2、および N9K-C9364C プラットフォーム スイッチのデフォルト VRF でサポートされます。この SR-TE 機能の制限は次のとおりです。
  - アンダーレイ IPv6 はサポートされません。SRv6 は代替です。
  - BGP の専用ファブリックにおける PCE の欠点のため、BGP アンダーレイを使用した PCE はサポートされていません。
  - NXOS が BGP-LS で LSA をアドバタイズできないため、PCE を使用した OSPF-SRTE はサポートされていません。
  - 合計 1000 の SRTE ポリシー スケール、130K v4 の BGP デフォルト VRF（v4）、および 1000 のアンダーレイ SR プレフィックスをサポートします。
- Cisco NX-OS リリース 10.2(3)F 以降、カラー専用（CO）ビットのオプションがルートマップに追加されています。SRTE ポリシーを使用している特定のプレフィックスの CO ビットの値が変更された場合、BGP は古いポリシーを削除し、新しいポリシーを追加します。この機能は、Cisco Nexus 9300-EX、9300-FX、9300-FX2、9300-GX、および 9300-GX2 プラットフォーム スイッチでサポートされます。

構成プロセス：デフォルト VRF を介した SRTE

構成プロセスは次のとおりです。

1. ネクストホップを変更しない: ネクストホップは、入力ノードで SR ポリシーを計算するために使用されます。プレフィックスがアップストリームにアドバタイズされるため、プレフィックスの SR ドメインのネクストホップを保持する必要があります。したがって、ホップバイホップの ebgp の場合、すべての上流ルータでネクストホップが変更されていない必要があります。
2. 出力ノード、入力ノード、ネットワーク/再配布、またはデフォルト発信元で拡張コミュニティ カラーを設定します。
3. 入力ノードは、カラー拡張されたコミュニティを受信すると、それを SR ポリシーに一致させます。
4. SR ポリシーのエンドポイントは、カラー拡張コミュニティのプレフィックスとカラーのネクストホップから派生します。

このセクションには、デフォルト VRF での SRTE の構成に関する次のトピックが含まれています。

ネクストホップ変更なしの構成

デフォルト VRF オーバーレイの中間（スパイン）ノードでネクスト ホップを変更せずに構成し、ネクスト ホップが変更されないようにするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的

ステップ 1	**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 ABC`
	`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 ip next-hop unchanged**
	例：
	`switch(config-route-map)# set ip next-hop unchanged`
	`switch(config-route-map)#`
		ネクストホップを変更せずに設定します。

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	[**no**] **router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp1`
	`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。
		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 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。

	コマンドまたはアクション	目的
ステップ 7	**address-family ipv4 unicast** 例： `switch(config-router-neighbor)# address-family ipv4 unicast switch(config-router-neighbor-af)#`	IPv4 アドレス ファミリ タイプのルータのアドレスファミリ構成モードを開始します。
ステップ 8	**route-map** *map-name* **out** 例： `switch(config-router-neighbor-af)# route-map ABC out switch(config-router-neighbor-af)#`	発信ルートに設定された BGP ポリシーを適用します。

拡張コミュニティ カラーの構成

このセクションは、次のトピックで構成されています。

出力ノードでの拡張コミュニティ カラーの構成

プレフィックスが出力ノードによって通知されるときに、出力ノードで拡張コミュニティ カラーを構成するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**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 ABC 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	**set extcommunity color** *color-num* [**co-flag** *co-flag*] 例： `switch(config-route-map)# set extcommunity color 20 [co-flag 00] switch(config-route-map)#`	カラー拡張コミュニティの BGP 外部コミュニティ属性を設定します。 **co-flag**：カラー専用フラグを使用して、正確なカラーとエンドポイントのポリシーが見つからない場合に、カラーのみに基づいてトラフィックを SR ポリシーに誘導できるかどうかを制御します。デフォルトは 00 です。

	コマンドまたはアクション	目的
		（注） co-flag 00 を選択して、カラーとネクストホップに基づきデフォルトの自動ステアリングを指定します。co-flag が 00 もしくはデフォルトに設定されている場合、リクエストされたカラーとエンドポイントを持つポリシーのバインド SID がルーティングに使用されます。 co-flag 01 を選択し、カラーにのみ基づいてトラフィックを誘導します。co-flag が 01 に設定され、リクエストされたカラーとエンドポイントを持つポリシーが存在する場合、ポリシーのバインド SID がルーティングに使用されます。ポリシーが存在しないが、同じカラーを持つ null エンドポイント ポリシーが存在する場合、null エンドポイント ポリシーのバインド SID がルーティングに使用されます。

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	[**no**] **router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp1`
	`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。
		
		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。

	コマンドまたはアクション	目的

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 ipv4 unicast**
	例：
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		IPv4 アドレス ファミリ タイプのルータのアドレスファミリ構成モードを開始します。

ステップ 8	**route-map** *map-name* **out**
	例：
	`switch(config-router-neighbor-af)# route-map ABC out`
	`switch(config-router-neighbor-af)#`
		発信ルートに設定された BGP ポリシーを適用します。
		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。

入力ノードにおける拡張コミュニティ カラーの構成

SRTE ポリシーがインスタンス化される入力ノードによってプレフィックスが通知されるときに、入力ノードで拡張コミュニティ カラーを構成するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的

ステップ 1	**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 ABC`
	`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	**set extcommunity color** *color-num* [**co-flag** *co-flag*]
	例：
	`switch(config-route-map)# set extcommunity color 20 [co-flag 00]`
	`switch(config-route-map)#`
		カラー拡張コミュニティの BGP 外部コミュニティ属性を設定します。
		co-flag：カラー専用フラグを使用して、正確なカラーとエンドポイントのポリシーが見つからない場合に、カラーのみに基づいてトラフィックを SR ポリシーに誘導できるかどうかを制御します。デフォルトは 00 です。

	コマンドまたはアクション	目的
		（注） co-flag 00 を選択して、カラーとネクストホップに基づきデフォルトの自動ステアリングを指定します。co-flag が 00 もしくはデフォルトに設定されている場合、リクエストされたカラーとエンドポイントを持つポリシーのバインド SID がルーティングに使用されます。 co-flag 01 を選択し、カラーにのみ基づいてトラフィックを誘導します。co-flag が 01 に設定され、リクエストされたカラーとエンドポイントを持つポリシーが存在する場合、ポリシーのバインド SID がルーティングに使用されます。ポリシーが存在しないが、同じカラーを持つ null エンドポイント ポリシーが存在する場合、null エンドポイント ポリシーのバインド SID がルーティングに使用されます。

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	[**no**] **router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp1`
	`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。
		
		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。

	コマンドまたはアクション	目的

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 ipv4 unicast**
	例：
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		IPv4 アドレス ファミリ タイプのルータのアドレスファミリ構成モードを開始します。

ステップ 8	**route-map** *map-name* **in**
	例：
	`switch(config-router-neighbor-af)# route-map ABC in`
	`switch(config-router-neighbor-af)#`
		構成された BGP ポリシーを受信ルートに適用します。
		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。

出力ノードでのネットワーク/再配布コマンドの拡張コミュニティカラー構成

プレフィックスが出力ノードによって通知されるときに、出力ノードで network/redistribute コマンドの拡張コミュニティ カラーを構成するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的

ステップ 1	**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 ABC`
	`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	**set extcommunity color** *color-num* [**co-flag** *co-flag*]
	例：
	`switch(config-route-map)# set extcommunity color 20 [co-flag 00]`
	`switch(config-route-map)#`
		カラー拡張コミュニティの BGP 外部コミュニティ属性を設定します。
		co-flag：カラー専用フラグを使用して、正確なカラーとエンドポイントのポリシーが見つからない場合に、カラーのみに基づいてトラフィックを SR ポリシーに誘導できるかどうかを制御します。デフォルトは 00 です。

	コマンドまたはアクション	目的
		（注） co-flag 00 を選択して、カラーとネクストホップに基づきデフォルトの自動ステアリングを指定します。co-flag が 00 もしくはデフォルトに設定されている場合、リクエストされたカラーとエンドポイントを持つポリシーのバインド SID がルーティングに使用されます。 co-flag 01 を選択し、カラーにのみ基づいてトラフィックを誘導します。co-flag が 01 に設定され、リクエストされたカラーとエンドポイントを持つポリシーが存在する場合、ポリシーのバインド SID がルーティングに使用されます。ポリシーが存在しないが、同じカラーを持つ null エンドポイント ポリシーが存在する場合、null エンドポイント ポリシーのバインド SID がルーティングに使用されます。

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	[**no**] **router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp1`
	`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。
		
		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。

	コマンドまたはアクション	目的

ステップ 6	**address-family ipv4 unicast**
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		VRF インスタンスの IPv4 アドレス ファミリを指定し、アドレス ファミリ構成モードを開始します。

ステップ 7	**redistribute static route-map** *map-name* **out**
	例：
	`switch(config-router-af)# redistribute static route-map ABC`
	`switch(config-router-af)#`
		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 8	**network** *ip-prefix* [**route-map** *map-name*]
	例：
	`switch(config-router-af)# network 1.1.1.1/32 route-map ABC`
	`switch(config-router-af-network)#`
		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。

