IPv6 VPN over MPLS の実装

Border Gateway Protocol（BGP; ボーダー ゲートウェイ プロトコル）- Multiprotocol Label Switching（MPLS; マルチプロトコル ラベル スイッチング）の Virtual Private Network（VPN; バーチャル プライベート ネットワーク）機能は、Provider Edge（PE; プロバイダー エッジ）ベースの VPN モデルの実装を表します。このマニュアルでは、IPv6 VPN over MPLS（6VPE）機能について説明します。

原則として、IPv4 VPN と IPv6 VPN との間に相違点はありません。IPv4 と IPv6 のどちらにおいても、マルチプロトコル BGP が MPLS VPN for IPv6（VPNv6）アーキテクチャの中心的存在となります。サービス プロバイダー バックボーンを介して IPv6 ルートを配布するために使用され、同じ手順を使用して、重複するアドレス、再配布ポリシー、およびスケーラビリティの問題が処理されます。

機能情報の確認

ご使用のソフトウェア リリースによっては、この章に記載されている機能の中に、一部サポートされていないものがあります。最新の機能情報と注意事項については、ご使用のプラットフォームとソフトウェア リリースに対応したリリース ノートを参照してください。この章に記載されている機能の詳細、および各機能がサポートされているリリースのリストについては、「IPv6 VPN over MPLS の実装の機能情報」（P.56）を参照してください。

Cisco Feature Navigator を使用すると、プラットフォーム、および Cisco ソフトウェア イメージの各サポート情報を検索できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn からアクセスしてください。Cisco.com のアカウントは必要ありません。

目次

IPv6 VPN over MPLS を実装するための前提条件

IPv6 VPN の動作を設定する前に、ネットワークで次の Cisco IOS サービスが稼動している必要があります。

- プロバイダー バックボーン ルータにおける MPLS
- VPN PE ルータがあるプロバイダー ルータにおける VPN コード付き MPLS
- VPN サービスを提供するすべてのルータにおける BGP
- すべての MPLS 対応ルータにおけるシスコ エクスプレス フォワーディング スイッチング
- Class of Service（CoS; サービス クラス）機能

IPv6 VPN over MPLS を実装するための制約事項

6VPE は、MPLS IPv4 シグナリング コアをサポートします。MPLS IPv6 シグナリング コアはサポートされません。

IPv6 VPN over MPLS の実装に関する情報

マルチプロトコル BGP は、IPv4 と IPv6 の両方において MPLS IPv6 VPN アーキテクチャの中心的存在です。サービス プロバイダー バックボーンを介して IPv6 ルートを配布するために使用され、同じ手順を使用して、重複するアドレス、再配布ポリシー、およびスケーラビリティの問題が処理されます。

IPv6 には重複するアドレス空間はありませんが、IPv6 アドレスの先頭には Route Distinguisher（RD; ルート識別子）が付加されます。Network Layer Reachability Information（NLRI; ネットワーク レイヤ到達可能性情報）の 3 タプル形式（長さ、IPv6 プレフィクス、およびラベルを含む）は、マルチプロトコル BGP を使用してこれらのルートを配布するように定義されます。拡張コミュニティ アトリビュート（ルート ターゲット）は、エクスポートされたルートにタグを付け、インポートされたルートをフィルタリングすることによって、ルーティング情報の再配布を制御するために使用されます。

スケーラビリティを実現するために、ルート リフレクタを使用してルーティング パスを集中させ、完全 PE メッシュを回避することができます。ルート リフレッシュ、自動ルート フィルタリング、アウトバウンド ルート フィルタリングなどの IPv6 の BGP 機能は、各 PE に保持されるルートの数を削減するのに役立ちます。

このマニュアルでは、IPv4 と IPv6 間の次の相違点を中心に説明します。

- 新しいマルチプロトコル BGP IPv6 VPN アドレス ファミリの作成と IPv6 VPN アドレス形式の仕様
- 新しい IPv6 VPN NLRI の仕様
- ルータに IPv4 ベースの MPLS コアがある場合の BGP ネクストホップ符号化の仕様

プロバイダー間トポロジおよび Carrier Supporting Carrier（CSC）トポロジなどの一部の IPv6 VPN 機能は、BGP-MPLS IPv6 VPN に固有です。たとえば、Autonomous System Boundary Router（ASBR; 自律システム境界ルータ）間のリンクは、転送されるアドレス ファミリとは関係なく、IPv4 だけ、IPv6 だけ、または両方をサポートすることがあります。

