



## MPLS L3VPN の概要

MPLS VPN を定義する前に、VPN 全般を定義する必要があります。VPN の説明を次に示します。

- パブリック インフラストラクチャを介してプライベート ネットワーク サービスを提供する、IP ベースのネットワーク
- インターネットまたはその他のパブリック ネットワークやプライベート ネットワークを介してプライベートに相互通信できる一連のサイト

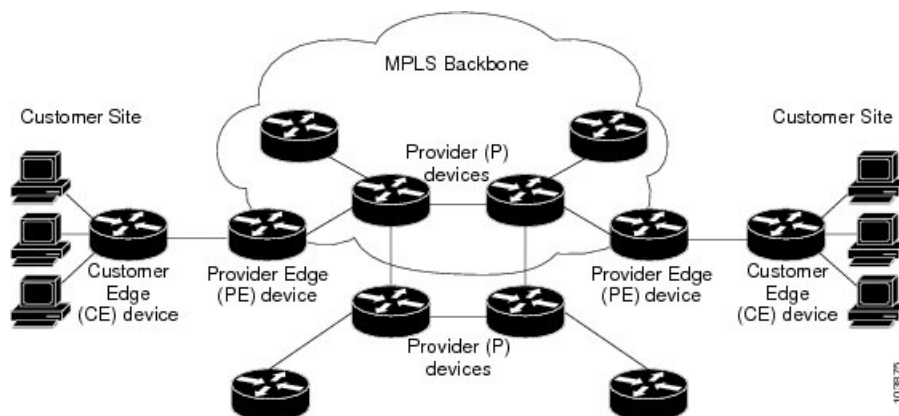
通常の VPN は、完全メッシュのトンネル、または相手先固定接続（PVC）を VPN 内のすべてのサイトに設定することで作成されます。このタイプの VPN は、新しいサイトを追加した場合に VPN 内の各エッジ デバイスを変更する必要があるため、維持または拡張が簡単ではありません。

MPLS ベースの VPN は、レイヤ 3 に作成され、ピア モデルに基づきます。ピア モデルによって、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。サービス 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、カスタマー側では何もする必要がありません。

MPLS VPN の管理や拡張は、従来の VPN よりも簡単です。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。

次の図に、基本的な MPLS VPN のトポロジを示します。

図 1: 基本的な MPLS VPN トポロジ



MPLS VPN の基本的なコンポーネントは次のとおりです。

- 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、ルーティングされるパケットに VPN ラベルを付加しません。VPN ラベルは正しいプライベート ネットワークまたはカスタマー エッジルータにデータ パケットを送信するために使用されます。
- PE ルータ：着信パケットを受信す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。MPLS コア ラベルも付加します。PE ルータは、CE ルータに直接接続します。
- カスタマー (C) ルータ：インターネット サービス プロバイダー (ISP) または企業ネットワークのルータ。
- 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 ISP 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。
- [MPLS L3VPN の仕組み \(2 ページ\)](#)
- [MPLS レイヤ 3 VPN の実装方法 \(10 ページ\)](#)
- [VRF-lite \(46 ページ\)](#)
- [セグメント ルーティングを使用した MPLS L3VPN サービス \(50 ページ\)](#)
- [MPLS L3VPN の実装：リファレンス \(55 ページ\)](#)

## MPLS L3VPN の仕組み

MPLS VPN 機能は、MPLS ネットワークのエッジでイネーブルになっています。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。
- VPN バージョン 4 (VPNv4) ルートに CE ルーティング情報を変換する。

- 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 VPNv4 ルートを交換する。

## MPLS L3VPN の主要コンポーネント

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。

- 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、VPN コミュニティのすべてのメンバのリストです。VPN ルート ターゲットは、各 VPN コミュニティ メンバに設定する必要があります。
- VPN コミュニティ PE ルータのマルチプロトコル BGP (MP-BGP) ピアリング : MP-BGP は、VPN コミュニティのすべてのメンバに VRF 到達可能性情報を伝播します。MP-BGP ピアリングは、VPN コミュニティのすべての PE ルータに設定されている必要があります。
- MPLS 転送 : MPLS は、VPN 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 つの指定されたサイトを複数の VPN 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 VPN 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。

詳細については、[MPLS L3VPN の主要コンポーネント : 詳細 \(56 ページ\)](#) を参照してください。

## MPLS L3VPN の制約事項

での MPLS L3VPN の実装には次の制約事項があります。

- Cisco NCS 540 シリーズ ルータがサポートする ECMP パスは 16 個のみです。
- 出力 MTU を超える MPLS パケットのフラグメンテーションはサポートされていません。IP から MPLS インポジションでのフラグメンテーションもサポートされていません。そのため、MPLS コア内のすべてのインターフェイスで最大 MTU (9216) 値を使用することをお勧めします。
- L3VPN プレフィックスのルックアップでは常に 1 つのパスを生成します。IGP レベル または BGP レベルに複数のパスがある場合、各レベルでのパスの選択はコントロールプレーンでプレフィックス ハッシュを使用して実行されます。選択したパスは、データプレーンでプログラミングされます。
- TTL の伝達を無効にすることはできません。TTL の伝達は、IP から MPLS と MPLS から IP で常に行われます。

上記の特定の制約事項とは別に、MPLS L3VPN の実装に関する一般的な制約事項は にも適用されます。

- MPLS VPN の自律システムまたはサブ自律システム間に eBGP ルーティングを設定する場合は、マルチホップ VPN-IPv4 eBGP はサポートされません。
- MPLS VPN は、IPv4 アドレス ファミリーだけをサポートします。

次のプラットフォームの制約事項は、Cisco NCS 540 シリーズ ルータのみに適用されます。

- MPLS-TE の統計情報はサポートされていません。
- MPLS の統計情報は、**show mpls forwarding** コマンド出力ではサポートされておらず、任意の MPLS の統計情報は表示されません。

IPv4 ルートおよび MPLS ラベルを交換する ASBR に MPLS VPN Inter-AS を設定する場合は、次の制約事項が適用されます。

- eBGP マルチホップが設定されたネットワークでは、非隣接ルータ間にラベル スイッチド パス (LSP) を設定する必要があります。



(注) BGP スピーカーに接続する物理インターフェイスは FIB および MPLS をサポートする必要があります。

## L3VPN の Inter-AS サポート

ここでは、次の内容について説明します。

### Inter-AS サポート：概要

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ルーティング プロトコルを使用する、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。

VPN が大規模になるにつれて、その要件も多くなります。場合によっては、VPN が異なる地理的エリアの異なる自律システムに存在する必要があります。また、一部の VPN は、複数のサービス プロバイダーにまたがって設定する必要があります (オーバーラッピング VPN)。VPN の複雑性および場所に関係なく、自律システム間の接続はシームレスでなければなりません。

MPLS VPN Inter-AS には次の利点があります。

- VPN が複数のサービス プロバイダー バックボーンをまたがることが可能。

異なる自律システムを実行する複数のサービス プロバイダーが、共同で同じエンドカスタマーに MPLS VPN サービスを提供できます。あるカスタマー サイトから開始し、さまざまな VPN サービス プロバイダー バックボーンを通過して、同じカスタマーの別のサイトに到達するように VPN を設定できます。以前は、MPLS VPN は、単一の BGP 自律システム サービス プロバイダー バックボーンだけを通過できました。この機能は、複数の自

律システムが、サービス プロバイダーのカスタマー サイト間に連続性がありシームレスなネットワークを形成できます。

- VPN が異なるエリアに存在可能。

サービスプロバイダーは、異なる地理的エリアに VPN を作成できます。すべての VPN トラフィックフローを（エリア間で）1 箇所のポイントを通してさせるようにすると、エリア間のネットワーク トラフィックのレートをより適切に制御できます。

- iBGP メッシングを最適化するための連合が可能。

自律システム内の内部ボーダー ゲートウェイ プロトコル（iBGP）メッシングがより整理され、管理しやすくなります。自律システムを複数の異なるサブ自律システムに分割した後、それらを単一の連合に分類できます。連合を形成するサブ自律システム間でのラベル付き VPN-IPv4 ネットワーク層到達可能性情報（NLRI）の交換がサポートされているため、サービス プロバイダーはこの機能を使用して、連合全体で MPLS VPN を提供できます。

## Inter-AS および ASBR

異なるサービスプロバイダーの異なる自律システムは、VPN-IPv4 アドレスの形式で IPv4 NLRI および IPv6 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。その後、Interior Gateway Protocol（IGP）によって、各 VPN および各自律システム全体に、VPN-IPv4 プレフィックスのネットワーク層情報が配布されます。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。

- 自律システム内では、ルーティング情報は IGP を使用して共有されます。
- 自律システム間では、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異なる自律システム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ドメイン間ルーティング システムを設定できます。

eBGP の主な機能は、自律システムのルートに関する情報を含む、自律システム間のネットワーク到達可能性情報を交換することです。自律システムは、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。

MPLS VPN でサポートされている相互 AS 設定には次のものがあります。

- プロバイダー間 VPN：異なるボーダー エッジルータによって接続された、2 つ以上の自律システムを含む MPLS VPN。各自律システムは、eBGP を使用してルートを交換します。自律システム間では、IGP 情報（ルーティング情報）は交換されません。
- BGP 連合：単一の自律システムを複数のサブ自律システムに分割してから、指定された単一の連合として分類した MPLS VPN。ネットワークでは、連合は単一の自律システムとして認識されます。異なる自律システム内のピアは、eBGP セッションを介して通信しますが、これらのピアは iBGP ピアである場合と同様にルート情報を交換できます。

## 連合

連合とは、複数のサブ自律システムをグループ化したものです。連合を使用することによって、自律システム内のピアデバイスの合計数を減らすことができます。連合では、自律システムが複数のサブ自律システムに分割され、自律システムに連合識別子が割り当てられます。VPNは、異なる自律システムまたは連合を形成する複数のサブ自律システムで実行される、複数のサービス プロバイダーにまたがることができます。

連合において、各サブ自律システムと他のサブ自律システムとの関係は、フルメッシュになっています。サブ自律システム間の通信は、Open Shortest Path First (OSPF) や Intermediate System-to-Intermediate System (IS-IS) などの IGP を使用して行われます。また、各サブ自律システムには、他のサブ自律システムへの EBGP 接続もあります。連合 eBGP (CEBGP) ボーダーエッジルータは、指定されたサブ自律システム間で next-hop-self アドレスを転送します。next-hop-self アドレスによって、BGP では、プロトコルでネクストホップを選択するのではなく、ネクストホップとして指定されたアドレスを使用することが強制されます。

次の 2 つの方法で、異なるサブ自律システムに連合を設定できます。

- next-hop-self アドレスが CEBGP ボーダーエッジルータ間でだけ転送されるようにルータを設定します (双方向)。サブ自律システム ボーダーのサブ自律システム (IBGP ピア) では、next-hop-self アドレスは転送されません。各サブ自律システムは、単一の IGP ドメインとして実行されます。ただし、CEBGP ボーダーエッジルータアドレスは、IGP ドメイン内で認識されます。
- next-hop-self アドレスが CEBGP ボーダーエッジルータ間 (双方向)、およびサブ自律システム ボーダーの iBGP ピア内で転送されるようにルータを設定します。各サブ自律システムは、単一の IGP ドメインとして実行されますが、ドメイン内の PE ルータ間で next-hop-self アドレスの転送もします。CEBGP ボーダーエッジルータアドレスは、IGP ドメイン内で認識されます。

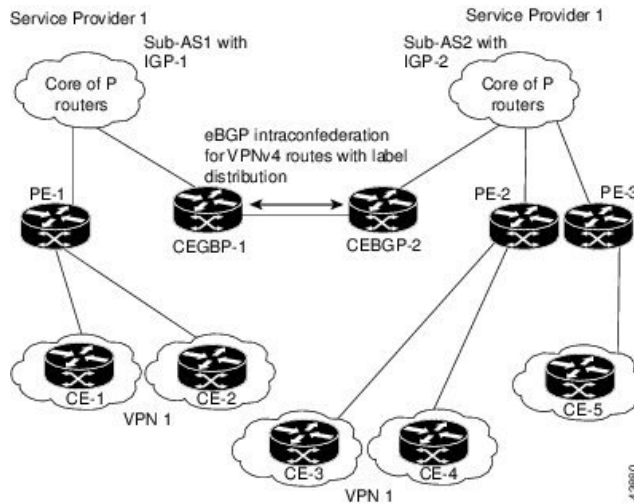


(注) 連合内の 2 つのサブ自律システム間での eBGP 接続の図で、2 つの自律システムがルートを交換し、パケットを転送する方法を示します。連合内のサブ自律システムでは、ルートの交換およびパケットの転送に同様の方法が使用されます。

次の図に、一般的な MPLS VPN 連合設定を示します。この設定の場合：

- 2 つの CEBGP ボーダーエッジルータは、2 つの自律システム間で VPN-IPv4 アドレスおよびラベルを交換します。
- 配布元ルータはネクストホップアドレスおよびラベルを変更して、next-hop-self アドレスを使用します。
- IGP-1 および IGP-2 では、CEBGP-1 と CEBGP-2 のアドレスが認識されます。

図 2: 連合内の 2 つのサブ自律システム間での eBGP 接続



この連合設定の特徴は次のとおりです。

- CEBGP ボーダー エッジルータは、サブ自律システム間の隣接ピアとして機能します。サブ自律システムは、eBGP を使用してルート情報を交換します。
- 各 CEBGP ボーダー エッジルータ (CEBGP-1、CEBGP-2) は、ルートを次のサブ自律システムに配布する前に、ルータのラベルを割り当てます。CEBGP ボーダー エッジルータは、BGP のマルチプロトコル拡張を使用して、VPN-IPv4 アドレスとしてルートを配布します。ラベルおよび VPN 識別子は、NLRI の一部として符号化されます。
- 各 PE および CEBGP ボーダー エッジルータは、ルートを再配布する前に、各 VPN-IPv4 アドレスプレフィックスに独自のラベルを割り当てます。CEBGP ボーダー エッジルータは、ラベル付き IPV-IPv4 アドレスを交換します。ラベルには、(eBGP ネクストホップ属性の値として) next-hop-self アドレスが含まれています。サブ自律システム内では、CEBGP ボーダー エッジルータ アドレスが iBGP ネイバー全体に配布され、2 つの CEBGP ボーダー エッジルータが両方の連合で認識されます。

## MPLS VPN Inter-AS BGP ラベルの配布



(注) このセクションは、Inter-AS over IP トンネルの場合には適用されません。

ASBR が、プロバイダー エッジ (PE) ルータの MPLS ラベル付き IPv4 ルートを交換するように MPLS VPN Inter-AS ネットワークを設定できます。ルートリフレクタ (RR) は、マルチホップマルチプロトコル外部ボーダー ゲートウェイ プロトコル (eBGP) を使用して VPN-IPv4 ルートを交換します。このように Inter-AS システムを設定する方法は、多くの場合 MPLS VPN Inter-AS BGP ラベル配布と呼ばれます。