出力ノードで Default-Originate の拡張コミュニティ カラーの構成

デフォルトのプレフィックスが出力ノードによって通知されたときに、出力ノードで default-originate の拡張コミュニティ カラー構成するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的

ステップ 1	**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 ABC`
	`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。
		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 3	**set extcommunity color** *color-num* [**co-flag** *co-flag*] 例： `switch(config-route-map)# set extcommunity color 20 [co-flag 00]`	カラー拡張コミュニティの BGP 外部コミュニティ属性を設定します。 **co-flag**：カラー専用フラグを使用して、正確なカラーとエンドポイントのポリシーが見つからない場合に、カラーのみ

	コマンドまたはアクション	目的
		に基づいてトラフィックを SR ポリシーに誘導できるかどうかを制御します。デフォルトは 00 です。 （注） co-flag 00 を選択して、カラーとネクストホップに基づきデフォルトの自動ステアリングを指定します。co-flag が 00 もしくはデフォルトに設定されている場合、リクエストされたカラーとエンドポイントを持つポリシーのバインド SID がルーティングに使用されます。 co-flag 01 を選択し、カラーにのみ基づいてトラフィックを誘導します。co-flag が 01 に設定され、リクエストされたカラーとエンドポイントを持つポリシーが存在する場合、ポリシーのバインド SID がルーティングに使用されます。ポリシーが存在しないが、同じカラーを持つ null エンドポイント ポリシーが存在する場合、null エンドポイント ポリシーのバインド SID がルーティングに使用されます。

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	[**no**] **router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp1`
	`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 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 ipv4 unicast**
	例：
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		IPv4 アドレス ファミリ タイプのルータのアドレスファミリ構成モードを開始します。

ステップ 8	**default-originate** [**route-map** *map-name*]
	例：
	`switch(config-router-neighbor-af)# default-originate route-map ABC`
	`switch(config-router-neighbor-af)#`
		BGP ピアへのデフォルト ルートを作成します。
		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。

入力ピアの BGP の構成（SRTE ヘッドエンド）

入力ピア（SRTE ヘッドエンド）の BGP を構成するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的

ステップ 1	**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	[**no**] **feature bgp**
	例：
	`switch(config)# feature bgp`
	`switch(config)`
		BGP を有効にします。
		この no コマンド形式を使用して、この機能を無効にします。

ステップ 3	[**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 4	**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 5	**neighbor** *ip-address*
	例：
	`switch(config-router-af)# neighbor 209.165.201.1`
	`switch(config-router-af-neighbor)#`
		リモート BGP ピアの IPv4 アドレスを設定します。ip-address の形式は x.x.x.x です。

ステップ 6	**remote-as** *as-number* 例： `switch(config-router-af-neighbor)# remote-as 64497`	リモート BGP ピアの AS 番号を設定します。
ステップ 7	**update-source** *interface number* 例： `switch(config-router-af-neighbor)# update-source loopback 300`	BGP セッションの送信元を指定し、更新します。
ステップ 8	**ebgp-multihop** *ttl-value* 例： `switch(config-router-af-neighbor)# ebgp-multihop 5`	eBGP マルチホップの eBGP TTL を設定します。有効な範囲は 2 ～ 255 です。このコマンドの使用後、BGP セッションを手動でリセットする必要があります。
ステップ 9	**exit** 例： `switch(config-router-af-neighbor)# exit`	ネイバー コンフィギュレーション モードを終了します。

ステップ 10	**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 11	**route-map** *map-name* **in** 例：	SRTE 入力ピアのルート マップを指定します。

	コマンドまたはアクション	目的
	`switch(config-router-af)# route-map color 401 in`	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。 （注） NLRI に適用できる拡張コミュニティ カラーは 1 つのみなので、適用されたルート ポリシー/ルート マップは、以前の拡張コミュニティ カラーが存在する場合は上書きします。

入力ピアの BGP 構成（SRTE エンドポイント）

出力ピア（SRTE エンドポイント）の BGP を構成するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的

ステップ 1	**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	[**no**] **feature bgp**
	例：
	`switch(config)# feature bgp`
	`switch(config)`
		BGP を有効にします。
		この no コマンド形式を使用して、この機能を無効にします。

ステップ 3	[**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。
		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 4	**neighbor** *ip-address*
	例：
	`switch(config-router)# neighbor 209.165.201.1`
	`switch(config-router-neighbor)#`
		リモート BGP ピアの IPv4 アドレスを設定します。ip-address の形式は x.x.x.x です。

	コマンドまたはアクション	目的
ステップ 5	**remote-as** *as-number* 例： `switch(config-router-neighbor)# remote-as 64497`	リモート BGP ピアの AS 番号を設定します。
ステップ 6	**update-source** *interface-number* 例： `switch(config-router-neighbor)# update-source loopback 300`	BGP セッションの送信元を指定し、更新します。
ステップ 7	**ebgp-multihop** *ttl-value* 例： `switch(config-router-neighbor)# ebgp-multihop 5`	eBGP マルチホップの eBGP TTL を設定します。有効な範囲は 2 ～ 255 です。このコマンドの使用後、BGP セッションを手動でリセットする必要があります。
ステップ 8	**exit** 例： `switch(config-router-af-neighbor)# exit`	ネイバー コンフィギュレーション モードを終了します。

ステップ 9	**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 10	**send-community**
	例：
	`switch(config-router-af)# send-community`
	`switch(config-router-af)#`
		BGP コミュニティ属性を BGP ネイバーに送信する必要があることを指定します。

ステップ 11	**send-community extended**
	例：
	`switch(config-router-af)#send-community extended`
	`switch(config-router-af)#`
		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 12	**route-map** *map-name* **out**
	例：
	`switch(config-router-af)# route-map color 301 out`
	`switch(config-router-af)#`
		SRTE 出力ピアのルート マップを指定します。
		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。

	コマンドまたはアクション	目的
		（注） NLRI に適用できる拡張コミュニティ カラーは 1 つのみなので、適用されたルート ポリシー/ルート マップは、以前の拡張コミュニティ カラーが存在する場合は上書きします。

入力ピア用 SRTE の構成

入力ピア（SRTE ヘッドエンド）の SRTE を構成するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的

ステップ 1	**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	[**no**] **feature mpls segment-routing traffic-engineering**
	例：
	`switch(config)# feature mpls segment-routing traffic-engineering`
	`switch(config)`
		MPLS SRTE を有効にします。
		この no コマンド形式を使用して、この機能を無効にします。

ステップ 3	**segment-routing**
	例：
	`switch(config)#segment-routing`
	`switch(config-sr)#`
		セグメント ルーティング構成モードを開始します。

ステップ 4	**traffic-engineering**
	例：
	`switch(config-sr)# traffic-engineering`
	`switch(config-sr-te)#`
		トラフィック エンジニアリング モードに入ります。

ステップ 5	**segment-list name** *path*
	例：
	`switch(config-sr-te)# segment-list name path`
	`switch(config-sr-te-exp-seg-list)#`
		明示的なセグメント リストを構成します。

ステップ 6	**index 1 mpls label** *label-ID* 例：	セグメント リストに MPLS ラベルを作成します。

	コマンドまたはアクション	目的
	`switch(config-sr-te-exp-seg-list)# index 1 mpls label 16601 switch(config-sr-te-exp-seg-list)#`	
ステップ 7	**index 2 mpls label** *label-ID* 例： `switch(config-sr-te-exp-seg-list)# index 2 mpls label 16501 switch(config-sr-te-exp-seg-list)#`	セグメント リストに MPLS ラベルを作成します。
ステップ 8	**policy** *policy-name*-**bgp** 例： `switch(config-sr-te-exp-seg-list)# policy dci1-edge1-bgp switch(config-sr-te-exp-seg-list)#`	SRTE ポリシー名を指定します。
ステップ 9	**color** *color-num* **endpoint** *endpoint ID* 例： `switch(config-sr-te)# color 13401 endpoint 1.0.3.1`	ポリシーのカラーとエンドポイントを指定します（SRTE 出力ノード ループバック）。
ステップ 10	**candidate-paths** 例： `switch(config-sr-te-color)# candidate-paths`	SRTE カラー ポリシーの候補パスを指定します。
ステップ 11	**preference** *preference-number* 例： `switch(cfg-cndpath)# preference 100`	候補パスの優先順位を指定します。
ステップ 12	**explicit segment-list** *path* 例： `switch(cfg-pref)# explicit segment-list path`	明示セグメント リストを指定します。

デフォルト VRF 経由の SRTE 構成例

次の例は、デフォルトの VRF 構成を介した SRTE を示しています。

構成例：ネクストホップ変更なし

```
route-map ABC
  set ip next-hop unchanged

router bgp 1
  neighbor 1.2.3.4
    address-family ipv4 unicast
       route-map ABC out
```

構成例：拡張コミュニティ カラー

このセクションには、拡張コミュニティ カラーの次の構成例が含まれます。

構成例：出力ノード

```
ip prefix-list pfx1 seq 5 permit 7.7.7.7/32
ip prefix-list pfx2 seq 5 permit 5.0.0.0/24
route-map ABC
  match ip address prefix-list pfx1 pfx2
    set extcommunity color 20

router bgp 1
  neighbor 1.2.3.4
    address-family ipv4 unicast
       route-map ABC out
```

入力ノードの構成例

```
ip prefix-list pfx1 seq 5 permit 7.7.7.7/32
ip prefix-list pfx2 seq 5 permit 5.0.0.0/24
route-map ABC
  match ip address prefix-list pfx1 pfx2
    set extcommunity color 20

router bgp 1
  neighbor 1.2.3.4
    address-family ipv4 unicast
       route-map ABC in
```

出力ノードでネットワーク/再配布コマンドの構成例

```
route-map ABC
  set extcommunity color 20

router bgp 1
  address-family ipv4 unicast
    redistribute static route-map ABC
    network 1.1.1.1/32 route-map ABC
```

構成例：出力ノードでデフォルトの生成をする場合

```
route-map ABC
  set extcommunity color 20

router bgp 1
    neighbor 1.2.3.4
       address-family ipv4 unicast
           default-originate route-map ABC
```

構成例：入力ピアの BGP（SRTE ヘッドエンド）

```
DCI-1(config)# show running-config bgp
feature bgp
router bgp 100
  address-family ipv4 unicast
  neighbor 1.0.3.1
    remote-as 101
    update-source loopback0
    ebgp-multihop 255
    address-family ipv4 unicast
       route-map color-3401 in
```

構成例：出力ピアの BGP（SRTE エンドポイント）

この例は、SRTE 明示パス エンドポイントの置換構成を示しています。

```
Edge-1(config)# show running-config bgp
feature bgp
router bgp 101
neighbor 1.0.1.1
    remote-as 100
    update-source loopback0
    ebgp-multihop 255
    address-family ipv4 unicast
      send-community
      send-community extended
      route-map color-3401 out
```

構成例：SRTE の入力ピア（SRTE ヘッドエンド）

```
DCI-1# show running-config srte
feature mpls segment-routing traffic-engineering
segment-routing
  traffic-engineering
    segment-list name dci1-edge1
      index 1 mpls label 16601
      index 2 mpls label 16501
    policy dci1-edge1-bgp
      color 13401 endpoint 1.0.3.1
      candidate-paths
        preference 30
          explicit segment-list dci1-edge1
```

デフォルト VRF を介した SRTE 構成の確認

デフォルトの VRF 構成を介した SRTE に関する適切な詳細を表示するには、次のいずれかのタスクを実行します。

表 *15：*デフォルト *VRF* 構成を介した *SRTE* の確認

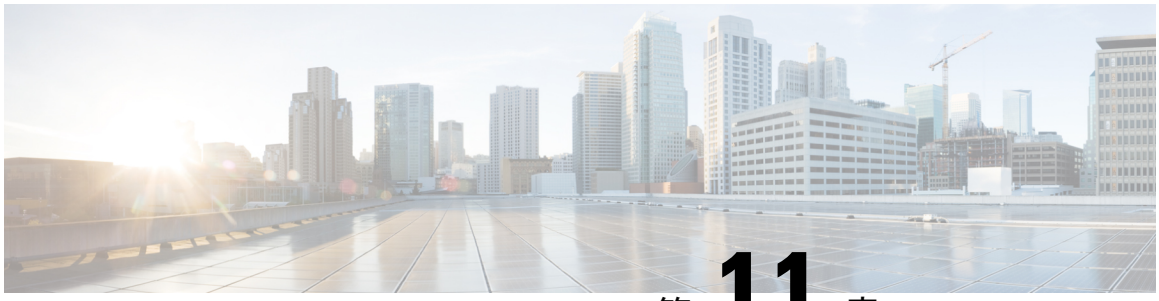
コマンド	目的
show running-config bgp	入力ピアまたは SRTE ヘッドエンドに関する情報を表示します。
show running-config bgp	出力ピアまたは SRTE エンドポイントに関する情報を表示します。
show running-config srte	入力ピアの SRTE ポリシーに関する情報を表示します。

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 9000* シリーズ ユニキャスト ルーティング設定ガイド

関連資料



第 11 章

MPLS セグメント ルーティング OAM 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）セグメント ルーティング OAM 機能について説明します。

MPLS セグメント ルーティング OAM について

MPLS セグメント ルーティング（SR）は、Cisco Nexus 9000 シリーズ スイッチに展開されています。MPLS セグメント ルーティング（SR）の展開に伴い、セグメント ルーティング ネットワークの設定ミスや障害を解決するために、いくつかの診断ツールが必要になります。セグメント ルーティング保守運用管理（OAM）は、ネットワークの障害検出とトラブルシューティングに役立ちます。これを使用することで、サービス プロバイダーはラベル スイッチド パス（LSP）をモニタしてフォワーディングの問題を迅速に特定できます。

MPLS SR OAM は、診断目的で 2 つの主要な機能を提供します。

1. MPLS ping
2. MPLS Traceroute

セグメント ルーティング OAM 機能は、次の FEC タイプをサポートします。

- SR-IGP IS-IS IPv4 プレフィックスへの ping およびトレースルート。これにより、IS-IS SR アンダーレイで配布されるプレフィックス SID の検証が可能になります。
- BGP IPv4 プレフィックスへの ping およびトレースルート。これにより、BGP SR アンダーレイで配布されるプレフィックス SID の検証が可能になります。

- 汎用 IPv4 プレフィックスへの ping およびトレースルート。これにより、配布を実行したプロトコルに依存しない SR アンダーレイで配布されたプレフィックス SID の検証が可能になります。検証は、ユニキャスト ルーティング情報ベース（URIB）とユニキャスト ラベル情報ベース（ULIB）をチェックすることによって実行されます。
- Nil FEC プレフィックスへの ping およびトレースルート。これにより、ping またはトレースルートが通過するパスをより詳細に制御して、MPLS SR プレフィックスに対応する部分に限ったデータ プレーンのみの検証が可能になります。パスは、SR-TE ポリシー名または SR-TE ポリシーのカラーとエンドポイントを使用して指定できます。

Cisco Nexus 9000 シリーズ スイッチで MPLS OAM を有効にするには、**feature mpls oam** CLI コマンドを使用します。Cisco Nexus 9000 シリーズ スイッチで MPLS OAM を無効にするには、**no feature mpls oam** CLI コマンドを使用します。

セグメント ルーティング Ping

IP ping が IP ホストへの接続を検証するのと同様に、MPLS ping は、MPLS ラベル スイッチド パス (LSP) に沿った単方向の連続性を検証するために使用されます。検証される LSP を表す FEC を提供することにより、MPLS ping は次のことを実行します。

- FEC のエコー要求が LSP のエンドポイントに到達することを確認します。Nil FEC を除き、他のすべての FEC タイプについては、エンドポイントがその FEC の正しい出力先であることを確認します。
- 低密度ラウンド トリップ時間を測定します。
- 低密度ラウンド トリップ遅延を測定します。

MPLS LSP ping 機能を使用して、LSP に沿った入力ラベル スイッチ ルータ（LSR）と出力 LSR 間の接続を確認します。MPLS LSP ping は、Internet Control Message Protocol（ICMP）のエコー要求メッセージと応答メッセージと同様に、LSP の検証に MPLS エコーの要求メッセージと応答メッセージを使用します。MPLS エコー要求パケットの宛先 IP アドレスは、ラベル スタックの選択に使用されるアドレスとは異なります。宛先 IP アドレスは 127.x.y.z/8 アドレスとして定義され、LSP が壊れている場合は IP パケットがそれ自体の宛先へ IP を切り替えないようにします。

セグメント ルーティング Traceroute

MPLS traceroute は、LSP の各ホップでフォワーディング プレーンおよびコントロール プレーンを検証して、障害を切り分けます。traceroute は、TTL 1 から始まり単調増加する存続可能時間（TTL）で MPLS エコー要求を送信します。TTL の有効期限が過ぎると、中継ノードはソフトウェアで要求を処理し、ターゲット FEC と目的の中継ノードへの LSP があるかどうかを確認します。中継ノードは、検証が成功した場合、ネクストホップに到達するための上記の検証とラベルスタックの結果を指定するリターンコードと、宛先に向かうネクストホップの ID を含むエコー応答を送信します。発信元は、TTL + 1 を含む次のエコー要求をビルドするために

エコー応答を処理します。宛先が FEC に対する出力であると応答するまで、プロセスが繰り返されます。

MPLS LSP のトレースルート機能を使用して、LSP の障害ポイントを隔離します。これはホップバイホップエラーのローカリゼーションとパストレースに使用されます。MPLS LSP traceroute 機能は、エコー要求を伝送するパケットの存続可能時間（TTL）値の期限切れに依存します。MPLS エコー要求メッセージが中継ノードを見つけると TTL 値をチェックし、期限が切れている場合はコントロールプレーンにパケットが渡されます。それ以外の場合は、メッセージが転送されます。エコー メッセージがコントロール プレーンに渡されると、要求メッセージの内容に基づいて応答メッセージが生成されます。