ASBR が IPv4 ルートおよび MPLS ラベルを交換するように Inter-AS システムを設定すると次の利点が得られます。

- ASBR にすべての VPN-IPv4 ルートを格納する必要がなくなります。ルートリフレクタを使用して VPN-IPv4 ルートを格納し、PE ルータに転送すると、ASBR がすべての VPN-IPv4 ルートを保持し、VPN-IPv4 ラベルに基づいてルートを転送する設定と比較して、改善されたスケーラビリティが得られます。
- ルートリフレクタが VPN-IPv4 ルートを持つようにすることでも、ネットワーク境界での設定は簡素化されます。
- 非 VPN コア ネットワークが、VPN トラフィックの中継ネットワークとして動作できます。MPLS ラベルの付いた IPv4 ルートを非 MPLS VPN サービスプロバイダー経由で送信できます。
- 隣接ラベルスイッチルータ（LSR）間で他のラベル配布プロトコルが必要なくなります。隣接する 2 つの LSR が BGP ピアでもある場合、BGP で MPLS ラベルの配布を実行できます。これら 2 つの LSR 間で、他のラベル配布プロトコルは必要ありません。

## IPv4 ルートと MPLS ラベルの交換



(注) このセクションは、Inter-AS over IP トンネルの場合には適用されません。

VPN サービスプロバイダー ネットワークを設定して、MPLS ラベル付き IPv4 ルートを交換できます。次のように VPN サービスプロバイダー ネットワークを設定できます。

- ルートリフレクタは、マルチホップマルチプロトコル eBGP を使用して VPN-IPv4 ルートを交換します。この設定では、自律システムをまたがってネクストホップ情報および VPN ラベルが維持されます。
- ローカル PE ルータ（たとえば次の図の PE1）は、リモート PE ルータ（PE2）のルートおよびラベル情報を把握する必要があります。

この情報は、次のいずれかの方法で PE ルータおよび ASBR 間で交換できます。

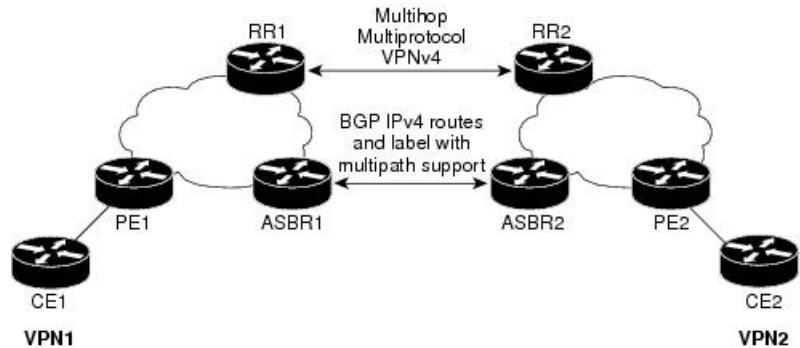
- 内部ゲートウェイプロトコル（IGP）とラベル配布プロトコル（LDP）：ASBR は、eBGP から学習した IPv4 ルートおよび MPLS ラベルを IGP および LDP に再配布できます。IGP および LDP から eBGP への再配布も可能です。
- 内部ボーダーゲートウェイプロトコル（iBGP）IPv4 ラベル配布：ASBR および PE ルータは、直接 iBGP セッションを使用して、VPN-IPv4 と IPv4 ルートおよび MPLS ラベルを交換できます。

または、ルートリフレクタが、ASBR から学習した IPv4 ルートおよび MPLS ラベルを VPN の PE ルータに反映できます。この学習した IPv4 ルートおよび MPLS ラベルの反映は、ルートリフレクタと IPv4 ルートおよび MPLS ラベルを交換するように ASBR をイネーブルにすることで実現します。ルートリフレクタは、VPN-IPv4 ルートも VPN の PE ルータに反映します。たとえば、VPN1 では、RR1 は、学習した VPN-IPv4 ルート、および ASBR1 から学習した IPv4 ルートと MPLS ラベルを PE1 に反映します。ルートリフレクタを使用して VPN-IPv4 ルートを



格納し、それらを PE ルータおよび ASBR 経由で転送することによって、スケーラブルな設定が可能となります。

図 3: eBGP および iBGP を使用してルートと MPLS ラベルを配布する VPN



## BGP ルーティング情報

BGP ルーティング情報には、次の項目が含まれています。

- 宛先の IP アドレスであるネットワーク番号（プレフィックス）。
- ルートがローカルルータに到達するために通過する他の AS のリスト（自律システム（AS）パス）。リスト内の最初の AS はローカルルータに最も近いシステムです。リスト内の最後の AS はローカルルータから最も遠いシステムであり、通常、ルートの始点となる AS です。
- ネクスト ホップなどの、AS パスについての他の情報を提供するパス属性。

## BGP メッセージと MPLS ラベル

MPLS ラベルは、ルータが送信するアップデートメッセージに含まれています。ルータ間では、次のタイプの BGP メッセージが交換されます。

- オープン メッセージ：ルータがネイバー ルータとの間で TCP 接続を確立すると、ルータ間でオープンメッセージが交換されます。このメッセージには、ルータが属する自律システムの数とメッセージを送信したルータの IP アドレスが含まれています。
- アップデート メッセージ：ルータのルートが新規作成、変更、または切断された場合、ルータはネイバールータにアップデートメッセージを送信します。このメッセージには、使用可能なルートの IP アドレスのリストを含む NLRI が含まれます。アップデートメッセージには、使用できなくなったすべてのルートが含まれています。また、アップデートメッセージには、使用可能なパスと使用できないパスの両方のパス属性と長さも含まれています。アップデートメッセージでは、VPN-IPv4 ルートのラベルは RFC 2858 の規定に従って符号化されます。また、アップデートメッセージでは、IPv4 ルートのラベルは RFC 3107 の規定に従って符号化されます。
- キープアライブ メッセージ：ルータ間では、ネイバー ルータがルーティング情報を交換可能であるかどうかを判断するためにキープアライブメッセージが交換されます。ルータは、定期的にこれらのメッセージを送信します（シスコルータのデフォルトは 60 秒で

す)。キープアライブ メッセージには、ルーティング データは含まれていません。メッセージ ヘッダーのみが含まれています。

- 通知メッセージ：ルータでエラーが検出されると、通知メッセージが送信されます。

## ルートをを使用した MPLS ラベルの送信

BGP (eBGP および iBGP) でルートを配布するときに、そのルートにマッピングされている MPLS ラベルも配布できます。ルートの MPLS ラベルマッピング情報は、ルートについての情報を含む BGP 更新メッセージによって伝送されます。ネクスト ホップが変わらない場合は、ラベルも維持されます。

両方の BGP ルータで `show bgp neighbors ip-address` コマンドを発行すると、それらのルータでルートとともに MPLS ラベルを送信できることがルータ間で相互にアドバタイズされます。ルータ間で MPLS ラベルを送信可能であると正常にネゴシエーションされると、それらのルータからのすべての発信 BGP アップデートに MPLS ラベルが追加されます。

# MPLS レイヤ 3 VPN の実装方法

MPLS L3VPN の実装には、次の主要タスクが含まれています。

- [コア ネットワークの設定 \(11 ページ\)](#)
- [MPLS VPN カスタマーの接続 \(17 ページ\)](#)

## MPLS L3VPN の実装の前提条件

MPLS L3VPN を設定する前提条件は次のとおりです。

- 次のコマンドに適切なタスク ID を含むタスク グループに関連付けられているユーザ グループに属している必要があります。
  - BGP
  - IGP
  - MPLS
  - MPLS レイヤ 3 VPN
- ユーザ グループの割り当てが原因でコマンドを使用できないと考えられる場合、AAA 管理者に連絡してください。
- MPLS レイヤ 3 VPN を設定するには、ルータは MPLS 転送および転送情報ベース (FIB) をサポートしている必要があります。

## コア ネットワークの設定

MPLS LDP コアを介して MPLS L3VPN サービスを転送するネットワーク トポロジを検討します。

CE1 - HundredGigE 0/9/0/0 - HundredGigE 0/9/0/0 - PE - HundredGigE 0/9/0/1 - HundredGigE 0/9/0/1  
- P Node - HundredGigE 0/9/0/0 - HundredGigE 0/9/0/0 - PE2 - HundredGigE 0/9/0/1 - hundredGigE  
0/9/0/1 - CE2

コア ネットワークの設定には、次の主要タスクが含まれています。

- [MPLS VPN カスタマーのニーズの評価 \(11 ページ\)](#)
- [コアにおけるルーティング プロトコルの設定 \(11 ページ\)](#)
- [コアにおける MPLS の設定 \(12 ページ\)](#)
- [FIB がコア上で有効になっているかどうかの確認 \(14 ページ\)](#)
- [PE ルータおよびルート リフレクタでの Multiprotocol BGP の設定 \(14 ページ\)](#)

## MPLS VPN カスタマーのニーズの評価

MPLS VPN を設定する前に、コア ネットワーク トポロジは最良のサービスを MPLS VPN カスタマーに提供できるように識別される必要があります。次に示すタスクは、コア ネットワーク トポロジを識別するのに役立ちます。

- ネットワークのサイズを識別します。  
必要なルータとポートの数を決定するには、次の内容を特定します。
  - サポートするカスタマーの数
  - カスタマーごとに必要な VPN 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 (VRF) の数
- コアで必要なルーティング プロトコルを決定します。
- MPLS VPN コアで BGP ロード シェアリングおよび冗長パスが必要であるかどうかを決定します。

## コアにおけるルーティング プロトコルの設定

コアでは、RIP、OSPF、または IS-IS をルーティング プロトコルとして使用できます。

PE1 - HundredGigE 0/9/0/1 - HundredGigE 0/9/0/1 - P Node - HundredGigE 0/9/0/0 - HundredGigE  
0/9/0/0 - PE2

### 設定例

次に、ルーティング プロトコルとしてコアで OSPF を設定するステップの例を示します。

```
Router-PE1#configure
Router-PE1(config)#router ospf dc-core
```

```
Router-PE1(config-ospf)#address-family ipv4 unicast
Router-PE1(config-ospf)#area 1
Router-PE1(config-ospf-ar)#
Router-PE1(config-ospf-ar)#
Router-PE1(config-ospf-vrf-ar-if)#commit
```

## 実行コンフィギュレーション

```
router ospf dc-core
router-id 13.13.13.1
address-family ipv4 unicast
area 1

interface HundredGigE 0/9/0/1
!
!
!
```

## 確認

- OSPF ネイバーを確認し、*State* に「FULL」と表示されることを確認します。

```
Router-PE1# show ospf neighbor
Neighbors for OSPF dc-core

Neighbor ID      Pri   State           Dead Time   Address          Interface
16.16.16.1       1     FULL/DR         00:00:34    191.22.1.2      HundredGigE 0/9/0/1
    Neighbor is up for 1d18h

Total neighbor count: 1
```

## 関連項目

- [MPLS レイヤ 3 VPN の実装方法（10 ページ）](#)

## 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（LDP）を設定する必要があります。

また、コアでセグメントルーティングを使用して MPLS L3VPN サービスを転送することもできます。詳細については、[MPLS コアでのセグメントルーティングの設定（51 ページ）](#)を参照してください。

## 設定例

次に、MPLS コアで LDP を設定するステップの例を示します。

```
Router-PE1#configure
Router-PE1(config)#mpls ldp
```

```
Router-PE1(config-ldp)#router-id 13.13.13.1
Router-PE1(config-ldp)#address-family ipv4
Router-PE1(config-ldp-af)#exit
Router-PE1(config-ldp)#
Router-PE1(config-ldp)#
Router-PE1(config-ldp)#interface hundredGigE 0/9/0/0
Router-PE1(config-ldp)#commit
```

PE2 と P ルータでもこの設定を繰り返します。

## 実行コンフィギュレーション

```
mpls ldp
router-id 13.13.13.1
address-family ipv4
!

interface hundredGigE 0/9/0/0
!
!
```

## 確認

- ネイバー（16.16.16.1）がコアインターフェイスを介して起動していることを確認します。

```
Router-PE1#show mpls ldp neighbor
Peer LDP Identifier: 16.16.16.1:0
  TCP connection: 16.16.16.1:47619 - 13.13.13.1:646
  Graceful Restart: No
  Session Holdtime: 180 sec
  State: Oper; Msgs sent/rcvd: 40395/35976; Downstream-Unsolicited
  Up time: 2w2d
  LDP Discovery Sources:
    IPv4: (1)
      HundredGigE 0/9/0/0
    IPv6: (0)
  Addresses bound to this peer:
    IPv4: (6)
      10.64.98.32      87.0.0.2      88.88.88.14      50.50.50.50
      178.0.0.1       192.1.1.1
    IPv6: (0)
```

## 関連項目

- [MPLS レイヤ 3 VPN の実装方法（10 ページ）](#)

MPLS LDP の設定の詳細については、『*MPLS Configuration Guide for Cisco NCS 560 Series Routers*』の「*Implementing MPLS Label Distribution Protocol*」の章を参照してください。

## FIB がコア上で有効になっているかどうかの確認

転送情報ベース（FIB）は、プロバイダーエッジ（PE）ルータを含むコア内のすべてのルータでイネーブルにする必要があります。FIB が有効になっているかどうかを判断する方法については、『*IP Addresses and Services Configuration Guide for Cisco NCS 560 Series Routers*』の「*Implementing Cisco Express Forwarding*」モジュールを参照してください。

## PE ルータおよびルート リフレクタでの Multiprotocol BGP の設定

Multiprotocol BGP（MP-BGP）は VRF 到達可能性情報を VPN コミュニティのすべてのメンバに伝達します。VPN コミュニティ内のすべての PE ルータに MP-BGP ピアリングを設定する必要があります。

### 設定例

次に、PE1 上で MP-BGP を設定する例を示します。PE2 のループバックアドレス（20.20.20.1）を PE1 のネイバーとして指定します。同様に、PE2 ノードでもこの設定を実行する必要があります。その場合は、PE1 のループバック アドレス（13.13.13.1）を PE2 のネイバーとして指定します。

```
Router-PE1#configure
Router-PE1(config)#router bgp 2001
Router-PE1(config-bgp)#bgp router-id 13.13.13.1
Router-PE1(config-bgp)#address-family ipv4 unicast
Router-PE1(config-bgp-af)#exit
Router-PE1(config-bgp)#address-family vpnv4 unicast
Router-PE1(config-bgp-af)#exit
Router-PE1(config-bgp)#neighbor 20.20.20.1
Router-PE1(config-bgp-nbr)#remote-as 2001
Router-PE1(config-bgp-nbr)#update-source loopback 0
Router-PE1(config-bgp-nbr)#address-family ipv4 unicast
Router-PE1(config-bgp-nbr-af)#exit
Router-PE1(config-bgp-nbr)#address-family vpnv4 unicast
Router-PE1(config-bgp-nbr-af)#exit
Router-PE1(config-bgp-nbr)#exit
/* VRF configuration */
Router(config-bgp)# vrf vrf1601
Router-PE1(config-bgp-vrf)#rd 2001:1601
Router-PE1(config-bgp-vrf)#address-family ipv4 unicast
Router-PE1(config-bgp-vrf-af)#label mode per-vrf
Router-PE1(config-bgp-vrf-af)#redistribute connected
Router-PE1(config-bgp-vrf-af)#commit
```

### 実行コンフィギュレーション

```
router bgp 2001
  bgp router-id 13.13.13.1
  address-family ipv4 unicast
  !
  address-family vpnv4 unicast
  !
  neighbor 20.20.20.1
    remote-as 2001
    update-source Loopback0
```

```

address-family vpnv4 unicast
!
address-family ipv4 unicast
!
!
vrf vrf1601
rd 2001:1601
address-family ipv4 unicast
  label mode per-vrf
  redistribute connected
!
!