MPLS SR OAM に関する注意事項と制限事項

MPLS OAM Nil FEC に関する注意事項と制限事項は次のとおりです。

- MPLS OAM Nil FEC は、Nexus 9300-FX プラットフォーム スイッチでサポートされています。
- MPLS OAM Nil FEC は、-R ライン カード搭載の Cisco Nexus 9500 プラットフォーム スイッチではサポートされていません。
- Cisco NX-OS リリース 9.3(1) でサポートされるすべての新しい FEC タイプでは、1 つのラベル スタックのみがサポートされます。FEC スタック変更 TLV サポートおよび関連する検証はサポートされていません。この制限は、Nil FEC には適用されません。
- Cisco NX-OS リリース 9.3(1) では、RFC 8287 で記述されている SR-IGP の「任意の」プレフィックス タイプおよび隣接関係 SID はサポートされていません。
- OSPF ping とトレースルートは、Cisco NX-OS リリース 9.3(1) ではサポートされていません。
- Cisco NX-OS リリース 9.3(3) 以降、MPLS OAM Nil FEC は Cisco Nexus 9300-GX プラットフォーム スイッチでサポートされます。
- **ping mpls nil-fec** コマンドおよび **traceroute mpls nil-fec** コマンドには、最大 4 つのラベルを指定できます。この値は、プラットフォームを照会することによって適用されるもので、現在、Cisco Nexus 9000 シリーズ スイッチはラベル スタックを 5 に制限しています。これは、Nil FEC エコー リクエストの場合、内部的に余分な明示的ヌルが追加されるため、指定できるラベルが最大 4 つであることを意味します。
- ping およびトレースルート コマンドで指定されるネクストホップは、発信元で接続されたネクストホップでなければならず、再帰的ネクストホップであってはなりません。
- ツリートレースはサポートされていません。
- NIl FEC は、意図されたターゲットを特定するための情報を一切保持しません。パケットは正しくないノードで誤転送されることがありますが、非ヌル ラベルをポップした後にパケットがノードに到達した場合、検証が成功を返す可能性があります。

- Nil FEC は、情報を転送するだけで動作します。定義上、コントロール プレーンと転送プレーン間の不整合を検出することはできません。
- Nil FEC ping およびトレースルートは、デアグリゲータ（VRF ごと）ラベルではサポートされていません。これには、BGP EVPN レイヤー 3 のデアグリゲータ ラベルが含まれます。
- Broadcom チップセットを使用する Cisco Nexus 9000 シリーズ スイッチでは、ソフトウェアがクエリを送信して、パケットがどの ECMP を使用するかを判断できるようにするサポートはありません。このことは、次の例に示すように、これらのスイッチの 1 つを通過する MPLS トレースルートでは、複数の ECMP がある場合、次のホップでエラーが表示される可能性があることを意味します。

  ```
  D 2 6.0.0.2 MRU 1496 [Labels: 2003/explicit-null Exp: 0/0] 4 ms
  ```

- OAM を使用して BGP EPE LSP をテストする場合（たとえば、ping/トレースルート ラベル スタックの最後のラベルが EPE ラベルである場合）、OAM は、最終ルータで OAM が有効になっていて、着信インターフェイスで MPLS が有効になっている場合にのみ、成功を返します。

  たとえば、A---B---C のようにセットアップされていて、A と B が SR ネットワーク内にあり、B が PE のように動作し、C が CE のように動作する場合、B は C を BGP EPE ピア（B で出力エンジニアリングを使用）として設定します。この場合、C は着信インターフェイスで OAM および MPLS 転送を有効にする必要があります。

Nil FEC の MPLS ping とトレースルート

Nil-FEC LSP ping およびトレースルートの操作は、通常の MPLS ping およびトレースルートの拡張機能です。Nil FEC LSP ping およびトレースルート機能は、セグメント ルーティングと MPLS スタティックをサポートしています。また、他のすべての LSP タイプに対する追加の診断ツールとしても機能します。

他の FEC タイプとは異なり、Nil FEC はコントロール プレーンの検証を提供しません。FEC ping またはトレースルート プローブは、MPLS OAM 機能が有効になっているすべてのスイッチに到達できます。

この機能は、オペレータに以下を指定することを許可することで、オペレータがラベル スタックを自由にテストできるようにします。

- ラベル スタック
- Outgoing interface
- ネクストホップ アドレス

セグメント ルーティングの場合、ルーティング パスに沿った各セグメント ノード ラベルおよび隣接関係ラベルは、イニシエータのラベル スイッチ ルータ（LSR）からのエコー要求メッセージのラベル スタックに入れられます。MPLS データ プレーンは、このパケットをラベル

スタック ターゲットに転送し、ラベル スタック ターゲットはエコー メッセージを送り返します。

[**ping**|**traceroute**] **mpls nil-fec labels** *comma-separated-labels* [**output** {**interface** *tx-interface*} [**nexthop** *nexthop-ip-addr*]] CLI コマンドを使用して、ping またはトレースルートを実行します。

SR-TE ポリシー名またはカラーとエンドポイントを設定した場合は、次の CLI コマンドを使用して ping またはトレースルートを実行し、既存の SR-TE ポリシー情報を使用できます。

[**ping**|**traceroute**] **mpls nil-fec** [**policy name** *name*] [**endpoint** *nexthop-ip-addr*] [**on-demand color** *color-num*] CLI コマンドで、ping またはトレースルートを実行します。

BGP および IGP プレフィックス SID 用の MPLS ping および トレースルート

プレフィックス SID 用の MPLS ping およびトレースルートの操作は、次のような BGP および IGP シナリオでサポートされています。

- IS-IS レベル内
- IS-IS レベル間
- BGP SR アンダーレイ

これらの FEC タイプは、追加のコントロール プレーン チェックを実行して、パケットが誤ってルーティングされないようにします。この検証により、ping された FEC タイプがスイッチに接続され、他のノードに配布されることが保証されます。Nil FEC はこの検証を提供しません。

MPLS エコー要求パケットは、ターゲット FEC スタック サブ TLV を伝送します。ターゲット FEC サブ TLV は、レスポンダによって FEC 検証のために使用されます。IGP/BGP IPv4 プレフィックス サブ TLV がターゲット FEC スタック サブ TLV に追加されました。IGP/BGP IPv4 プレフィックス サブ TLV には、プレフィックス SID、プレフィックス長、およびプロトコル（IS-IS）が含まれています。

トレースルートを実行するには、**ping**|**traceroute sr-mpls** A.B.C.D/LEN fec-type [*bgp* | *igp {isis}* | *generic*] CLI コマンドを使用します。

セグメント ルーティング OAM の確認

このセクションでは、セグメント ルーティング OAM 機能を確認するために使用できる CLI コマンドについて説明します。

- セグメント ルーティング OAM IS-IS の確認 （318 ページ）

セグメント ルーティング OAM IS-IS の確認

次の ping コマンドは、基盤となるネットワークが IS-IS の場合の SR OAM を表示するために使用されます。

```
switch# ping sr-mpls 11.1.1.3/32 fec-type igp isis

Sending 5, 100-byte MPLS Echos to IGP Prefix SID(IS-IS) FEC 11.1.1.3/32,
     timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type Ctrl-C to abort.
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/3 ms
 Total Time Elapsed 18 ms

switch# traceroute sr-mpls 11.1.1.3/32 fec-type igp isis


Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type Ctrl-C to abort.
  0 172.18.1.2 MRU 1500 [Labels: 16103 Exp: 0]
L 1 172.18.1.1 MRU 1504 [Labels: implicit-null Exp: 0] 4 ms
! 2 172.18.1.10 3 ms

switch# ping sr-mpls 11.1.1.3/32 fec-type igp isis verbose

Sending 5, 100-byte MPLS Echos to IGP Prefix SID(IS-IS) FEC 11.1.1.3/32,
     timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type Ctrl-C to abort.
!    size 100, reply addr 172.18.1.10, return code 3
!    size 100, reply addr 172.18.1.10, return code 3
!    size 100, reply addr 172.18.1.10, return code 3
!    size 100, reply addr 172.18.1.10, return code 3
!    size 100, reply addr 172.18.1.10, return code 3

Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/3 ms
 Total Time Elapsed 17 ms

switch# ping sr-mpls 11.1.1.3/32 fec-type igp isis destination 127.0.0.1 127.0.0.2 repeat
 1 verbose
```

```
Sending 1, 100-byte MPLS Echos to IGP Prefix SID(IS-IS) FEC 11.1.1.3/32,
     timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type Ctrl-C to abort.
Destination address 127.0.0.1
!    size 100, reply addr 172.18.1.10, return code 3

Destination address 127.0.0.2
!    size 100, reply addr 172.18.1.22, return code 3

Success rate is 100 percent (2/2), round-trip min/avg/max = 3/3/3 ms
 Total Time Elapsed 8 ms
```

Ping およびトレースルート CLI コマンドの使用例

IGP または BGP SR ping およびトレースルートの例

CLI を使用して、明示的な発信情報で Ping を実行する

fec CLI コマンドを使用して IS-IS SR ping を実行し、fec CLI コマンドを使用して BGP ping を実行します。 **ping sr-mpls fec-type igp isis ping sr-mpls fec-type bgp**

```
switch# ping sr-mpls 11.1.1.3/32 fec-type igp isis

Sending 5, 100-byte MPLS Echos to IGP Prefix SID(IS-IS) FEC 11.1.1.3/32,
     timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type Ctrl-C to abort.
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/3 ms
 Total Time Elapsed 18 ms

switch# ping sr-mpls 11.1.1.3/32 fec-type igp isis verbose

Sending 5, 100-byte MPLS Echos to IGP Prefix SID(IS-IS) FEC 11.1.1.3/32,
     timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
```

```
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type Ctrl-C to abort.
!    size 100, reply addr 172.18.1.10, return code 3
!    size 100, reply addr 172.18.1.10, return code 3
!    size 100, reply addr 172.18.1.10, return code 3
!    size 100, reply addr 172.18.1.10, return code 3
!    size 100, reply addr 172.18.1.10, return code 3

Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/3 ms
 Total Time Elapsed 17 ms
```

Nil FEC ping およびトレースルートの例

CLI を使用して、明示的な発信情報で Ping を実行する

ping を実行するには、**ping sr-mpls nil-fec labels** *comma-separated-labels* [**output** {**interface** *tx-interface*} [**nexthop** *nexthop-ip-addr*]] CLI コマンドを使用します。

たとえば、次のコマンドは、ラベル スタック内の最も外側の 2 つのラベル（2001 と 2000）を持つ MPLS パケットを、ネクストホップ IP アドレスが 4.0.0.2 のインターフェイス イーサネット 1/1 から送信します。

```
switch# ping mpls nil-fec labels 2001,2000 output interface e1/1 nexthop 4.0.0.2
```

ネクストホップは接続されたネクストホップであることが必須です。再帰的には解決されません。

上記の CLI 形式は簡易版です。[**output** {**interface** *tx-interface*} [**nexthop** *nexthop-ip-addr*]] は、VSH サーバー内に存在することが必須です。例：

```
switch# ping mpls nil-fec labels 1,2 ?
output Output options
switch# ping mpls nil-fec labels1,2
^
% Invalid command at '^' marker.
```

CLI を使用して SRTE ポリシーからの発信情報で ping を実行する

次の CLI コマンドを使用して、ping を実行します。

```
switch# ping mpls nil-fec policy name policy1
switch# ping mpls nil-fec policy endpoint 2.0.0.1 color 16
```

CLI を使用した明示的な発信情報でのトレースルートの実行

次の CLI コマンドを使用して、トレースルートを実行します。

```
switch# ping mpls nil-fec labels 2001,2000 output interface e1/1 nexthop 4.0.0.2
```

CLI を使用して SRTE ポリシーからの発信情報で traceroute を実行する

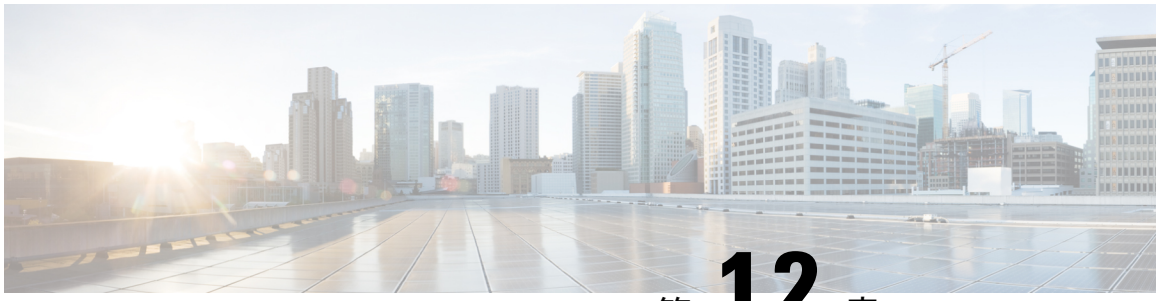
次の CLI コマンドを使用して、トレースルートを実行します。

```
switch# traceroute mpls nil-fec policy name policy1
switch# traceroute mpls nil-fec policy endpoint 2.0.0.1 color 16
```

統計情報の表示

次のコマンドを使用して、ローカル MPLS OAM サービスによって送信されたエコー要求に関する統計情報を表示します。

```
show mpls oam echo statistics
```



第 12 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 9508 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 9508 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 9508 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。

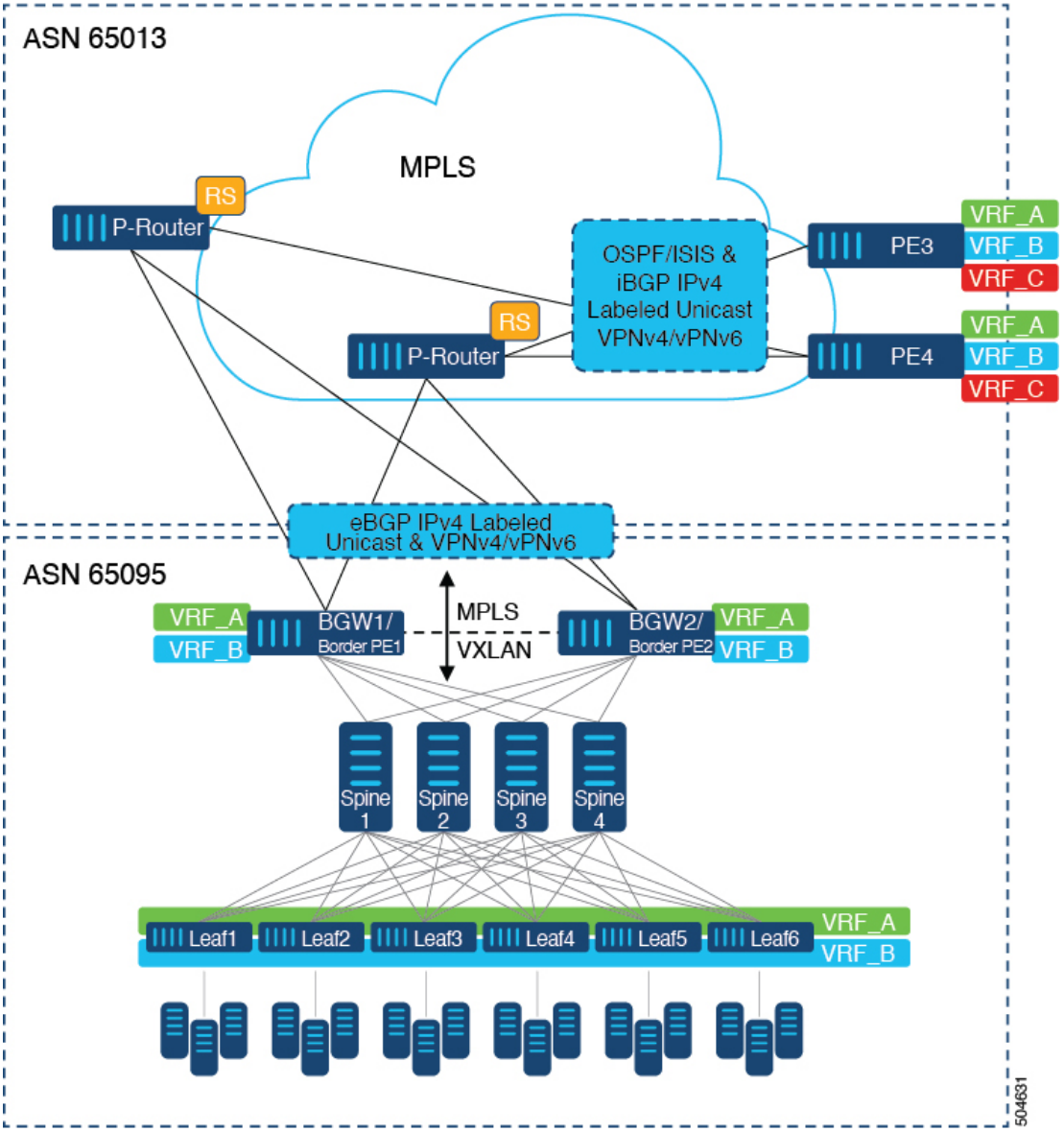
- **InterAS オプション B ライト** – InterAS オプション B 機能のサポートは、Cisco NX-OS 6.2(2) リリースでは制限されています。ライト詳細は、「InterAS オプション B（ライト バージョン）の構成」セクションに記載されています。

EVPN と L3VPN (MPLS) のシームレスな統合の構成に関する情報

データセンター（DC）展開では、EVPN コントロール プレーン ラーニング、マルチテマルチテナント、シームレスモビリティ、冗長性、水平スケーリングが容易になるなどの利点から、VXLAN EVPN を採用しています。同様に、コアネットワークはそれぞれの機能を持つさまざまなテクノロジーに移行します。ラベル配布プロトコル（LDP）およびレイヤ3 VPN（L3VPN）を備えたMPLSは、データセンターを相互接続する多くのコアネットワークに存在します。

VXLAN EVPNにデータセンター（DC）が確立され、マルチテナント対応のトランスポートを必要とするコアネットワークでは、シームレスな統合が自然に必要になります。さまざまなコントロールプレーンプロトコルとカプセル化（ここではVXLANからMPLSベースのコアネットワークまで）をシームレスに統合するために、Cisco Nexus 9000シリーズスイッチは、データセンターとコアルータ（プロバイダールータまたはプロバイダーエッジルータ）。