```

## 確認

- BGP の状態が確立されているかどうか、およびリモート AS と ローカル AS が同じ値（この例では 2001）を示すかどうかを確認します。

```
Router-PE1#show bgp neighbor
```

```

BGP neighbor is 20.20.20.1
  Remote AS 2001, local AS 2001, internal link
  Remote router ID 20.20.20.1
  BGP state = Established, up for 1d19h
  NSR State: None
  Last read 00:00:04, Last read before reset 00:00:00
  Hold time is 60, keepalive interval is 20 seconds
  Configured hold time: 60, keepalive: 30, min acceptable hold time: 3
  Last write 00:00:16, attempted 19, written 19
  Second last write 00:00:36, attempted 19, written 19
  Last write before reset 00:00:00, attempted 0, written 0
  Second last write before reset 00:00:00, attempted 0, written 0
  Last write pulse rcvd Apr 12 10:31:20.739 last full not set pulse count 27939
  Last write pulse rcvd before reset 00:00:00
  Socket not armed for io, armed for read, armed for write
  Last write thread event before reset 00:00:00, second last 00:00:00
  Last KA expiry before reset 00:00:00, second last 00:00:00
  Last KA error before reset 00:00:00, KA not sent 00:00:00
  Last KA start before reset 00:00:00, second last 00:00:00
  Precedence: internet
  Non-stop routing is enabled
  Multi-protocol capability received
  Neighbor capabilities:
    Route refresh: advertised (old + new) and received (old + new)
    Graceful Restart (GR Awareness): received
    4-byte AS: advertised and received
    Address family IPv4 Unicast: advertised and received
    Address family VPNv4 Unicast: advertised and received
  Received 25595 messages, 0 notifications, 0 in queue
  Sent 8247 messages, 0 notifications, 0 in queue
  Minimum time between advertisement runs is 0 secs
  Inbound message logging enabled, 3 messages buffered
  Outbound message logging enabled, 3 messages buffered

For Address Family: IPv4 Unicast
  BGP neighbor version 484413
  Update group: 0.4 Filter-group: 0.3 No Refresh request being processed
  Inbound soft reconfiguration allowed
  NEXT_HOP is always this router
  AF-dependent capabilities:
    Outbound Route Filter (ORF) type (128) Prefix:

```

```

    Send-mode: advertised, received
    Receive-mode: advertised, received
    Graceful Restart capability received
    Remote Restart time is 120 seconds
    Neighbor did not preserve the forwarding state during latest restart
    Additional-paths Send: advertised and received
    Additional-paths Receive: advertised and received
    Route refresh request: received 1, sent 1
    Policy for incoming advertisements is pass-all
    Policy for outgoing advertisements is pass-all
    24260 accepted prefixes, 24260 are bestpaths
    Cumulative no. of prefixes denied: 0.
    Prefix advertised 2000, suppressed 0, withdrawn 0
    Maximum prefixes allowed 1048576
    Threshold for warning message 75%, restart interval 0 min
    AIGP is enabled
    An EoR was received during read-only mode
    Last ack version 484413, Last synced ack version 0
    Outstanding version objects: current 0, max 1
    Additional-paths operation: Send and Receive
    Send Multicast Attributes
    Advertise VPNv4 routes enabled with defaultReoriginate,disable Local with
    stitching-RT option

For Address Family: VPNv4 Unicast
BGP neighbor version 798487
Update group: 0.2 Filter-group: 0.1 No Refresh request being processed
AF-dependent capabilities:
    Graceful Restart capability received
    Remote Restart time is 120 seconds
    Neighbor did not preserve the forwarding state during latest restart
    Additional-paths Send: advertised and received
    Additional-paths Receive: advertised and received
    Route refresh request: received 0, sent 0
    29150 accepted prefixes, 29150 are bestpaths
    Cumulative no. of prefixes denied: 0.
    Prefix advertised 7200, suppressed 0, withdrawn 0
    Maximum prefixes allowed 2097152
    Threshold for warning message 75%, restart interval 0 min
    AIGP is enabled
    An EoR was received during read-only mode
    Last ack version 798487, Last synced ack version 0
    Outstanding version objects: current 0, max 1
    Additional-paths operation: Send and Receive
    Send Multicast Attributes
    Advertise VPNv4 routes enabled with defaultReoriginate,disable Local with
    stitching-RT option

Connections established 1; dropped 0
Local host: 13.13.13.1, Local port: 35018, IF Handle: 0x00000000
Foreign host: 20.20.20.1, Foreign port: 179
Last reset 00:00:00

```

- PE1 ですべての IP アドレスが PE2 から学習されているかを確認します。

```
Router-PE1#show bgp vpnv4 unicast
```

```

BGP router identifier 13.13.13.1, local AS number 2001
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0 RD version: 0

```



```

BGP main routing table version 798487
BGP NSR Initial initsync version 15151 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network        Next Hop           Metric LocPrf Weight Path
Route Distinguisher: 2001:1601 (default for vrf vrf1601)
*> 20.13.1.1/32    192.13.26.5                        0 7501 i
*> 20.13.1.2/32    192.13.26.5                        0 7501 i
*> 20.13.1.3/32    192.13.26.5                        0 7501 i
*> 20.13.1.4/32    192.13.26.5                        0 7501 i
*> 20.13.1.5/32    192.13.26.5                        0 7501 i
*>i20.14.1.1/32    14.14.14.1                        100 0 8501 i
*>i20.14.1.2/32    14.14.14.1                        100 0 8501 i
*>i20.14.1.3/32    14.14.14.1                        100 0 8501 i
*>i20.14.1.4/32    14.14.14.1                        100 0 8501 i
*>i20.14.1.5/32    14.14.14.1                        100 0 8501 i

```

### 関連項目

- [コア ネットワークの設定 \(11 ページ\)](#)
- [カスタマー接続を有効にする PE ルータ上での VRF の定義 \(18 ページ\)](#)

Multiprotocol BGP の詳細については、*BGP Configuration Guide for Cisco NCS 560 Series Routers* を参照してください。

### 関連コマンド

## MPLS VPN カスタマーの接続

MPLS VPN カスタマーの接続には、次の主要なタスクが含まれています。

- [カスタマー接続を有効にする PE ルータ上での VRF の定義 \(18 ページ\)](#)
- [各 VPN カスタマー用の PE ルータ上での VRF インスタンスの設定 \(19 ページ\)](#)
- PE ルータと CE ルータ間のルーティング プロトコルの設定

次のいずれかのオプションを使用します。

- [PE ルータと CE ルータ間のルーティング プロトコルとしての BGP の設定 \(20 ページ\)](#)
- [PE ルータと CE ルータ間のルーティング プロトコルとしての RIPv2 の設定 \(24 ページ\)](#)
- [PE ルータと CE ルータ間のスタティック ルートの設定 \(25 ページ\)](#)
- [PE ルータと CE ルータ間のルーティング プロトコルとしての OSPF の設定 \(26 ページ\)](#)

## カスタマー接続を有効にする PE ルータ上での VRF の定義

VPN ルーティングおよび転送 (VRF) では、PE ルータに接続されているカスタマー サイトの VPN メンバーシップを定義します。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。VRF には、そのサイトがメンバとなっている VPN 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。VPN ルーティング情報の配布は、BGP 拡張コミュニティによって実装される VPN ルートターゲットコミュニティを使用して制御されます。

### 設定例

次に、VRF インスタンス (vrf1601) を設定し、インポートとエクスポートのルートとターゲット (2001:1601) を指定します。インポート ルート ポリシーは、ローカル VPN にインポート可能なポリシーです。エクスポート ルート ポリシーは、ローカル VPN からエクスポート可能なポリシーです。インポート ルートターゲット設定では、エクスポートされたルートのルートターゲットのうちの 1 つがローカル VPN インポート ターゲットの 1 つと一致する場合にエクスポートされた VPN ルートを VPN にインポートできます。ルートが他の PE ルータにアドバタイズされる際に、エクスポート ルート ターゲットが拡張コミュニティとしてルートとともに送信されます。

```
Router-PE1#configure
Router-PE1 (config) #vrf vrf1601
Router-PE1 (config-vrf) #address-family ipv4 unicast
Router-PE1 (config-vrf-af) #import route-target
Router-PE1 (config-vrf-af-import-rt) #2001:1601
Router-PE1 (config-vrf-af-import-rt) #exit
Router-PE1 (config-vrf-af) #export route-target
Router-PE1 (config-vrf-af-export-rt) #2001:1601
Router-PE1 (config-vrf-af-export-rt) #commit
```

この VRF インスタンスが対応する BGP インスタンスに関連付けられます。

### 実行コンフィギュレーション

```
vrf vrf1601
  address-family ipv4 unicast
    import route-target
      2001:1601
    !
  export route-target
    2001:1601
  !
!
```

### 確認

インポートとエクスポートのルート ターゲットを確認します。

```
Router-PE1#show vrf vrf1601
```

VRF	RD	RT	AFI	SAFI
vrf1601	2001:1601	import 2001:1601	IPv4	Unicast
		export 2001:1601	IPv4	Unicast

### 関連項目

- [各 VPN カスタマー用の PE ルータ上での VRF インスタンスの設定 \(19 ページ\)](#)
- [PE ルータおよびルート リフレクタでの Multiprotocol BGP の設定 \(14 ページ\)](#)

## 各 VPN カスタマー用の PE ルータ上での VRF インスタンスの設定

VRF インスタンスを作成したら、その VRF インスタンスを PE ルータ上のインターフェイスまたはサブインターフェイスと関連付ける必要があります。



- (注) インターフェイスの VRF の割り当て、削除、または変更を行う前に、IPv4 または IPv6 アドレスをインターフェイスから削除する必要があります。これを事前に行わない場合、IP インターフェイス上での VRF 変更はすべて拒否されます。

### 設定例

この例では、IP アドレス *192.13.26.6* を PE1 ルータ上のインターフェイス (*HundredGigE0/9/0/1.1601*) に割り当て、VRF インスタンス *vrf1601* をそのインターフェイスに関連付けます。

```
Router-PE1#configure
Router-PE1(config)#interface HundredGigE 0/9/0/1.1601
Router-PE1(config-if)#vrf vrf1601
Router-PE1(config-if)#ipv4 address 192.13.26.6 255.255.255.252
Router-PE1(config-if)#encapsulation dot1q 1601
Router-PE1(config)#commit
```

### 実行コンフィギュレーション

```
interface HundredGigE 0/9/0/1.1601
vrf vrf1601
ipv4 address 192.13.26.6 255.255.255.252
encapsulation dot1q 1601
!
```

### 確認

- VRF が関連付けられたインターフェイスが起動していることを確認します。

```
Router-PE1#show ipv4 vrf vrf1601 interface
interface HundredGigE 0/9/0/1.1601 is Up, ipv4 protocol is Up
```

```

Vrf is vrf1601 (vrfid 0x60000001)
Internet address is 192.13.26.6/30
MTU is 1518 (1500 is available to IP)
Helper address is not set
Multicast reserved groups joined: 224.0.0.2 224.0.0.1
Directed broadcast forwarding is disabled
Outgoing access list is not set
Inbound common access list is not set, access list is not set
Proxy ARP is disabled
ICMP redirects are never sent
ICMP unreachable are always sent
ICMP mask replies are never sent
Table Id is 0xe0000001

```

### 関連項目

- [カスタマー接続を有効にする PE ルータ上での VRF の定義（18 ページ）](#)

## PE ルータと CE ルータ間のルーティング プロトコルの設定

### PE ルータと CE ルータ間のルーティング プロトコルとしての BGP の設定

BGP は、各 VPN の VPN-IPv4 プレフィックスに関する到達可能性情報を配布します。PE から PE または PE からルート リフレクタ（RR）へのセッションは iBGP セッション、PE から CE へのセッションは eBGP セッションです。PE から CE への eBGP セッションは、直接または間接的に接続できます（eBGP マルチホップ）。

CE-1 HundredGigE 0/9/0/0 - HundredGigE 0/9/0/0 PE-1

### 設定例

次に、PE ルータと CE ルータ間のルーティング プロトコルとして BGP を設定するステップの例を示します。この例のルートポリシー、*pass-all* は、設定してから付加する必要があります。

#### PE1 :

```

Router-PE1#configure
Router-PE1(config)#router bgp 2001
Router-PE1(config-bgp)#bgp router-id 13.13.13.1
Router-PE1(config-bgp)#address-family ipv4 unicast
Router-PE1(config-bgp-af)#exit
Router-PE1(config-bgp)#address-family vpnv4 unicast
Router-PE1(config-bgp-af)#exit
/* VRF configuration */
Router-PE1(config-bgp)#vrf vrf1601
Router-PE1(config-bgp-vrf)#rd 2001:1601
Router-PE1(config-bgp-vrf)#address-family ipv4 unicast
Router-PE1(config-bgp-vrf-af)#label mode per-vrf
Router-PE1(config-bgp-vrf-af)#redistribute connected
Router-PE1(config-bgp-vrf-af)#exit
Router-PE1(config-bgp-vrf)#neighbor 192.13.26.5
Router-PE1(config-bgp-vrf-nbr)#remote-as 7501
Router-PE1(config-bgp-vrf-nbr)#address-family ipv4 unicast
Router-PE1(config-bgp-vrf-nbr-af)#route-policy pass-all in
Router-PE1(config-bgp-vrf-nbr-af)#route-policy pass-all out

```

```
Router-PE1 (config-bgp-vrf-nbr-af) #commit
```

**CE1 :**

```
Router-CE1#configure
Router-CE1 (config) #router bgp 2001
Router-CE1 (config-bgp) #bgp router-id 8.8.8.1
Router-CE1 (config-bgp) #address-family ipv4 unicast
Router-CE1 (config-bgp-af) #exit
Router-CE1 (config-bgp) #address-family vpnv4 unicast
Router-CE1 (config-bgp-af) #exit
Router-CE1 (config-bgp) #neighbor 192.13.26.6
Router-CE1 (config-bgp-nbr) #remote-as 2001
Router-CE1 (config-bgp-nbr) #address-family ipv4 unicast
Router-CE1 (config-bgp-nbr-af) #route-policy pass-all in
Router-CE1 (config-bgp-nbr-af) #route-policy pass-all out
Router-CE1 (config-bgp-nbr-af) #commit
```

## 実行コンフィギュレーション

**PE1 :**

```
router bgp 2001
  bgp router-id 13.13.13.1
  address-family ipv4 unicast
  !
  address-family vpnv4 unicast
  !
  vrf vrf1601
    rd 2001:1601
    address-family ipv4 unicast
      label mode per-vrf
      redistribute connected
    !
  neighbor 192.13.26.5
    remote-as 7501
    address-family ipv4 unicast
      route-policy pass-all in
      route-policy pass-all out
    !
  !
  !
```

**CE1 :**

```
router bgp 7501
  bgp router-id 8.8.8.1
  address-family ipv4 unicast
  !
  address-family vpnv4 unicast
  !
  neighbor 192.13.26.6
    remote-as 2001
    address-family ipv4 unicast
      route-policy pass-all in
      route-policy pass-all out
    !
  !
```

!