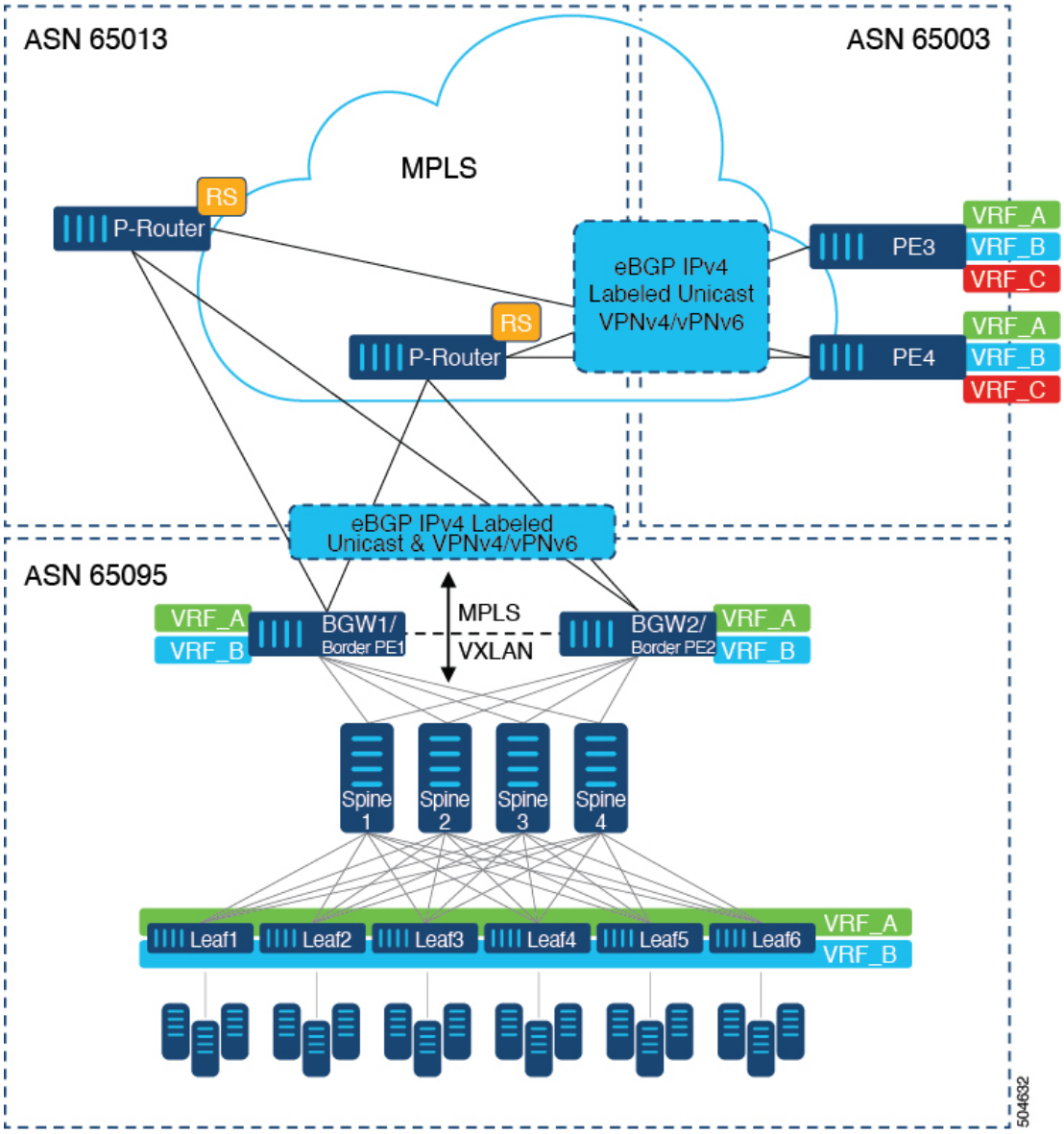
図 *17 : DC*からコアネットワークドメインへの分離を使用したトポロジ



上の図では、VXLAN EVPNを実行する単一のデータセンターファブリックが示されています。データセンターに存在する VRF（VRF_A、VRF_B）は、MPLS を実行する WAN /コア上で拡張する必要があります。データセンター ファブリック ボーダー スイッチは、L3VPN（VPNv4/VPNv6）を使用して VXLAN BGP EVPN と MPLS ネットワークを相互接続するボーダー ゲートウェイ/ボーダー プロバイダ エッジ（BGW1/ボーダー PE1、BGW2/ボーダー PE2）として機能します。BPE は、IPv4 ラベル付きユニキャストと VPNv4 / VPNv6 アドレス ファミリ（AF）を使用して、eBGPを介してプロバイダ ルータ（P-Router）と相互接続されます。P ルータは、前述の AFの BGP ルート リフレクタとして機能し、iBGP を介して MPLS プロバイダ エッジ（PE3、PE4）に必要なルートをリレーします。コントロール プレーンとしての BGP の使用に加えて、同じ自律システム（AS）内の MPLS ノード間では、ラベル配布に IGP（OSPF または ISIS）が使用されます。上の図に示すPE（PE3、PE4）から、Inter-ASオプションAを使

用して、データセンターまたはコアネットワークVRFを別の外部ネットワークに拡張できます。この図では 1 つのデータセンターのみを示していますが、MPLS ネットワークを使用して複数のデータセンター ファブリックを相互接続できます。

図 *18 :* コア ネットワーク内の複数の管理ドメイン



別の導入シナリオは、コアネットワークが複数の管理ドメインまたは自律システム（AS）に分かれている場合です。上の図では、VXLAN EVPNを実行する単一のデータセンターファブリックが示されています。データセンターに存在する VRF（VRF_A、VRF_B）は、MPLS を実行する WAN /コア上で拡張する必要があります。データセンター ファブリック ボーダー スイッチは、L3VPN（VPNv4/VPNv6）を使用して VXLAN BGP EVPN と MPLS ネットワークを相互接続するボーダー ゲートウェイ/ボーダープロバイダエッジ（BGW1/ボーダー PE1、BGW2/ボーダー PE2）として機能します。BPE は、IPv4 ラベル付きユニキャストと VPNv4 / VPNv6

アドレス ファミリ（AF）を使用して、eBGPを介してプロバイダ ルータ（P-Router）と相互接続されます。P ルータは前述の AF の BGP ルート サーバとして機能し、eBGP を介して MPLS プロバイダ エッジ（PE3、PE4）に必要なルートをリレーします。 MPLS ノード間では、他のコントロールプレーンプロトコルは使用されません。前のシナリオと同様に、PE（PE3、PE4）は Inter-AS オプション A で動作して、データセンターまたはコア ネットワーク VRF を外部ネットワークに拡張できます。この図では1つのデータセンターのみを示していますが、MPLS ネットワークを使用して複数のデータセンター ファブリックを相互接続できます。

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 には、次の注意事項と制限事項があります。

- InterAS オプション B は、BGP コンフェデレーション AS ではサポートされていません。
- InterAS オプション B は、-R ライン カード搭載の Cisco Nexus 9500 プラットフォーム スイッチでサポートされます。
- Cisco NX-OS リリース 10.3(2)F 以降、InterAS オプション B（BGP-3107 または RFC 3107 の実装）は、-FX または -FX または-GX2 ラインカードで Nexus 9300-FX/FX2/FX3/GX/GX2 および Cisco 9500 プラットフォーム スイッチでサポートされますが、次の制限があります。
  - PUSH 操作の InterAS ラベルのインポジション（IP から MPLS または VxLAN へのカプセル化解除、および InterAS ラベルの MPLS カプセル化）のみがサポートされます。
  - InterAS ラベルの MPLS ラベル SWAP 動作はサポートされず、MPLS スイッチングは行われません。

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	**import l2vpn evpn reoriginate** 例： `switch(config-router-neighbor-af)# import l2vpn evpn reoriginate`	標準のルートターゲット識別子と一致するルートターゲット識別子を持つレイヤ 3 BGP EVPN NLRI からのルーティング情報のインポートを構成し、このルーティング情報を、スティッチング ルートターゲット識別子に割り当てる再発信の後に、BGP EVPN ネイバーへエクスポートします。
ステップ 9	**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。

	コマンドまたはアクション	目的
ステップ 10	**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 11	**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 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。

EVPN と L3VPN (MPLS) のシームレスな統合の構成

Border Provider Edge (Border PE) の次の手順では、VXLAN ドメインから MPLS ドメインへのルートをインポートして、他の方向へのルートを再開始します。

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`	グローバル設定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	**nv overlay evpn** 例： `switch(config)# nv overlay evpn`	VXLAN を有効にします。
ステップ 4	**feature bgp** 例： `switch(config)# feature bgp`	BGP を有効にします。
ステップ 5	**feature mpls l3vpn** 例： `switch(config)# feature mpls l3vpn`	レイヤ 3 VPN を有効にします。