## 確認

## • PE1 :

```

Router-PE1#show bgp neighbor
BGP neighbor is 192.13.26.5
  Remote AS 6553700, local AS 2001, external link
  Administratively shut down
  Remote router ID 192.13.26.5
  BGP state = Established
  NSR State: None
  Last read 00:00:04, Last read before reset 00:00:00
  Hold time is 60, keepalive interval is 20 seconds
  Configured hold time: 60, keepalive: 30, min acceptable hold time: 3
  Last write 00:00:16, attempted 19, written 19
  Second last write 00:00:36, attempted 19, written 19
  Last write before reset 00:00:00, attempted 0, written 0
  Second last write before reset 00:00:00, attempted 0, written 0
  Last write pulse rcvd Apr 12 10:31:20.739 last full not set pulse count 27939
  Last write pulse rcvd before reset 00:00:00
  Socket not armed for io, armed for read, armed for write
  Last write thread event before reset 00:00:00, second last 00:00:00
  Last KA expiry before reset 00:00:00, second last 00:00:00
  Last KA error before reset 00:00:00, KA not sent 00:00:00
  Last KA start before reset 00:00:00, second last 00:00:00
  Precedence: internet
  Non-stop routing is enabled
  Graceful restart is enabled
  Restart time is 120 seconds
  Stale path timeout time is 360 seconds
  Enforcing first AS is enabled
  Multi-protocol capability not received
  Received 0 messages, 0 notifications, 0 in queue
  Sent 0 messages, 0 notifications, 0 in queue
  Minimum time between advertisement runs is 30 secs
  Inbound message logging enabled, 3 messages buffered
  Outbound message logging enabled, 3 messages buffered

For Address Family: IPv4 Unicast
  BGP neighbor version 0
  Update group: 0.2 Filter-group: 0.0 No Refresh request being processed
  Inbound soft reconfiguration allowed
  AF-dependent capabilities:
    Outbound Route Filter (ORF) type (128) Prefix:
      Send-mode: advertised
      Receive-mode: advertised
    Graceful Restart capability advertised
      Local restart time is 120, RIB purge time is 600 seconds
      Maximum stalepath time is 360 seconds
    Route refresh request: received 0, sent 0
    Policy for incoming advertisements is pass-all
    Policy for outgoing advertisements is pass-all
    0 accepted prefixes, 0 are bestpaths
    Cumulative no. of prefixes denied: 0.
    Prefix advertised 0, suppressed 0, withdrawn 0
    Maximum prefixes allowed 1048576
    Threshold for warning message 75%, restart interval 0 min
    An EoR was not received during read-only mode
    Last ack version 1, Last synced ack version 0
    Outstanding version objects: current 0, max 0

```

```

Additional-paths operation: None
Advertise VPNv4 routes enabled with defaultReoriginate,disable Local with
stitching-RT option
Advertise VPNv6 routes is enabled with default option

Connections established 1; dropped 0
Local host: 192.13.26.6, Local port: 23456, IF Handle: 0x00000000
Foreign host: 192.13.26.5, Foreign port: 179
Last reset 03:12:58, due to Admin. shutdown (CEASE notification sent - administrative
shutdown)
Time since last notification sent to neighbor: 03:12:58
Notification data sent:
    None
External BGP neighbor not directly connected.

```

#### • CE1 :

```

Router-CE1#show bgp neighbor
BGP neighbor is 192.13.26.6
Remote AS 2001, local AS 6553700, external link
Remote router ID 192.13.26.6
BGP state = Established
NSR State: None
Last read 00:00:04, Last read before reset 00:00:00
Hold time is 60, keepalive interval is 20 seconds
Configured hold time: 60, keepalive: 30, min acceptable hold time: 3
Last write 00:00:16, attempted 19, written 19
Second last write 00:00:36, attempted 19, written 19
Last write before reset 00:00:00, attempted 0, written 0
Second last write before reset 00:00:00, attempted 0, written 0
Last write pulse rcvd Apr 12 10:31:20.739 last full not set pulse count 27939
Last write pulse rcvd before reset 00:00:00
Socket not armed for io, armed for read, armed for write
Last write thread event before reset 00:00:00, second last 00:00:00
Last KA expiry before reset 00:00:00, second last 00:00:00
Last KA error before reset 00:00:00, KA not sent 00:00:00
Last KA start before reset 00:00:00, second last 00:00:00
Precedence: internet
Non-stop routing is enabled
Graceful restart is enabled
Restart time is 120 seconds
Stale path timeout time is 360 seconds
Enforcing first AS is enabled
Multi-protocol capability not received
Received 0 messages, 0 notifications, 0 in queue
Sent 0 messages, 0 notifications, 0 in queue
Minimum time between advertisement runs is 30 secs
Inbound message logging enabled, 3 messages buffered
Outbound message logging enabled, 3 messages buffered

For Address Family: IPv4 Unicast
BGP neighbor version 0
Update group: 0.1 Filter-group: 0.0 No Refresh request being processed
Inbound soft reconfiguration allowed
AF-dependent capabilities:
    Outbound Route Filter (ORF) type (128) Prefix:
        Send-mode: advertised
        Receive-mode: advertised
    Graceful Restart capability advertised
        Local restart time is 120, RIB purge time is 600 seconds
        Maximum stalepath time is 360 seconds

```

```

Route refresh request: received 0, sent 0
Policy for incoming advertisements is pass-all
Policy for outgoing advertisements is pass-all
0 accepted prefixes, 0 are bestpaths
Cumulative no. of prefixes denied: 0.
Prefix advertised 0, suppressed 0, withdrawn 0
Maximum prefixes allowed 1048576
Threshold for warning message 75%, restart interval 0 min
An EoR was not received during read-only mode
Last ack version 1, Last synced ack version 0
Outstanding version objects: current 0, max 0
Additional-paths operation: None

Connections established 0; dropped 0
Local host: 192.13.26.5, Local port: 179, IF Handle: 0x00000000
Foreign host: 192.13.26.6, Foreign port: 23456
Last reset 00:00:00
External BGP neighbor not directly connected.

```

## 関連項目

- [MPLS VPN カスタマーの接続 \(17 ページ\)](#)
- [PE ルータおよびルート リフレクタでの Multiprotocol BGP の設定 \(14 ページ\)](#)

ARP の詳細については、『*BGP Configuration Guide for Cisco NCS 560 Series Routers*』を参照してください。

## PE ルータと CE ルータ間のルーティング プロトコルとしての RIPv2 の設定

### 設定例

次に、PE ルータと CE ルータ間のルーティング プロトコルとして RIPv2 を設定するステップの例を示します。VRF インスタンス `vrf1601` を RIP コンフィギュレーション モードで設定し、それぞれ対応するインターフェイス（PE1 では `HundredGigE 0/9/0/0.1601`、CE1 では `HundredGigE 0/9/0/0.1601`）をその VRF に関連付けます。**redistribute** オプションで、RIP に再配布するルート指定します。

#### PE1 :

```

Router-PE1#configure
Router-PE1(config)#router rip
Router-PE1(config-rip)#vrf vrf1601
Router-PE1(config-rip-vrf)#interface HundredGigE 0/9/0/0.1601
Router-PE1(config-bgp-vrf-if)#exit
Router-PE1(config-bgp-vrf)#redistribute bgp 2001
Router-PE1(config-bgp-vrf)#redistribute connected
Router-PE1(config-bgp-vrf)#commit

```

#### CE1 :

```

Router-CE1#configure
Router-CE1(config)#router rip
Router-CE1(config-rip)#vrf vrf1601

```



```
Router-CE1(config-rip-vrf)#interface HundredGigE 0/9/0/0.1601
Router-CE1(config-bgp-vrf-if)#exit
Router-CE1(config-bgp-vrf)#redistribute connected
Router-CE1(config-bgp-vrf)#commit
```

## 実行コンフィギュレーション

### PE1 :

```
Router-PE1#show running-config router rip
router rip
vrf vrf1601
interface HundredGigE 0/9/0/0.1601
!
redistribute bgp 2001
redistribute connected
!
!
```

### CE1 :

```
Router-CE1#show running-config router rip
router rip
vrf vrf1601
interface HundredGigE 0/9/0/0.1601
!
redistribute connected
!
!
```

## 関連項目

- [MPLS VPN カスタマーの接続 \(17 ページ\)](#)

## PE ルータと CE ルータ間のスタティック ルートの設定

### 設定例

この例では、スタティック ルートを VRF、vrf1601 に割り当てます。

```
Router-PE1#configure
Router-PE1(config)#router static
Router-PE1(config-static)#vrf vrf1601
Router-PE1(config-static-vrf)#address-family ipv4 unicast
Router-PE1(config-static-vrf-afi)#23.13.1.1/32 HundredGigE0/9/0/0.1601 192.13.3.93
Router-PE1(config-static-vrf-afi)#commit
```

CE1 で対応するインターフェイス値を使用して設定を繰り返します。

## 実行コンフィギュレーション

### PE1 :

```

router static
vrf vrf1601
address-family ipv4 unicast
23.13.1.1/32 HundredGigE0/9/0/0.1601 192.13.3.93
!
!
!

```

**CE1 :**

```

router static
vrf vrf1601
address-family ipv4 unicast
23.8.1.2/32 HundredGigE0/9/0/0.1601 192.8.3.94
!
!
!

```

### 関連項目

- [MPLS VPN カスタマーの接続 \(17 ページ\)](#)

### 関連コマンド

- router static

## PE ルータと CE ルータ間のルーティング プロトコルとしての OSPF の設定

PE ルータと CE ルータ間のルーティング プロトコルとして、RIP、OSPF、または ISIS を使用できます。

### 設定例

次に、OSPF ルーティング プロトコルを使用する PE-CE ルーティング セッションを設定するステップの例を示します。VRF インスタンス *vrf1601* を **router ospf** コンフィギュレーション モードで設定します。OSPF プロセスのルータ ID は 13.13.13.1 です。**redistribute** オプションで、OSPF に再配布するルートを指定します。OSPF 領域は *1* に設定し、HundredGigE 0/9/0/0.1601 をその領域に関連付けてルーティングできるようにします。

**PE1 :**

```

Router-PE1#configure
Router-PE1 (config)#router ospf pe-ce-ospf-vrf
Router-PE1 (config-ospf)#router-id 13.13.13.1
Router-PE1 (config-ospf)#vrf vrf1601
Router-PE1 (config-ospf-vrf)#redistribute connected
Router-PE1 (config-ospf-vrf)#redistribute bgp 2001
Router-PE1 (config-ospf-vrf)#area 1
Router-PE1 (config-ospf-vrf-ar)#interface HundredGigE 0/9/0/0.1601
Router-PE1 (config-ospf-vrf-ar)# commit

```

PE2 ノードでもこの設定を繰り返します。

CE1 :

```
Router-CE1#configure
Router-CE1(config)#router ospf ospf pe-ce-1
Router-CE1(config-ospf)#router-id 8.8.8.1
Router-CE1(config-ospf)#vrf vrf1601
Router-CE1(config-ospf-vrf)#area 1
Router-CE1(config-ospf-vrf-ar)#interface HundredGigE 0/9/0/0.1601
Router-CE1(config-ospf-vrf-ar)#commit
```

実行コンフィギュレーション

PE1 :

```
router ospf pe-ce-ospf-vrf
router-id 13.13.13.1
vrf vrf1601
 redistribute connected
 redistribute bgp 2001
area 1
 interface HundredGigE 0/9/0/0.1601
!
!
!
!
```

CE1 :

```
router ospf pe-ce-1
router-id 8.8.8.1
vrf vrf1601
area 1
 interface HundredGigE 0/9/0/0.1601
!
!
!
!
```

関連項目

- [MPLS VPN カスタマーの接続 \(17 ページ\)](#)

## MPLS L3VPN 設定の確認

MPLS L3VPN が正常に設定されるようにするには、次を確認する必要があります。

- [L3VPN トラフィック フローの確認 \(28 ページ\)](#)
- [アンダーレイ \(転送\) の確認 \(28 ページ\)](#)
- [オーバーレイ \(L3VPN\) の確認 \(30 ページ\)](#)

## L3VPN トラフィック フローの確認

- VRF（vrf1601）に関連付けられているラベルの切り替えられたバイト数を確認します。

P ノード :

```
Router-P#show mpls forwarding
Local  Outgoing  Prefix      Outgoing  Next Hop    Bytes
Label  Label      or ID       Interface                Switched
-----
24119  Pop         20.20.20.1/32  Hu0/9/0/0    191.31.1.90  2170204180148
```

PE2 :

```
Router#show mpls forwarding
Local  Outgoing  Prefix      Outgoing  Next Hop    Bytes
Label  Label      or ID       Interface                Switched
-----
24031  Aggregate  vrf1601: Per-VRF Aggr[V]  \
                                         vrf1601                11124125835
```

## アンダーレイ（転送）の確認

- LDP ネイバー接続がそれぞれのネイバーで確立されているかどうかを確認します。

```
Router-PE1#show mpls ldp neighbor
Peer LDP Identifier: 16.16.16.1:0
TCP connection: 16.16.16.1:47619 - 13.13.13.1:646
Graceful Restart: No
Session Holdtime: 180 sec
State: Oper; Msgs sent/rcvd: 40395/35976; Downstream-Unsolicited
Up time: 2w2d
LDP Discovery Sources:
  IPv4: (1)
    hundredGigE 0/9/0/0
  IPv6: (0)
Addresses bound to this peer:
  IPv4: (6)
    10.64.98.32    87.0.0.2      88.88.88.14    50.50.50.50
    178.0.0.1     192.1.1.1
  IPv6: (0)
```