	コマンドまたはアクション	目的
ステップ 6	**feature interface-vlan** 例： `switch(config)# feature interface-vlan`	VLAN インターフェイスを有効にします。
ステップ 7	**feature vn-segment-vlan-based** 例： `switch(config)# feature vn-segment-vlan-based`	VLAN ベースの VN セグメントを有効にします
ステップ 8	**feature nv overlay** 例： `switch(config)# feature nv overlay`	VXLAN を有効にします。
ステップ 9	**router bgp** *autonomous-system-number* 例： `switch(config)# router bgp 65095`	BGP を設定します。 *autonomous-system-number* の値は 1ー4294967295 です。
ステップ 10	**address-family ipv4 unicast** 例： `switch(config-router)# address-family ipv4 unicast`	IPv4 のアドレス ファミリを設定します。
ステップ 11	**network** *address* 例： `switch(config-router-af)# network 10.51.0.51/32`	MPLS-SR ドメイン向けに BGP にプレフィックスを挿入します。 （注） Border PE での MPLS-SR トンネルデポジションのすべての実行可能なネクストホップは、network ステートメントを介してアドバタイズする必要があります（/32 のみ）。
ステップ 12	**allocate-label all** 例： `switch(config-router-af)# allocate-label all`	network ステートメントによって挿入されたすべてのプレフィックスのラベル割り当てを設定します。
ステップ 13	**exit** 例： `switch(config-router-af)# exit`	コマンド モードを終了します。
ステップ 14	**neighbor** *address* **remote-as** *number* 例：	ルート リフレクター—に対して iBGP ネイバーの IPv4 アドレスおよびリモート

	コマンドまたはアクション	目的
	`switch(config-router)# neighbor 10.95.0.95 remote-as 65095`	自律システム（AS）番号を定義します。
ステップ 15	**update-source** *type/id* 例： `switch(config-router)# update-source loopback0`	eBGP ピアリングのインターフェイスを定義します。
ステップ 16	**address-family l2vpn evpn** 例： `switch(config-router)# address-family l2vpn evpn`	L2VPN EVPN キャスト アドレス ファミリを設定します。
ステップ 17	**send-community both** 例： `switch(config-router-af)# send-community both`	BGP ネイバーのコミュニティを設定します。
ステップ 18	**import vpn unicast reoriginate** 例： `switch(config-router-af)# import vpn unicast reoriginate`	新しい Route-Target でルートを再発信します。オプションのルートマップを使用するように拡張できます。
ステップ 19	**exit** 例： `switch(config-router-af)# exit`	コマンド モードを終了します。
ステップ 20	**neighbor** *address* **remote-as** *number* 例： `switch(config-router)# neighbor 10.51.131.131 remote-as 65013`	P ルーターに対して eBGP ネイバーの IPv4 アドレスおよびリモート自律システム（AS）番号を定義します。
ステップ 21	**update-source** *type/id* 例： `switch(config-router)# update-source Ethernet1/1`	eBGP ピアリングのインターフェイスを定義します。
ステップ 22	**address-family ipv4 labeled-unicast** 例： `switch(config-router)# address-family ipv4 labeled-unicast`	IPv4 ラベル付きユニキャストのアドレス ファミリを設定します。
ステップ 23	**send-community both** 例： `switch(config-router-af)# send-community both`	BGP ネイバーのコミュニティを設定します。

	コマンドまたはアクション	目的
ステップ 24	**exit** 例： `switch(config-router-af)# exit`	コマンド モードを終了します。
ステップ 25	**neighbor** *address* **remote-as** *number* 例： `switch(config-router)# neighbor 10.131.0.131 remote-as 65013`	eBGP ネイバーの IPv4 アドレスおよびリモート自律システム（AS）番号を定義します。
ステップ 26	**update-source** *type/id* 例： `switch(config-router)# update-source loopback0`	eBGP ピアリングのインターフェイスを定義します。
ステップ 27	**ebgp-multihop** *number* 例： `switch(config-router)# ebgp-multihop 5`	リモート ピアにマルチホップ TTL を指定します。*number* の範囲は 2 ～ 255 です。
ステップ 28	**address-family vpnv4 unicast** 例： `switch(config-router)# address-family vpnv4 unicast`	VPNv4 または VPNv6 のアドレス ファミリを設定します。
ステップ 29	**send-community both** 例： `switch(config-router-af)# send-community both`	BGP ネイバーのコミュニティを設定します。
ステップ 30	**import l2vpn evpn reoriginate** 例： `switch(config-router-af)# import l2vpn evpn reoriginate`	新しい Route-Target でルートを再発信します。オプションのルートマップを使用するように拡張できます。
ステップ 31	**exit** 例： `switch(config-router-af)# exit`	コマンド モードを終了します。

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。

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。
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 アドレスを設定し、ルータ BGP ネイバー 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 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 コンフィギュレーション モードを終了し、BGP コンフィギュレーション 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 アドレスを設定し、ルータ BGP 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	**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。

	コマンドまたはアクション	目的
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。

EVPN と L3VPN（MPLS）のシームレスな統合の構成例

シナリオ：DC から コア ネットワーク ドメイン分離および MPLS ネットワーク内 IGP

次に示すのは、VXLAN ドメインから MPLS ドメインへ、および逆方向にルートをインポートおよび再発信するために必要な CLI 設定の例です。サンプルCLI設定は、それぞれのロールに必要な設定のみを示しています。

ボーダー PE

```
hostname BL51-N9336FX2
install feature-set mpls

feature-set mpls

feature bgp
feature mpls l3vpn
feature ospf
feature interface-vlan
feature vn-segment-vlan-based
feature nv overlay

nv overlay evpn

mpls label range 16000 23999 static 6000 8000

vlan 2000
  vn-segment 50000

vrf context VRF_A
  vni 50000
  rd auto
  address-family ipv4 unicast
    route-target both auto
    route-target both auto evpn
```

```
    route-target import 50000:50000
    route-target export 50000:50000
  address-family ipv6 unicast
    route-target both auto
    route-target both auto evpn
    route-target import 50000:50000
    route-target export 50000:50000

interface Vlan2000
  no shutdown
  vrf member VRF_A
  no ip redirects
  ip forward
  ipv6 address use-link-local-only
  no ipv6 redirects

interface nve1
  no shutdown
  host-reachability protocol bgp
  source-interface loopback1
  member vni 50000 associate-vrf

interface Ethernet1/1
  description TO_P-ROUTER
  ip address 10.51.131.51/24
  mpls ip forwarding
  no shutdown

interface Ethernet1/36
  description TO_SPINE
  ip address 10.95.51.51/24
  ip router ospf 10 area 0.0.0.0
  no shutdown

interface loopback0
  description ROUTER-ID
  ip address 10.51.0.51/32
  ip router ospf UNDERLAY area 0.0.0.0

interface loopback1
  description NVE-LOOPBACK
  ip address 10.51.1.51/32
  ip router ospf UNDERLAY area 0.0.0.0

router ospf UNDERLAY
  router-id 10.51.0.51

router bgp 65095
  address-family ipv4 unicast
    network 10.51.0.51/32
    allocate-label all
!
  neighbor 10.95.0.95
    remote-as 65095
    update-source loopback0
    address-family l2vpn evpn
      send-community
      send-community extended
      import vpn unicast reoriginate
!
  neighbor 10.51.131.131
    remote-as 65013
    update-source Ethernet1/1
    address-family ipv4 labeled-unicast
```

```
      send-community
      send-community extended
!
  neighbor 10.131.0.131
    remote-as 65013
    update-source loopback0
    ebgp-multihop 5
    address-family vpnv4 unicast
      send-community
      send-community extended
      import l2vpn evpn reoriginate
    address-family vpnv6 unicast
      send-community
      send-community extended
      import l2vpn evpn reoriginate

!
  vrf VRF_A
    address-family ipv4 unicast
      redistribute direct route-map fabric-rmap-redist-subnet
```

P ルーター

```
hostname P131-N9336FX2
install feature-set mpls

feature-set mpls

feature bgp
feature isis
feature mpls l3vpn

mpls label range 16000 23999 static 6000 8000

route-map RM_NH_UNCH permit 10
  set ip next-hop unchanged

interface Ethernet1/1
  description TO_BORDER-PE
  ip address 10.51.131.131/24
  ip router isis 10
  mpls ip forwarding
  no shutdown

interface Ethernet1/11
  description TO_PE
  ip address 10.52.131.131/24
  ip router isis 10
  mpls ip forwarding
  no shutdown

interface loopback0
  description ROUTER-ID
  ip address 10.131.0.131/32
  ip router isis 10

router isis 10
  net 49.0000.0000.0131.00
  is-type level-2
  address-family ipv4 unicast
    segment-routing mpls

router bgp 65013
  event-history detail
  address-family ipv4 unicast
```

```
    allocate-label all
!
  neighbor 10.51.131.51
    remote-as 65095
    update-source Ethernet1/1
    address-family ipv4 labeled-unicast
      send-community
      send-community extended
!
  neighbor 10.51.0.51
    remote-as 65095
    update-source loopback0
    ebgp-multihop 5
    address-family vpnv4 unicast
      send-community
      send-community extended
      route-map RM_NH_UNCH out
    address-family vpnv6 unicast
      send-community
      send-community extended
      route-map RM_NH_UNCH out
!
  neighbor 10.52.131.52
    remote-as 65013
    update-source Ethernet1/11
    address-family ipv4 labeled-unicast
      send-community
      send-community extended
!
  neighbor 10.52.0.52
    remote-as 65013
    update-source loopback0
    address-family vpnv4 unicast
      send-community
      send-community extended
      route-reflector-client
      route-map RM_NH_UNCH out
    address-family vpnv6 unicast
      send-community
      send-community extended
      route-reflector-client
      route-map RM_NH_UNCH out
```