- FIB がラベルの更新を受け取っているかどうかを確認します。

```
Router-PE1#show mpls forwarding
Local  Outgoing  Prefix      Outgoing  Next Hop    Bytes
Label  Label      or ID       Interface                Switched
-----
24036  Pop         16.16.16.1/32  Hu0/9/0/0    191.22.1.2  293294
24037  24165      18.18.18.1/32  Hu0/9/0/0    191.22.1.2  500
24039  24167      20.20.20.1/32  Hu0/9/0/0    191.22.1.2  17872433
        24167      20.20.20.1/32  Hu0/9/0/0    191.22.3.2   6345
```

```
24041 Aggregate vrf1601: Per-VRF Aggr[V] \
                                         vrf1601
7950400999
```

- ハードウェアのラベルが更新されているかどうかを確認します。

```
Router-PE1#show mpls forwarding labels 24001 hardware egress
```

Local Label	Outgoing Label	Prefix or ID	Outgoing Interface	Next Hop	Bytes Switched
24039	24167	20.20.20.1/32	Hu0/9/0/0	191.22.1.2	N/A
	24167	20.20.20.1/32	Hu0/9/0/0	191.22.3.2	N/A

```
Show-data Print at RPLC
```

```
LEAF - HAL pd context :
sub-type : MPLS, ecd_marked:0, has_collapsed_ldi:0
collapse_bwalk_required:0, ecdv2_marked:0
```

```
Leaf H/W Result:
```

```
Leaf H/W Result on NP:0
```

Label	SwitchAction	EgressIf	Programmed
24039	0	0x 200185	<b>Programmed</b>

```
nrLDI eng ctx:
```

```
flags: 0x101, proto: 2, npaths: 0, nbuckets: 1
ldi_tbl_idx: 0xc37e40, ecd_ref_cft: 0
pbts_ldi_tbl_idx: 0x0, fastnrldi:0x0
```

```
NR-LDI H/W Result for path 0 [index: 0xc37e40 (BE), common to all NPs]:
```

```
ECMP Sw Idx: 12811840 HW Idx: 200185 Path Idx: 0
```

```
NR-LDI H/W Result for path 1 [index: 0xc37e41 (BE), common to all NPs]:
```

```
ECMP Sw Idx: 12811841 HW Idx: 200185 Path Idx: 1
```

```
SHLDI eng ctx:
```

```
flags: 0x0, shldi_tbl_idx: 0, num_entries:0
```

```
SHLDI HW data for path 0 [index: 0 (BE)] (common to all NPs):
```

```
Unable to get HW NRLDI Element rc: 1165765120NRLDI Idx: 0
```

```
SHLDI HW data for path 1 [index: 0x1 (BE)] (common to all NPs):
```

```
Unable to get HW NRLDI Element rc: 1165765120NRLDI Idx: 1
```

```
TX H/W Result for NP:0 (index: 0x187a0 (BE)):
```

```
Next Hop Data
```

```
Next Hop Valid: YES
Next Hop Index: 100256
Egress Next Hop IF: 100047
Hw Next Hop Intf: 606
HW Port: 0
Next Hop Flags: COMPLETE
Next Hop MAC: e4aa.5d9a.5f2e
```

```
NHINDEX H/W Result for NP:0 (index: 0 (BE)):
```

```
NhIndex is NOT required on this platform
```

```
NHINDEX STATS: pkts 0, bytes 0 (no stats)
```

```

RX H/W Result on NP:0 [Adj ptr:0x40 (BE)]:
Rx-Adj is NOT required on this platform

TX H/W Result for NP:0 (index: 0x189a8 (BE)):

Next Hop Data
Next Hop Valid:      YES
Next Hop Index:      100776
Egress Next Hop IF:  100208
Hw Next Hop Intf:    607
HW Port:             0
Next Hop Flags:      COMPLETE
Next Hop MAC:        e4aa.5d9a.5f2d

NHINDEX H/W Result for NP:0 (index: 0 (BE)):
NhIndex is NOT required on this platform

NHINDEX STATS: pkts 0, bytes 0 (no stats)

RX H/W Result on NP:0 [Adj ptr:0x40 (BE)]:
Rx-Adj is NOT required on this platform

```

## オーバーレイ (L3VPN) の確認

### インポジションパス

- BGP ネイバー接続がそれぞれのネイバー ノードで確立されているかどうかを確認します。

```

Router-PE1#show bgp summary
BGP router identifier 13.13.13.1, local AS number 2001
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000000 RD version: 18003
BGP main routing table version 18003
BGP NSR Initial initsync version 3 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

BGP is operating in STANDALONE mode.

Process          RcvTblVer   bRIB/RIB    LabelVer    ImportVer    SendTblVer   StandbyVer
Speaker          18003       18003       18003       18003       18003        0

Neighbor        Spk   AS  MsgRcvd  MsgSent   TblVer  InQ  OutQ  Up/Down   St/PfxRcd
21.21.21.1      0    2001   19173   7671     18003    0    0    1d07h    4000
192.13.2.149    0    7001   4615    7773     18003    0    0    09:26:21  125

```

- BGP ルートがアドバタイズされ、学習されたかどうかを確認します。

```

Router-PE1#show bgp vpnv4 unicast
BGP router identifier 13.13.13.1, local AS number 2001
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0 RD version: 0
BGP main routing table version 305345

```

```

BGP NSR Initial initsync version 12201 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
               i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network      Next Hop      Metric LocPrf Weight Path
Route Distinguisher: 2001:1601 (default for vrf vrf1601)
*> 20.13.1.1/32      192.13.26.5              0 7501 i
*> 20.13.1.2/32      192.13.26.5              0 7501 i
*>i20.23.1.1/32      20.20.20.1              100 0 6553700 11501 i
*>i20.23.1.2/32      20.20.20.1              100 0 6553700 11501 i

```

- BGP ラベルを確認します。

```

Router-PE1#show bgp label table
Label    Type          VRF/RD      Context
24041    IPv4 VRF Table vrf1601     -
24042    IPv4 VRF Table vrf1602     -

```

- ルートがそれぞれの VRF にダウンロードされているかどうかを確認します。

```

Router-PE1#show cef vrf vrf1601 20.23.1.1
20.23.1.1/32, version 743, internal 0x5000001 0x0 (ptr 0x8f932174) [1], 0x0
(0x8fa99990), 0xa08 (0x8f9fba58)
Updated Apr 20 12:33:47.840
Prefix Len 32, traffic index 0, precedence n/a, priority 3
  via 20.20.20.1/32, 3 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0x8c0e3148 0x0]
    recursion-via-/32
    next hop VRF - 'default', table - 0xe0000000
    next hop 20.20.20.1/32 via 24039/0/21
    next hop 191.23.1.2/32 Hu0/0/1/1 Labels imposed {24059 24031}

```

## 判定結果パス

- インポジション ラベルと判定結果ラベルが割り当てられており、ラベル バインディング が L3VPN に対して交換されているかどうかを確認します。

```

Router-PE2#show mpls lsd forwarding
In_Label, (ID), Path_Info: <Type>
24030, (IPv4, 'default':4U, 13.13.13.1/32), 5 Paths
  1/1: IPv4, 'default':4U, Hu0/9/0/0, nh=191.31.1.93, lbl=24155,
        flags=0x0, ext_flags=0x0
24031, (VPN-VRF, 'vrf1601':4U), 1 Paths
  1/1: PopLkup-v4, 'vrf1601':4U, ipv4
24032, (VPN-VRF, 'vrf1602':4U), 1 Paths
  1/1: PopLkup-v4, 'vrf1602':4U, ipv4

```

- FIB がラベルの更新を受け取っているかどうかを確認します。

```

Router-PE2#show mpls forwarding
Local  Outgoing  Prefix          Outgoing  Next Hop      Bytes

```

Label	Label	or ID	Interface	Switched
24019	Pop	18.18.18.3/32	Hu0/0/1/0Hu0/9/0/0	191.31.1.89 11151725032
<b>24030</b>	24155	<b>13.13.13.1/32</b>	Hu0/9/0/0	191.31.1.89 3639895
<b>24031</b>	Aggregate	<b>vrf1601</b> : Per-VRF Aggr[V] \	vrf1601	32167647049

## ASBR で IPv4 ルートおよび MPLS ラベルを交換する MPLS VPN Inter-AS を使用する複数の自律システム間への VPN 接続性の提供



(注) このセクションは、Inter-AS over IP トンネルの場合には適用されません。

ここでは、次のタスクの手順を示します。

### 概念

.

### IPv4 ルートおよび MPLS ラベルを交換する ASBR の設定

次に、IPv4 ルートおよび MPLS ラベルを交換する自律システム境界ルータ（ASBR）を設定する例を示します。

#### 設定例

```
Router# configure
Router(config)#router bgp 500
Router(config-bgp)#address-family ipv4 unicast
Router(config-bgp-af)#allocate-label all
Router(config-bgp-af)#neighbor 16.1.1.1
Router(config-bgp-nbr)#remote-as 100
Router(config-bgp-nbr)#address-family ipv4 labeled-unicast
Router(config-bgp-nbr-af)#route-policy pass-all in
Router(config-bgp-nbr-af)#route-policy pass-all out
Router(config-bgp-nbr-af)#commit
```

#### 実行コンフィギュレーション

```
router bgp 500
bgp router-id 60.200.11.1
address-family ipv4 unicast
  allocate-label all
!
neighbor 16.1.1.1
  remote-as 100
address-family ipv4 labeled-unicast
  route-policy PASS-ALL in
  route-policy pass-all out
```



!

!

## 確認

Router#**show bgp ipv4 labeled-unicast**

```
BGP router identifier 60.200.11.1, local AS number 500
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000000 RD version: 10
BGP main routing table version 10
BGP NSR Initial initsync version 6 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
                i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network        Next Hop           Metric LocPrf Weight Path
*> 10.200.1.1/32    16.1.1.1              0             0 100 ?
*                  66.161.1.1            0             0 100 ?
*> 10.200.2.1/32    16.1.1.1              5             0 100 ?
*                  66.161.1.1            5             0 100 ?
*> 10.200.5.1/32    16.1.1.1             11             0 100 ?
*                  66.161.1.1            11             0 100 ?
*> 10.200.6.1/32    16.1.1.1              4             0 100 ?
*                  66.161.1.1            4             0 100 ?
*> 60.200.11.1/32   0.0.0.0               0            32768 ?
*>i60.200.12.1/32   60.200.12.1           0            100 0 ?
*>i60.200.13.1/32   60.200.13.1           0            100 0 ?
```

Router#**show bgp ipv4 labeled-unicast 10.200.1.1**

```
BGP routing table entry for 10.200.1.1/32
Versions:
  Process          bRIB/RIB   SendTblVer
  Speaker          31         31
    Local Label: 64006

Paths: (2 available, best #1)
  Advertised to peers (in unique update groups):
    60.200.12.1
  Path #1: Received by speaker 0
  Advertised to peers (in unique update groups):
    60.200.12.1
100
  16.1.1.1 from 16.1.1.1 (10.200.1.1)
    Received Label 3
    Origin incomplete, metric 0, localpref 100, valid, external, best, group-best,
    multipath, labeled-unicast
    Received Path ID 0, Local Path ID 0, version 31
    Origin-AS validity: not-found
```

Router#**show cef vrf default ipv4 10.200.1.1**

```
10.200.1.1/32, version 161, internal 0x5000001 0x0 (ptr 0x8910c440) [1], 0x0 (0x87f73bc0),
0xa00 (0x88f40118)
Updated May  3 18:10:47.034
Prefix Len 32, traffic index 0, precedence n/a, priority 4
Extensions: context-label:64006
  via 16.1.1.1/32, 3 dependencies, recursive, bgp-ext, bgp-multipath [flags 0x60a0]
    path-idx 0 NHID 0x0 [0x889e55a0 0x87b494b0]
    recursion-via-/32
```

```

next hop 16.1.1.1/32 via 16.1.1.1/32
  local label 64006
next hop 16.1.1.1/32 Te0/0/1/4/2 labels imposed {ImplNull ImplNull}
via 66.161.1.1/32, 3 dependencies, recursive, bgp-ext, bgp-multipath [flags 0x60a0]
path-idx 1 NHID 0x0 [0x89113870 0x87b493e8]
recursion-via-/32
next hop 66.161.1.1/32 via 66.161.1.1/32
  local label 64006
next hop 66.161.1.1/32 BE161 labels imposed {ImplNull ImplNull}
Router#

```

### 関連コマンド

- allocate-label all
- address-family ipv4 labeled-unicast

## VPN-IPv4 ルートを交換するようにルート リフレクタを設定する

次に、マルチホップを使用して VPN-IPv4 ルートを交換するようにルート リフレクタを設定する例を示します。ここでは、自律システム（AS）間でネクストホップ情報および VPN ラベルが維持されるように指定します。

### 設定例

```

Router# configure
Router(config)# router bgp 500
Router(config-bgp)# neighbor 10.200.2.1
Router(config-bgp-nbr)# remote-as 100
Router(config-bgp-nbr)# ebgp-multihop
Router(config-bgp-nbr)# update-source loopback0
Router(config-bgp-nbr)# address-family vpnv4 unicast
Router(config-bgp-nbr-af)# route-policy pass-all in
Router(config-bgp-nbr-af)# route-policy pass-all out
Router(config-bgp-nbr-af)# next-hop-unchanged
Router(config-bgp-nbr)# address-family vpnv6 unicast
Router(config-bgp-nbr-af)# route-policy pass-all in
Router(config-bgp-nbr-af)# route-policy pass-all out
Router(config-bgp-nbr-af)# next-hop-unchanged

```

### 実行コンフィギュレーション

```

Router#show run router bgp 500
router bgp 500
  bgp router-id 60.200.13.1
  address-family ipv4 labeled-unicast
    allocate-label all
  !
  address-family vpnv4 unicast
  !
  address-family ipv6 unicast
  !
  address-family vpnv6 unicast
  !
  neighbor 10.200.1.1
    remote-as 100
    ebgp-multihop 255
    update-source Loopback0
    address-family vpnv4 unicast
      route-policy PASS-ALL in

```

```

route-policy PASS-ALL out
next-hop-unchanged
!
address-family vpnv6 unicast
route-policy PASS-ALL in
route-policy PASS-ALL out
next-hop-unchanged
!