プロバイダー エッジ（PE）

```
hostname L52-N93240FX2
install feature-set mpls

feature-set mpls

feature bgp
feature isis
feature mpls l3vpn

mpls label range 16000 23999 static 6000 8000

vrf context VRF_A
  rd auto
  address-family ipv4 unicast
    route-target import 50000:50000
    route-target export 50000:50000
  address-family ipv6 unicast
    route-target import 50000:50000
    route-target export 50000:50000
```

```
interface Ethernet1/49
  description TO_P-ROUTER
  ip address 10.52.131.52/24
  ip router isis 10
  mpls ip forwarding
  no shutdown

interface loopback0
  description ROUTER-ID
  ip address 10.52.0.52/32
  ip router isis 10

router isis 10
  net 49.0000.0000.0052.00
  is-type level-2
  address-family ipv4 unicast
    segment-routing mpls

router bgp 65013
  address-family ipv4 unicast
    network 10.52.0.52/32
    allocate-label all
!
  neighbor 10.52.131.131
    remote-as 65013
    update-source Ethernet1/49
    address-family ipv4 labeled-unicast
      send-community
      send-community extended
!
  neighbor 10.131.0.131
    remote-as 65013
    update-source loopback0
    address-family vpnv4 unicast
      send-community
      send-community extended
    address-family vpnv6 unicast
      send-community
      send-community extended
!
  vrf VRF_A
    address-family ipv4 unicast
      redistribute direct route-map fabric-rmap-redist-subnet
```

シナリオ：DC からコアへ、およびコア ネットワーク ドメイン分離内（MPLS ネットワーク内の eBGP）

次に示すのは、VXLAN ドメインから MPLS ドメインへ、および逆方向にルートをインポートおよび再発信するために必要な CLI 設定の例です。サンプル CLI 構成は、シナリオ 1 とは異なるノード（P-Router ロールと Provider Edg（PE）ロール）のみを示しています。ボーダーPEは両方のシナリオで同じままです。

P ルーター

```
hostname P131-N9336FX2
install feature-set mpls

feature-set mpls

feature bgp
feature mpls l3vpn
```

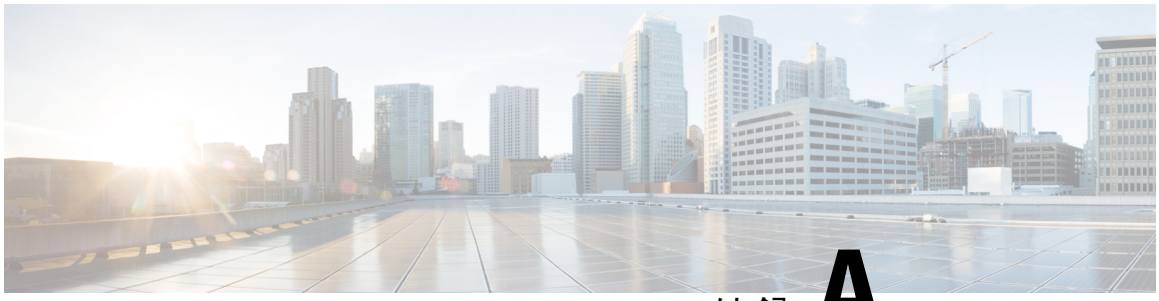
```
mpls label range 16000 23999 static 6000 8000

route-map RM_NH_UNCH permit 10
  set ip next-hop unchanged

interface Ethernet1/1
  description TO_BORDER-PE
  ip address 10.51.131.131/24
  mpls ip forwarding
  no shutdown

interface Ethernet1/11
  description TO_PE
  ip address 10.52.131.131/24
  mpls ip forwarding
  no shutdown

interface loopback0
  description ROUTER-ID
  ip address 10.131.0.131/32
  ip router isis 10

router bgp 65013
  event-history detail
  address-family ipv4 unicast
    network 10.131.0.131/32
    allocate-label all
!
  address-family vpnv4 unicast
    retain route-target all
  address-family vpnv6 unicast
    retain route-target all
!
  neighbor 10.51.131.51
    remote-as 65095
    update-source Ethernet1/1
    address-family ipv4 labeled-unicast
      send-community
      send-community extended
!
  neighbor 10.51.0.51
    remote-as 65095
    update-source loopback0
    ebgp-multihop 5
    address-family vpnv4 unicast
      send-community
      send-community extended
      route-map RM_NH_UNCH out
    address-family vpnv6 unicast
      send-community
      send-community extended
      route-map RM_NH_UNCH out
!
  neighbor 10.52.131.52
    remote-as 65003
    update-source Ethernet1/11
    address-family ipv4 labeled-unicast
      send-community
      send-community extended
!
  neighbor 10.52.0.52
    remote-as 65003
    update-source loopback0
    ebgp-multihop 5
```

```
    address-family vpnv4 unicast
      send-community
      send-community extended
      route-map RM_NH_UNCH out
    address-family vpnv6 unicast
      send-community
      send-community extended
      route-map RM_NH_UNCH out
```

プロバイダー エッジ（PE）

```
hostname L52-N93240FX2
install feature-set mpls

feature-set mpls

feature bgp
feature mpls l3vpn

mpls label range 16000 23999 static 6000 8000

vrf context VRF_A
  rd auto
  address-family ipv4 unicast
    route-target import 50000:50000
    route-target export 50000:50000
  address-family ipv6 unicast
    route-target import 50000:50000
    route-target export 50000:50000

interface Ethernet1/49
  description TO_P-ROUTER
  ip address 10.52.131.52/24
  mpls ip forwarding
  no shutdown

interface loopback0
  description ROUTER-ID
  ip address 10.52.0.52/32
  ip router isis 10

router bgp 65003
  address-family ipv4 unicast
    network 10.52.0.52/32
    allocate-label all
!
  neighbor 10.52.131.131
    remote-as 65013
    update-source Ethernet1/49
    address-family ipv4 labeled-unicast
      send-community
      send-community extended
!
  neighbor 10.131.0.131
    remote-as 65013
    update-source loopback0
    ebgp-multihop 5
    address-family vpnv4 unicast
      send-community
      send-community extended
    address-family vpnv6 unicast
      send-community
      send-community extended
!
  vrf VRF_A
```

```
address-family ipv4 unicast
  redistribute direct route-map fabric-rmap-redist-subnet
```



付録 A

MPLS レイヤ 3 VPN ラベル割り当ての設定

この章では、Cisco Nexus 9508 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。

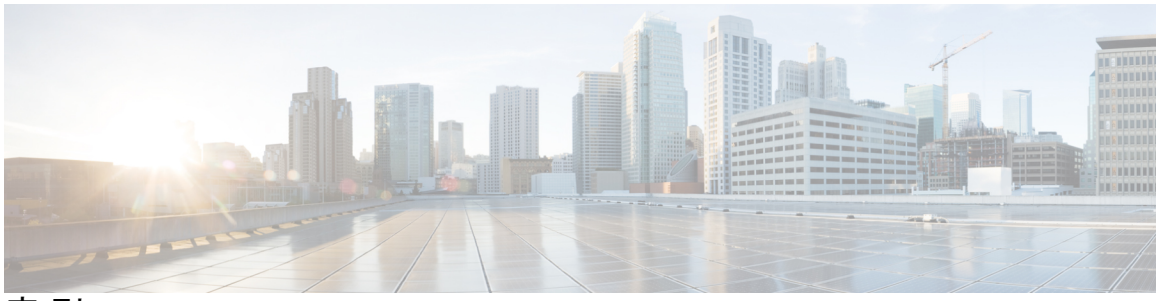
- ラベル スイッチングでサポートされる IETF RFC （345 ページ）

ラベル スイッチングでサポートされる IETF RFC

次の表に、デバイスのラベル スイッチングでサポートされている IETF RFC を示します。

RFC	タイトル
RFC 3107	『*Carrying Label Information in BGP-4*』
RFC 7752	*BGP* を使用したリンクステートおよびトラフィ リング（*TE*）のノースバウンド配信
RFC 8029	マルチプロトコル ラベル スイッチド（*MPLS*）ラ の障害の検出。
RFC 8287	セグメント ルーティング *(SR)* のラベル スイッチ *Ping/Traceroute IGP* プレフィックスおよび *MPLS* ンを持つ *IGP* 隣接セグメント識別子 *(SID)*。
Draft-ietf-idr-bgpls-segment-routing-epe-05	セグメント ルーティング *BGP* 出力ピア エンジニ 拡張 *draft-ietf-idr-bgpls-segment-routing-epe-05*

MPLS レイヤ 3 VPN ラベル割り当ての設定



索 引

T

V

し

ね

ろ

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。