```

## 確認

```

Router#show cef vrf vrf2001 ipv4 111.1.1.2/32 hardware egress location0/RP0/CPU0
111.1.1.2/32, version 39765, internal 0x5000001 0x0 (ptr 0x9f4d326c) [1], 0x0 (0xa0263058),
0x808 (0x899285b8)
Updated Oct 27 10:58:39.350
Prefix Len 32, traffic index 0, precedence n/a, priority 3
via 10.200.1.1/32, 307 dependencies, recursive, bgp-ext [flags 0x6020]
path-idx 0 NHID 0x0 [0x89a59100 0x0]
recursion-via-/32
next hop VRF - 'default', table - 0xe0000000
next hop 10.200.1.1/32 via 69263/0/21
next hop 63.13.1.1/32 Te0/3/0/17/0 labels imposed {24007 64007 64023}

LEAF - HAL pd context :
sub-type : IPV4, ecd_marked:0, has_collapsed_ldi:0
collapse_bwalk_required:0, ecdv2_marked:0
HW Walk:
LEAF:
PI:0x9f4d326c PD:0x9f4d3304 Rev:3865741 type: 0
FEC handle: 0x890c0198

LWLDI:
PI:0xa0263058 PD:0xa0263098 rev:3865740 p-rev: ldi type:0
FEC hdl: 0x890c0198 fec index: 0x0(0) num paths:1, bkup: 0

REC-SHLDI HAL PD context :
ecd_marked:0, collapse_bwalk_required:0, load_shared_lb:0

RSHLDI:
PI:0x9f17bfd8 PD:0x9f17c054 rev:0 p-rev:0 flag:0x1
FEC hdl: 0x890c0198 fec index: 0x20004fa6(20390) num paths: 1
Path:0 fec index: 0x20004fa6(20390) DSP fec index: 0x2000120e(4622)
MPLS Encap Id: 0x4001381e

LEAF - HAL pd context :
sub-type : MPLS, ecd_marked:0, has_collapsed_ldi:0
collapse_bwalk_required:0, ecdv2_marked:0
HW Walk:
LEAF:
PI:0x89a59100 PD:0x89a59198 Rev:3864195 type: 2
FEC handle: (nil)

LWLDI:
EOS0/1 LDI:
PI:0xb9a51838 PD:0xb9a51878 rev:3864192 p-rev: ldi type:0
FEC hdl: 0x890c0818 fec index: 0x20004fa2(20386) num paths:1, bkup: 0
DSP fec index:0x2000120e(4622)
Path:0 fec index: 0x20004fa2(20386) DSP fec index:0x2000120e(4622)
MPLS encap hdl: 0x400145ed MPLS encap id: 0x400145ed Remote: 0
IMP LDI:
PI:0xb9a51838 PD:0xb9a51878 rev:3864192 p-rev:
FEC hdl: 0x890c0b58 fec index: 0x20004fa0(20384) num paths:1
Path:0 fec index: 0x20004fa0(20384) DSP fec index: 0x2000120e(4622)

```

```

MPLS encap hdl: 0x400145ec MPLS encap id: 0x400145ec Remote: 0

REC-SHLDI HAL PD context :
ecd_marked:0, collapse_bwalk_required:0, load_shared_lb:0

RSHLDI:
PI:0xb7e387f8 PD:0xb7e38874 rev:0 p-rev:0 flag:0x1
FEC hdl: 0x890c0e98 fec index: 0x20004f9e(20382) num paths: 1
Path:0 fec index: 0x20004f9e(20382) DSP fec index: 0x2000120e(4622)

LEAF - HAL pd context :
sub-type : MPLS, ecd_marked:0, has_collapsed_ldi:0
collapse_bwalk_required:0, ecdv2_marked:0
HW Walk:
LEAF:
PI:0x89a59028 PD:0x89a590c0 Rev:31654 type: 2
FEC handle: (nil)

LWLDI:
PI:0x8c69c1c8 PD:0x8c69c208 rev:31653 p-rev:31652 ldi type:5
FEC hdl: 0x8903a718 fec index: 0x0(0) num paths:1, bkup: 0
Path:0 fec index: 0x0(0) DSP:0x0
IMP LDI:
PI:0x8c69c1c8 PD:0x8c69c208 rev:31653 p-rev:31652
FEC hdl: 0x8903aa58 fec index: 0x2000120e(4622) num paths:1
Path:0 fec index: 0x2000120e(4622) DSP:0x518
MPLS encap hdl: 0x40013808 MPLS encap id: 0x40013808 Remote: 0

SHLDI:
PI:0x8af02580 PD:0x8af02600 rev:31652 dpa-rev:66291 flag:0x0
FEC hdl: 0x8903a718 fec index: 0x2000120d(4621) num paths: 1 bkup paths: 0
p-rev:2373
Path:0 fec index: 0x2000120d(4621) DSP:0x518 Dest fec index: 0x0(0)

TX-NHINFO:
PD: 0x89bf94f0 rev: 2373 dpa-rev: 9794 Encap hdl: 0x8a897628
Encap id: 0x40010002 Remote: 0 L3 int: 1043 npu_mask: 4

```

## 関連コマンド

- address-family vpnv4 unicast
- allocate-label all
- ebgp-multihop
- next-hop-unchanged

## AS 内でリモート ルートを反映するルート リフレクタの設定

次に、ルートリフレクタ（RR）が、自律システム境界ルータから学習した IPv4 ルートおよびラベルを、自律システム内のプロバイダー エッジ（PE）ルータに反映できるようにする例を示します。これは、ASBR および PE を RR のルート リフレクタ クライアントとしてマークすることで実現されます。

### 設定例

```

Router#configure
Router(config)#router bgp 500
Router(config-bgp)#address-family ipv4 unicast

```

```

Router(config-bgp-af)#allocate-label all
Router(config-bgp-af)#neighbor 60.200.11.1
Router(config-bgp-nbr)#remote-as 500
Router(config-bgp-nbr)#update-source loopback0
Router(config-bgp-nbr)#address-family ipv4 labeled-unicast
Router(config-bgp-nbr-af)#route-reflector-client
Router(config-bgp-nbr-af)#neighbor 60.200.12.1
Router(config-bgp-nbr)#remote-as 500
Router(config-bgp-nbr)#update-source loopback0
Router(config-bgp-nbr)#address-family ipv4 labeled-unicast
Router(config-bgp-nbr-af)#route-reflector-client
Router(config-bgp-nbr)#address-family vpnv4 unicast
Router(config-bgp-nbr-af)#route-reflector-client

```

### 実行コンフィギュレーション

```

Router#show run router bgp 500
router bgp 500
  bgp router-id 60.200.13.1
  address-family ipv4 unicast
    allocate-label all
  !
  address-family vpnv4 unicast
  !
  neighbor 60.200.11.1
    remote-as 500
    update-source Loopback0
  !
  address-family ipv6 labeled-unicast
    route-reflector-client
  !
  address-family vpnv6 unicast
  !
  !
  neighbor 60.200.12.1
    remote-as 500
    update-source Loopback0
  address-family ipv4 labeled-unicast
    route-reflector-client
  !
  address-family vpnv4 unicast
    route-reflector-client
  !

```

## ASBR で VPN-IPv4 アドレスを交換する MPLS VPN Inter-AS を使用する複数の自律システム間への VPN 接続性の提供

ここでは、次のタスクの手順を示します。

### IP トンネルで VPN-IPv4 アドレスを交換する ASBR の設定

外部ボーダー ゲートウェイ プロトコル (eBGP) 自律システム境界ルータ (ASBR) を、他の自律システムと VPN-IPv4 ルートを交換するように設定するには、次の作業を実行します。

## 手順

**ステップ 1** **configure**

例 :

```
RP/0/RP0/cpu 0: router# configure
```

XR コンフィギュレーション モードを開始します。

**ステップ 2** **router bgp *autonomous-system-number***

例 :

```
RP/0/RP0/cpu 0: router(config)# router bgp 120
RP/0/RP0/cpu 0: router(config-bgp)#
```

ボーダー ゲートウェイ プロトコル (BGP) 設定モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。

**ステップ 3** **address-family { *ipv4 tunnel* }**

例 :

```
RP/0/RP0/cpu 0: router(config-bgp)# address-family ipv4 tunnel
RP/0/RP0/cpu 0: router(config-bgp-af)#
```

IPv4 トンネル アドレス ファミリを設定します。

**ステップ 4** **address-family { *vpn4 unicast* }**

例 :

```
RP/0/RP0/cpu 0: router(config-bgp-af)# address-family vpnv4 unicast
```

VPNv4 アドレス ファミリを設定します。

**ステップ 5** **neighbor *ip-address***

例 :

```
RP/0/RP0/cpu 0: router(config-bgp-af)# neighbor 172.168.40.24
RP/0/RP0/cpu 0: router(config-bgp-nbr)#
```

ルータを BGP ルーティングのネイバー設定モードにして、ネイバーの IP アドレス 172.168.40.24 を ASBR eBGP ピアとして設定します。

**ステップ 6** **remote-as *autonomous-system-number***

例 :

```
RP/0/RP0/cpu 0: router(config-bgp-nbr)# remote-as 2002
```

ネイバーを作成し、そのネイバーをリモート自律システム番号に割り当てます。

**ステップ 7** **address-family { *vpn4 unicast* }**

例 :

```
RP/0/RP0/cpu 0: router(config-bgp-nbr)# address-family vpnv4 unicast
RP/0/RP0/cpu 0: router(config-bgp-nbr-af)#
```

VPNv4 アドレス ファミリを設定します。

#### ステップ 8 **route-policy route-policy-name { in }**

例 :

```
RP/0/RP0/cpu 0: router(config-bgp-nbr-af)# route-policy pass-all in
```

BGP ネイバーから受信されるアップデートに、ルーティング ポリシーを適用します。

- ルート ポリシーの名前を定義するために *route-policy-name* 引数を使用します。例では、ルート ポリシーの名前が **pass-all** として定義されていることを示します。
- インバウンド ルートのポリシーを定義するために **in** キーワードを使用します。

#### ステップ 9 **route-policy route-policy-name { out }**

例 :

```
RP/0/RP0/cpu 0: router(config-bgp-nbr-af)# route-policy pass-all out
```

BGP ネイバーから送信されたアップデートに、ルーティング ポリシーを適用します。

- ルート ポリシーの名前を定義するために *route-policy-name* 引数を使用します。例では、ルート ポリシーの名前が **pass-all** として定義されていることを示します。
- アウトバウンド ルートのポリシーを定義するために **out** キーワードを使用します。

#### ステップ 10 **neighbor ip-address**

例 :

```
RP/0/RP0/cpu 0: router(config-bgp-nbr-af)# neighbor 175.40.25.2
RP/0/RP0/cpu 0: router(config-bgp-nbr)#
```

ルータを BGP ルーティングのネイバー設定モードにして、ネイバーの IP アドレス 175.40.25.2 を VPNv4 iBGP ピアとして設定します。

#### ステップ 11 **remote-as autonomous-system-number**

例 :

```
RP/0/RP0/cpu 0: router(config-bgp-nbr)# remote-as 2002
```

ネイバーを作成し、そのネイバーをリモート自律システム番号に割り当てます。

#### ステップ 12 **update-source type interface-path-id**

例 :

```
RP/0/RP0/cpu 0: router(config-bgp-nbr)# update-source loopback0
```

BGP セッションが、特定のインターフェイスのプライマリ IP アドレスをローカル アドレスとして使用できるようにします。

### ステップ 13 **address-family { ipv4 tunnel }**

例：

```
RP/0/RP0/cpu 0: router(config-bgp-nbr)# address-family ipv4 tunnel
RP/0/RP0/cpu 0: router(config-bgp-nbr-af)#
```

IPv4 トンネル アドレス ファミリを設定します。

### ステップ 14 **address-family { vpnv4 unicast }**

例：

```
RP/0/RP0/cpu 0: router(config-bgp-nbr-af)# address-family vpnv4 unicast
```

VPNv4 アドレス ファミリを設定します。

### ステップ 15 **commit** コマンドまたは **end** コマンドを使用します。

**commit**：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end**：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## ASBR ピアへのスタティック ルートの設定

ASBR ピアにスタティック ルートを設定するには、次の作業を実行します。

手順

### ステップ 1 **configure**

例：

```
RP/0/RP0/cpu 0: router# configure
```

XR 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 static**

例：

```
RP/0/RP0/cpu 0: router(config)# router static
RP/0/RP0/cpu 0: router(config-static)#
```

ルータ スタティック コンフィギュレーション モードを開始します。



**ステップ 3 address-family ipv4 unicast**

例 :

```
RP/0/RP0/cpu 0: router(config-static)# address-family ipv4 unicast
RP/0/RP0/cpu 0: router(config-static-afi)#
```

IPv4 アドレス ファミリをイネーブルにします。

**ステップ 4 A.B.C.D/length next-hop**

例 :

```
RP/0/RP0/cpu 0: router(config-static-afi)# 10.10.10.10/32 10.9.9.9
```

宛先ルータのアドレスを入力します (IPv4 サブネット マスクを含む)。

**ステップ 5 commit コマンドまたは end コマンドを使用します。****commit** : 設定の変更を保存し、コンフィギュレーションセッションに留まります。**end** : 次のいずれかのアクションを実行することをユーザに要求します。

- [Yes] : 設定変更を保存し、コンフィギュレーションセッションを終了します。
- [No] : 設定変更をコミットせずにコンフィギュレーションセッションを終了します。
- [Cancel] : 設定変更をコミットせずに、コンフィギュレーションモードに留まります。

**連合内のサブ自律システム間で VPN ルートを交換する EBGp ルーティングの設定**

連合内のサブ自律システム間で VPN ルートを交換する外部ボーダー ゲートウェイ プロトコル (eBGp) ルーティングを設定するには、次の作業を実行します。



- (注) VPN-IPv4 eBGp ネイバーのホスト ルートが他のルータおよび PE ルータに (Interior Gateway Protocol (IGP) を使用して) 伝達されるようにするには、連合 eBGp (CEBGp) ルータの IGP 設定部分で **redistribute connected** コマンドを指定します。Open Shortest Path First (OSPF) を使用する場合、「redistribute connected」サブネットが存在する CEBGP インターフェイスで OSPF プロセスが有効にされていないことを確認します。

手順

**ステップ 1 configure**

例 :

```
RP/0/RP0/cpu 0: router# configure
```

XR コンフィギュレーション モード を開始します。

## ステップ 2 **router bgp autonomous-system-number**

例 :

```
RP/0/RP0/cpu 0: router(config)# router bgp 120
RP/0/RP0/cpu 0: router(config-bgp)#
```

BGP コンフィギュレーション モードを開始します。このモードでは BGP ルーティング プロセスを設定できます。

## ステップ 3 **bgp confederation peers peer autonomous-system-number**

例 :

```
RP/0/RP0/cpu 0: router(config-bgp)# bgp confederation peers 8
```

連合に属するピアの自律システム番号を設定します。

## ステップ 4 **bgp confederation identifier autonomous-system-number**

例 :

```
RP/0/RP0/cpu 0: router(config-bgp)# bgp confederation identifier 5
```

連合 ID の自律システム番号を指定します。

## ステップ 5 **address-family vpnv4 unicast**

例 :

```
RP/0/RP0/cpu 0: router(config-bgp)# address-family vpnv4 unicast
RP/0/RP0/cpu 0: router(config-bgp-af)#
```

VPNv4 アドレス ファミリを設定します。

## ステップ 6 **neighbor ip-address**

例 :

```
RP/0/RP0/cpu 0: router(config-bgp-af)# neighbor 10.168.40.24
RP/0/RP0/cpu 0: router(config-bgp-nbr)#
```

ルータを BGP ルーティングのネイバー設定モードにして、ネイバーの IP アドレス 10.168.40.24 を BGP ピアとして設定します。

## ステップ 7 **remote-as autonomous-system-number**

例 :

```
RP/0/RP0/cpu 0: router(config-bgp-nbr)# remote-as 2002
```

ネイバーを作成し、そのネイバーをリモート自律システム番号に割り当てます。

#### ステップ 8 **address-family vpnv4 unicast**

例 :

```
RP/0/RP0/cpu 0: router(config-bgp-nbr)# address-family vpnv4 unicast
RP/0/RP0/cpu 0: router(config-bgp-nbr-af)#
```

VPNv4 アドレス ファミリを設定します。

#### ステップ 9 **route-policy route-policy-name in**

例 :

```
RP/0/RP0/cpu 0: router(config-bgp-nbr-af)# route-policy In-Ipv4 in
```

BGP ネイバーから受信されるアップデートに、ルーティング ポリシーを適用します。

#### ステップ 10 **route-policy route-policy-name out**

例 :

```
RP/0/RP0/cpu 0: router(config-bgp-nbr-af)# route-policy Out-Ipv4 out
```

BGP ネイバーにアドバタイズされるアップデートに、ルーティング ポリシーを適用します。

#### ステップ 11 **next-hop-self**

例 :

```
RP/0/RP0/cpu 0: router(config-bgp-nbr-af)# next-hop-self
```

ネクストホップ計算をディセーブルにし、BGP アップデートのネクストホップ フィールドにユーザ自身のアドレスの挿入を許可します。

#### ステップ 12 **commit** コマンドまたは **end** コマンドを使用します。

**commit** : 設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** : 次のいずれかのアクションを実行することをユーザに要求します。

- [Yes] : 設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No] : 設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel] : 設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## ASBR 連合での MPLS 転送の設定

指定したインターフェイスで自律システム境界ルータ（ASBR）連合（BGP 内）への MPLS 転送を設定するには、次の作業を実行します。



- (注) この設定は、インターフェイスに関連付けられたピアに対応する暗黙のヌル書き換えを追加します。これは、（マルチホップ インスタンスで）BGP が LDP によって自動的にリライトをインストールしないようにするために必要です。

### 手順

#### ステップ 1 **configure**

例：

```
RP/0/RP0/cpu 0: router# configure
```

XR 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：

```
RP/0/RP0/cpu 0: router(config)# router bgp 120
RP/0/RP0/cpu 0: router(config-bgp)
```

BGP コンフィギュレーション モードを開始します。このモードでは BGP ルーティング プロセスを設定できます。

#### ステップ 3 **mpls activate**

例：

```
RP/0/RP0/cpu 0: router(config-bgp)# mpls activate
RP/0/RP0/cpu 0: router(config-bgp-mpls)#
```

BGP MPLS アクティブ化設定モードを開始します。

#### ステップ 4 **interface type interface-path-id**

例：

```
RP/0/RP0/cpu 0: router(config-bgp-mpls)# interface hundredGigE 0/9/0/0
```

インターフェイスで MPLS をイネーブルにします。

#### ステップ 5 **commit** コマンドまたは **end** コマンドを使用します。

**commit** : 設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** : 次のいずれかのアクションを実行することをユーザに要求します。

- [Yes] : 設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No] : 設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel] : 設定変更をコミットせずに、コンフィギュレーション モードに留まります。

---

## ASBR 連合ピアへのスタティック ルートの設定

Inter-AS 連合ピアにスタティック ルートを設定するには、次の作業を実行します。

### 手順

---

#### ステップ 1 **configure**

例 :

```
RP/0/RP0/cpu 0: router# configure
```

XR 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 static**

例 :

```
RP/0/RP0/cpu 0: router(config)# router static
RP/0/RP0/cpu 0: router(config-static)#
```

ルータ スタティック コンフィギュレーション モードを開始します。

#### ステップ 3 **address-family ipv4 unicast**

例 :

```
RP/0/RP0/cpu 0: router(config-static)# address-family ipv4 unicast
RP/0/RP0/cpu 0: router(config-static-afi)#
```

IPv4 アドレス ファミリをイネーブルにします。

#### ステップ 4 **A.B.C.D/length next-hop**

例 :

```
RP/0/RP0/cpu 0: router(config-static-afi)# 10.10.10.10/32 10.9.9.9
```

宛先ルータのアドレスを入力します (IPv4 サブネット マスクを含む)。

#### ステップ 5 **commit** コマンドまたは **end** コマンドを使用します。

**commit** : 設定の変更を保存し、コンフィギュレーションセッションに留まります。

**end** : 次のいずれかのアクションを実行することをユーザに要求します。

- [Yes] : 設定変更を保存し、コンフィギュレーションセッションを終了します。
- [No] : 設定変更をコミットせずにコンフィギュレーションセッションを終了します。
- [Cancel] : 設定変更をコミットせずに、コンフィギュレーションモードに留まります。

## VRF-lite

VRF-lite は、MPLS のない VRF 導入環境です。サービスプロバイダーは、VRF-lite により重複する IP アドレスで複数の VPN をサポートできます。この機能を使用することで、複数の VRF インスタンスをカスタマー エッジデバイス内でサポートできます。

VRF-lite インターフェイスはレイヤ 3 インターフェイスとする必要があります、また、このインターフェイスは、どのような場合も複数の VRF に属することができません。複数のインターフェイスのすべてが同じ VPN に参加している場合は、それら複数のインターフェイスを同じ VRF に含めることができます。

## VRF-lite の設定

2 つの VPN サイトがそれぞれにあり、それらのサイトが同じ PE ルータに接続している 2 つのカスタマーを検討します。各カスタマーに個別のルーティングテーブルを作成するため、VRF を使用します。各カスタマー（たとえば、vrf1 と vrf2）に 1 つの VRF を作成してから、それぞれの VRF に対応するルータのインターフェイスを追加します。各 VRF には、独自のルーティングテーブルと、それに対応して設定されたインターフェイスがあります。ルータのグローバルルーティングテーブルにはこれらのインターフェイスは表示されません。それに対し、VRF ルーティングテーブルには、VRF に追加したインターフェイスが表示されます。スタティックルーティング、または BGP や RIP などのルーティングプロトコルを使用して CE デバイスとルーティング情報を交換する PE ルータ。

つまり、VRF-lite 設定には、次の主要タスクが含まれます。

- VRF の作成
- インターフェイス下での VRF の設定
- ルーティングプロトコルでの VRF の設定

### 設定例

- VRF の作成 :

```
Router#configure
Router(config)#vrf vrf1
Router(config-vrf)#address-family ipv4 unicast
```

```

/* You must create route-policy pass-all before this configuration */
Router(config-vrf-af) #import from default-vrf route-policy pass-all
Router(config-vrf-af) #import route-target
Router(config-vrf-import-rt) #100:100
Router(config-vrf-import-rt) #exit
Router(config-vrf-af) #export route-target
Router(config-vrf-import-rt) #100:100
Router(config-vrf-import-rt) #exit
Router(config-vrf-import-rt) #commit

```

同様に、ルートとターゲットを 100:100 として vrf2 を作成します。

#### ・インターフェイス下での VRF の設定

```

Router#configure
Router(config) #interface HundredGigE0/9/0/0.2001
Router(config-subif) #vrf vrf1
Router(config-subif) #ipv4 address 192.0.2.2 255.255.255.252
Router(config-subif) #encapsulation dot1q 2001
Router(config-subif) #exit

Router(config) #interface HundredGigE0/9/0/0.2000
Router(config-subif) #vrf vrf2
Router(config-subif) #ipv4 address 192.0.2.5/30 255.255.255.252
Router(config-subif) #encapsulation dot1q 2000
Router(config-vrf-import-rt) #commit

```

同様に、vrf1 をインターフェイス HundredGigE 0/9/0/0.2001 に、vrf2 をインターフェイス HundredGigE 0/9/0/0.2000 に設定します。

#### ・ルーティング プロトコルでの VRF の設定 :

```

Router#configure
Router(config) #router rip
Router(config-rip) #vrf vrf1
Router(config-rip-vrf) #interface HundredGigE0/9/0/0.2001
Router(config-rip-vrf-if) #exit
Router(config-rip-vrf) #interface HundredGigE0/9/0/0.2001
Router(config-rip-vrf-if) #exit
Router(config-rip-vrf) #default-information originate
Router(config-vrf-import-rt) #commit

```

同様に vrf2 を RIP で HundredGigE 0/9/0/0.2000 に、vrf2 をインターフェイス HundredGigE 0/9/0/1.2000 に設定します。

### 実行コンフィギュレーション

```

/* VRF Configuration */

vrf vrf1
address-family ipv4 unicast
import route-target
100:100
!
export route-target

```

```

        100:100
    !
    !
    !
vrf vrf2
address-family ipv4 unicast
    import route-target
        100:100
    !
    export route-target
        100:100
    !
!
!

/* Interface Configuration */

interface HundredGigE 0/9/0/0.2001
vrf vrf1
ipv4 address 192.0.2.2 255.255.255.252
encapsulation dot1q 2001
!

interface HundredGigE 0/9/0/0.2000
vrf vrf2
ipv4 address 192.0.2.5/30 255.255.255.252
encapsulation dot1q 2000
!

interface HundredGigE 0/9/0/1.2001
vrf vrf1
ipv4 address 203.0.113.2 255.255.255.252
encapsulation dot1q 2001
!

interface HundredGigE 0/9/0/1.2000
vrf vrf2
ipv4 address 203.0.113.5 255.255.255.252
encapsulation dot1q 2000
!

/* Routing Protocol Configuration */
router rip
interface Loopback0
!
interface HundredGigE0/9/0/0
!
interface HundredGigE0/9/0/0.2000
!
interface HundredGigE0/9/0/0.2001
!
interface HundredGigE0/9/0/1
!
interface HundredGigE0/9/0/1.2000
!
interface HundredGigE0/9/0/1.2001
!

vrf vrf1
    interface HundredGigE0/9/0/0.2001
    !
    interface HundredGigE0/9/0/1.2001
    !

```



```

    default-information originate
    !
vrf vrf2
    interface HundredGigE0/9/0/0.2000
    !
    interface HundredGigE0/9/0/1.2000
    !
    default-information originate
    !

```

## 確認

```

Router#show route vrf vrf1
Mon Jul  4 19:12:54.739 UTC

```

```

Codes: C - connected, S - static, R - RIP, B - BGP, (>) - Diversion path
O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
U - per-user static route, o - ODR, L - local, G - DAGR, l - LISP
A - access/subscriber, a - Application route
M - mobile route, r - RPL, (!) - FRR Backup path

```

Gateway of last resort is not set

```

C    203.0.113.0/24 is directly connected, 00:07:01, HundredGigE0/9/0/1.2001
L    203.0.113.2/30 is directly connected, 00:07:01, HundredGigE0/9/0/1.2001
C    192.0.2.0/24 is directly connected, 00:05:51, HundredGigE0/9/0/1.2001
L    192.0.2.2/30 is directly connected, 00:05:51, HundredGigE0/9/0/1.2001

```

```

Router#show route vrf vrf2
Mon Jul  4 19:12:59.121 UTC

```

```

Codes: C - connected, S - static, R - RIP, B - BGP, (>) - Diversion path
O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
U - per-user static route, o - ODR, L - local, G - DAGR, l - LISP
A - access/subscriber, a - Application route
M - mobile route, r - RPL, (!) - FRR Backup path

```

Gateway of last resort is not set

```

R    198.51.100.53/30 [120/1] via 192.0.2.1, 00:01:42, HundredGigE0/9/0/0.2000
C    203.0.113.0/24 is directly connected, 00:08:43, HundredGigE0/9/0/1.2000
L    203.0.113.5/30 is directly connected, 00:08:43, HundredGigE0/9/0/1.2000
C    192.0.2.0/24 is directly connected, 00:06:17, HundredGigE0/9/0/0.2000
L    192.0.2.5/30 is directly connected, 00:06:17, HundredGigE0/9/0/0.2000

```

## 関連項目

- [VRF-lite \(46 ページ\)](#)

# セグメントルーティングを使用した MPLS L3VPN サービス

現在、MPLS L3VPN サービスの転送に MPLS Label Distribution Protocol (LDP) が広く使用されています。MPLS LDP の代わりにセグメントルーティングを使用して MPLS L3VPN サービスを転送することで、ネットワークトラフィックのレジリエンスとコンバージェンスが向上します。セグメントルーティングは、転送プレーンを変更することなく、MPLS アーキテクチャに直接適用できます。MPLS データプレーンを使用するセグメントルーティングネットワークでは、LDP またはその他のシグナリングプロトコルは不要です。その代わりに、IGP (IS-IS または OSPF) か BGP プロトコルによってラベル配布が実行されます。ネットワークからプロトコルを排除することでその操作が簡略化し、プロトコルとの対話式操作の必要をなくすることで堅牢性と安定性が向上します。セグメントルーティングは、従来の MPLS ネットワークよりも効率よくネットワーク帯域幅を利用し、遅延を低減します。

## セグメントルーティングを介した MPLS L3VPN の設定

### トポロジ

次に示すネットワークシナリオでは、MPLS L3VPN サービスがセグメントルーティングを使用して転送されます。

CE1 - HundredGigE 0/9/0/0 - HundredGigE 0/9/0/0 - PE - HundredGigE 0/9/0/1 - HundredGigE 0/9/0/1 - P Node - HundredGigE 0/9/0/0 - HundredGigE 0/9/0/0 - PE2 - HundredGigE 0/9/0/1 - hundredGigE 0/9/0/1 - CE2

このトポロジでは、CE1 と CE2 が 2 つのカスタマー ルータです。ISP には 2 つの PE ルータ、PE1 および PE2 と、P ルータがあります。RIP は、CE ルータと PE のルータ間のエッジプロトコルのサポートに使用されます。ラベル配布は IGP (IS-IS または OSPF) または BGP によって実行されます。このシナリオでは、OSPF を使用します。

カスタマーの自律システムは 65534 で、ISP の自律システム 65000 とピアリングを行います。これは、グローバル IPv4 テーブルへのルートのアドバタイズを防ぐため、VRF ピアリングである必要があります。ISP ルータ PE1 と PE2 には、カスタマー用の VRF (vrf1601 など) が含まれています。PE1 と PE2 が同じルートターゲットをエクスポートし、インポートしますが、これは必要ではありません。

このトポロジでは、接続されているネットワークをシミュレートするためにループバックインターフェイスを使用します。

### 設定

セグメントルーティングを介した MPLS L3VPN を正常に設定するには、次のタスクを実行する必要があります。

- [MPLS コアでのセグメントルーティングの設定 \(51 ページ\)](#)

- PE-CE でのプロトコル サポートの設定 ([MPLS VPN カスタマーの接続 \(17 ページ\)](#) を参照)
- PE-PE でのプロトコルサポートの設定 ([PE ルータおよびルートリフレクタでの Multiprotocol BGP の設定 \(14 ページ\)](#) を参照)

## MPLS コアでのセグメントルーティングの設定

この項では、MPLS コアでセグメントルーティングを有効にする設定手順について説明します。この設定はトポロジ内の PE1 ルータ、P ルータ、および PE2 ルータで対応する値を使用して実行する必要があります。

### 設定例

```
/* Configure Segment Routing using OSPF */

Router-PE1#configure
Router-PE1(config)# router ospf dc-sr
Router-PE1(config-ospf)#router-id 13.13.13.1
Router-PE1(config-ospf)#segment routing mpls
Router-PE1(config-ospf)#segment routing forwarding mpls
Router-PE1(config-ospf)#mpls ldp sync
Router-PE1(config-ospf)#mpls ldp auto-config
Router-PE1(config-ospf)#segment-routing mpls sr-prefer
Router-PE1(config-ospf)#segment-routing prefix-sid-map advertise-local
Router-PE1(config-ospf)#exit
Router-PE1(config-ospf)#area 1
Router-PE1(config-ospf-ar)#interface HundredGigE 0/9/0/0
Router-PE1(config-ospf-ar-if)#exit
Router-PE1(config-ospf-ar)#interface Loopback0
Router-PE1(config-ospf-ar-if)#prefix-sid index 1
Router-PE1(config-ospf-ar-if)#commit

/* * Configure segment routing global block */

Router# configure
Router(config)# segment-routing
Router(config-sr)# global-block 180000 200000
Router(config-sr)# commit
Router(config-sr)# exit

/* Configure Segment Routing using ISIS */

Router# configure
Router(config)# router isis ring
Router(config-isis)# is-type level-2-only
Router(config-isis)# net 49.0001.1921.6800.1001.00
Router(config-isis)# nsr
Router(config-isis)# distribute link-state
Router(config-isis)# nsf cisco
Router(config-isis)# address-family ipv4 unicast
Router(config-isis-af)# metric-style wide
Router(config-isis-af)# mpls traffic-eng level-1
Router(config-isis-af)# mpls traffic-eng router-id loopback0
Router(config-isis-af)# segment-routing mpls
Router(config-isis-af)# exit
```

```

!
Router(config-isis)# interface loopback0
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-af)# prefix-sid index 30101
Router(config-isis-af)# exit

```

## 実行コンフィギュレーション

### PE1 :

```

router ospf dc-sr
 router-id 13.13.13.1
 segment-routing mpls
 segment-routing forwarding mpls
 mpls ldp sync
 mpls ldp auto-config
 segment-routing mpls sr-prefer
 segment-routing prefix-sid-map receive
 segment-routing prefix-sid-map advertise-local
!
 area 1
  interface HundredGigE 0/9/0/0
  !
  interface Loopback0
   prefix-sid index 1
  !
!
!

configure
 segment-routing
  global-block 180000 200000
!
!

configure
 router isis ring
  net 49.0001.1921.6800.1001.00
  nsr
  distribute link-state
  nsf cisco
  address-family ipv4 unicast
   metric-style wide
   mpls traffic-eng level-1
   mpls traffic-eng router-id Loopback0
  segment-routing mpls
!
 interface Loopback0
  address-family ipv4 unicast
   prefix-sid index 30101
  !
!

```

### P ノード :

```

router ospf dc-sr
 router-id 16.16.16.1
 segment-routing mpls
 segment-routing forwarding mpls
 mpls ldp sync

```

```
mpls ldp auto-config
segment-routing mpls sr-prefer
segment-routing prefix-sid-map receive
segment-routing prefix-sid-map advertise-local
!
area 1
interface HundredGigE0/0/1/0
!
interface HundredGigE0/0/1/1
!
interface Loopback0
prefix-sid index 1
!
!
!

configure
segment-routing
global-block 180000 200000
!
!

configure
router isis ring
net 49.0001.1921.6800.1002.00
nsr
distribute link-state
nsf cisco
address-family ipv4 unicast
metric-style wide
mpls traffic-eng level-1
mpls traffic-eng router-id Loopback0
segment-routing mpls
!
interface Loopback0
address-family ipv4 unicast
prefix-sid index 30102
!
!
```

**PE2 :**

```
router ospf dc-sr
router-id 20.20.20.1
segment-routing mpls
segment-routing forwarding mpls
mpls ldp sync
mpls ldp auto-config
segment-routing mpls sr-prefer
segment-routing prefix-sid-map receive
segment-routing prefix-sid-map advertise-local
!
area 0
interface HundredGigE 0/12/0/0
!
interface Loopback0
prefix-sid index 1
!
!
!

configure
segment-routing
global-block 180000 200000
```

```

!
!

configure
router isis ring
net 49.0001.1921.6800.1003.00
nsr
distribute link-state
nsf cisco
address-family ipv4 unicast
metric-style wide
mpls traffic-eng level-1
mpls traffic-eng router-id Loopback0
segment-routing mpls
!
interface Loopback0
address-family ipv4 unicast
prefix-sid index 30103
!

```

### 関連項目

次のタスクを実行するとともに、セグメント ルーティングを介した MPLS L3VPN 設定を実行する必要があります。

- [MPLS VPN カスタマーの接続 \(17 ページ\)](#)
- [PE ルータおよびルート リフレクタでの Multiprotocol BGP の設定 \(14 ページ\)](#)

## セグメント ルーティングを介した MPLS L3VPN 設定の確認

- コア ルータの統計情報を確認し、IGP 転送ラベル（この例では 64003）のカウンタが増加していることを確認します。

**P ノード :**

```

Router-P#show mpls forwarding
Local  Outgoing  Prefix          Outgoing  Next Hop      Bytes
Label  Label       or ID           Interface                Switched
-----
64003  Pop         SR Pfx (idx 0)  Hu0/9/0/0  193.16.1.2    572842

```

- PE1 ルータの統計情報を確認します。

**PE1 :**

```

Router-P#show mpls forwarding
Local  Outgoing  Prefix          Outgoing  Next Hop      Bytes
Label  Label       or ID           Interface                Switched
-----
64001  60003      SR Pfx (idx 0)  Hu0/9/0/0  191.22.1.2    532978

```

- PE2 ルータの統計情報を確認し、VPN ラベル（この例では 24031）のカウンタが増加していることを確認します。

PE2 :

```
Router-PE2#show mpls forwarding
```

Local Label	Outgoing Label	Prefix or ID	Outgoing Interface	Next Hop	Bytes Switched
24031	Aggregate	vrf1601: Per-VRF Aggr[V]	\		
		vrf1601			501241

また、コマンドと出力例の詳細なリストについては、[MPLS L3VPN 設定の確認（27 ページ）](#)を参照してください。

## MPLS L3VPN の実装：リファレンス

### MPLS L3VPN の利点

MPLS L3VPN は、次の利点を提供します。

- サービス プロバイダーは、スケーラブルな VPN を展開し、付加価値サービスを提供できます。
- コネクションレス型サービスでは、ホスト間の通信を確立するために上記のアクションは必要はないことを保証します。
- 集中型サービス：レイヤ 3 に VPN を構築すると、VPN に代表されるユーザ グループに目的のサービスを配布できます。
- 拡張性：拡張性が高い VPN をコネクション型のポイントツーポイント オーバーレイを使用して作成します。
- セキュリティ：セキュリティは、（カスタマーから受信したパケットを確実に正しい VPN に配置するように）プロバイダーネットワークのエッジとバックボーンに提供されます。
- 統合 Quality of Service (QoS) サポート：QoS は、予測可能なパフォーマンスおよびポリシーの実装に対処し、MPLS VPN のさまざまなレベルのサービスをサポートする機能を提供します。
- 単純な移行：サービス プロバイダーは、単純な移行パスを使用して VPN サービスを導入できます。
- エンドカスタマーの移行が簡素化されます。CE ルータ上で MPLS をサポートする必要がなく、カスタマーのイントラネットに変更は必要ありません。

## MPLS L3VPN の主要コンポーネント：詳細

### 仮想ルーティングおよび転送テーブル

各 VPN は、1 つ以上の VPN ルーティングおよび転送（VRF）インスタンスに関連付けられています。VRF では、PE ルータに接続されているカスタマー サイトの VPN メンバーシップを定義します。VRF は、次のコンポーネントで構成されています。

- IP バージョン 4（IPv4）ユニキャスト ルーティング テーブル
- 取得された FIB テーブル
- 転送テーブルを使用する一連のインターフェイス
- ルーティング テーブルに格納されている情報を制御するための一連のルールおよびルーティング プロトコル パラメータ

これらのコンポーネントは一括して VRF インスタンスと呼ばれます。

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。VRF には、そのサイトがメンバとなっている VPN 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。

パケット転送情報は、VRF ごとに IP ルーティング テーブルおよび FIB テーブルに格納されます。各 VRF の一連のルーティング テーブルと FIB テーブルは別々に維持されます。これらのテーブルにより、VPN の外側に情報が転送されないようになっているほか、VPN の外側のパケットも VPN 内のルータに転送されないようになっています。

### VPN ルーティング情報：配信

VPN ルーティング情報の配布は、BGP 拡張コミュニティによって実装される VPN ルートターゲット コミュニティを使用して制御されます。VPN ルーティング情報は、次のように配布されます。

- CE ルータから学習した VPN ルートが BGP に注入されると、VPN ルート ターゲット拡張コミュニティ属性のリストが、そのルートに関連付けられます。通常、ルートターゲット コミュニティ拡張値のリストは、ルートの学習元の VRF に関連付けられているルートターゲットのエクスポート リストから設定されます。
- ルート ターゲット拡張コミュニティのインポート リストは、各 VRF に関連付けられています。インポート リストには、ルートが VRF にインポートされるために、ルートに設定されている必要のある、ルートターゲット拡張コミュニティ属性が定義されています。たとえば、ある特定の VRF のインポート リストにルート ターゲット拡張コミュニティ A、B、および C が含まれている場合、これらのルート ターゲット拡張コミュニティ A、B、または C のいずれかを伝送するすべての VPN ルートが VRF にインポートされます。



## VPN ルーティング情報の BGP 配布

PE ルータは、次のソースから IP プレフィックスを学習します。

- スタティック設定の CE ルータ
- CE ルータとの eBGP セッション
- Open Shortest Path First (OSPF) および Interior Gateway Protocol (IGP) としての RIP

IP プレフィックスは、IPv4 アドレス ファミリのメンバです。PE ルータは、IP プレフィックスを学習した後、その IP プレフィックスを 64 ビットのルート識別子に結合することで、VPN-IPv4 プレフィックスに変換します。生成されたプレフィックスは、VPN-IPv4 アドレス ファミリのメンバです。このプレフィックスによって、カスタマーサイトでグローバルに一意でない（登録されていないプライベート）IP アドレスが使用される場合でも、カスタマーのアドレスが一意に識別されます。VPN-IPv4 プレフィックスを生成するために使用されるルート識別子は、**rd** コマンドによって指定し、PE ルータ上で VRF に関連付けられます。

BGP は、各 VPN の VPN-IPv4 プレフィックスに関する到達可能性情報を配布します。BGP 通信は、次の 2 つのレベルで行われます。

- 内部 BGP (iBGP) : 自律システムと呼ばれる IP ドメイン内。
- 外部 BGP (eBGP) : 自律システム間。

BGP では、IPv4 以外のアドレス ファミリのサポートを定義する BGP プロトコル拡張 (RFC 2283、『Multiprotocol Extensions for BGP-4』を参照) を使用して、PE ルータ間の VPN-IPv4 プレフィックスの到達可能性情報を伝播します。この拡張を使用すると、指定された VPN のルートが、その VPN の他のメンバによってだけ学習されるようになり、VPN のメンバ間の相互通信が可能になります。

## MPLS 転送

VRF IP ルーティング テーブルおよび VRF FIB テーブルに格納されているルーティング情報に基づき、MPLS を使用してパケットが宛先に転送されます。

PE ルータでは、CE ルータから学習した各カスタマープレフィックスにラベルをバインドし、他の PE ルータにアドバタイズするプレフィックスのネットワーク到達可能性情報に、そのラベルを含めます。PE ルータは、プロバイダー ネットワークを介して CE ルータから受信したパケットを転送するときに、宛先 PE ルータから学習したラベルを使用してパケットにラベルを付けます。宛先 PE ルータは、ラベルの付いたパケットを受信すると、そのラベルをポップし、パケットを正しい CE ルータに誘導するために使用します。プロバイダーバックボーン経由のラベル転送は、ダイナミック ラベル スイッチングに基づきます。バックボーンを通過するカスタマー データ パケットが伝達するラベルには、2 つのレベルがあります。

- 最上位ラベルによって、パケットが正しい PE ルータに誘導されます。
- 2 番めのラベルは、PE ルータがパケットを CE ルータに転送する方法を示します。

## ルータ識別子の自動割り当て

iBGP ロードバランシングを利用するには、各ネットワークの VRF に一意のルータ識別子を割り当てる必要があります。異なる VPN から受信した同一である可能性があるプレフィックスを BGP が区別できるようにするため、VRF にルータ識別子が必要です。

それぞれが複数の VRF をサポートするネットワークに多くのルータが存在する場合、ネットワーク全体でルータ識別子を設定および管理すると、問題になる可能性があります。Cisco IOS XR ソフトウェアでは、**rd auto** コマンドを使用して VRF に一意のルータ識別子を割り当てることで、このプロセスが簡略化されています。

各ルータに一意のルータ識別子を割り当てるには、各ルータに一意の BGP ルータ ID を使用する必要があります。その場合は、**rd auto** コマンドを使用してタイプ 1 のルータ識別子を *ip-address:number* 形式を使用して VRF に割り当てます。IP アドレスは、BGP router-id ステートメントで指定し、その数値（0～65535 の範囲の未使用インデックスとして取得）は VRF 全体で一意です。

最後に、フェールオーバーまたはプロセス再起動後も VRF へのルータ識別子の割り当てが保持されるように、ルータ識別子の値はチェックポイントされます。ルータ識別子が VRF 用に明示的に設定されている場合、この値は自動ルータ識別子によって上書きされることはありません。