IPv6 VPN over MPLS を設定するには、次の概念を理解する必要があります。

- 「IPv6 VPN over MPLS（6VPE）のアドレッシングに関する考慮事項」（P.3）
- 「IPv6 VPN over MPLS の基本的な機能」（P.3）
- 「IPv6 MPLS VPN の高度な機能」（P.7）

IPv6 VPN over MPLS（6VPE）のアドレッシングに関する考慮事項

配置されている VPN モデル（Customer Edge（CE; カスタマー エッジ）ベース、PE ベースなど）に関係なく、ホストが 1 つの VPN 内の 1 つのサイトを使用して他のサイトやパブリック リソースと通信できるように、VPN のアドレッシング計画を定義する必要があります。

VPN IPv4 サイトは、多くの場合、アドレッシング計画にプライベート アドレッシングを使用します。これらのアドレスは、登録の必要はありませんが、パブリック ネットワーク上ではルーティング不可になります。プライベート サイト内のホストでパブリック ドメインにアクセスする必要がある場合、ホストは常に、そのホストの代わりにパブリック アドレスを検出するデバイスを通過します。IPv4 では、このデバイスに、ネットワーク アドレス変換またはアプリケーション プロキシを指定できます。

IPv6 ではより大きいアドレス空間を使用できるため、IPv6 アドレッシングを実現する最も簡単なアプローチは、プライベート アドレッシング計画に IPv6 グローバル アドレスを使用することです。また別のアプローチとして、Unique Local Address（ULA; ユニーク ローカル アドレス）を使用することもできます。ULA は、それらのスコープに基づいてサイト境界で簡単にフィルタリングできます。また、ULA は Internet Service Provider（ISP; インターネット サービス プロバイダー）非依存であり、永続的または間欠的なインターネット接続がないサイト内での通信に使用できます。

6VPE では、ULA は通常のグローバル アドレスとして処理されます。ULA プレフィクスは、パブリック ドメイン内に表示されないように、ルータ設定によってフィルタリングされます。ピア上のリンク ローカル アドレスが、BGP（IPv6 または IPv6 VPN）スピーカーによってアナウンスされることはありません。

パブリック ドメインにアクセスする必要があるプライベート サイト内のホストは、ルーティング可能なグローバル アドレスを使用してホストの代わりにパブリック リソースにアクセスする IPv6 アプリケーション プロキシ（Web ページにアクセスするための Web プロキシなど）を介して、これを行うことができます。または、ホスト自身のパブリック アドレスを使用することもできます。後者の場合、ULA が配置されているときには、IPv6 ホストもまたルーティング可能なグローバル アドレスを使用して設定されます。送信元アドレスの選択アルゴリズムを使用して、宛先アドレスを基にどちらか片方が選択されます。

IPv6 VPN over MPLS の基本的な機能

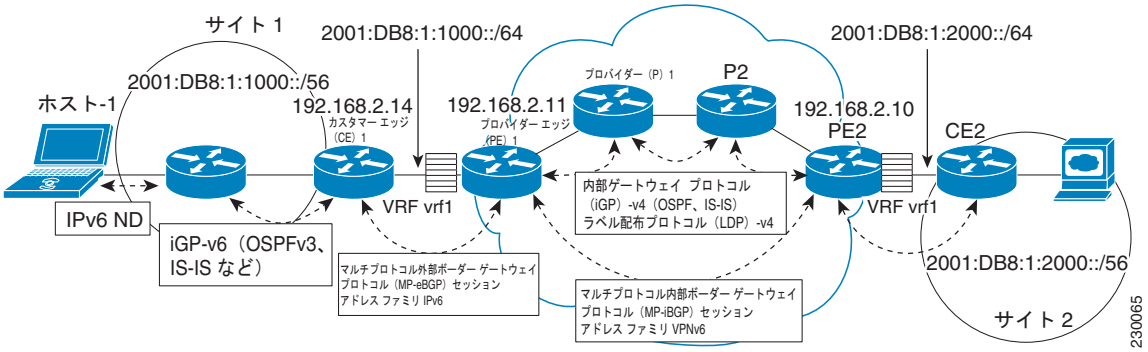
IPv6 への移行により、IPv4 と IPv6 の共存が長期間続くことになると予想されます。IPv6 VPN の配置方式では、既存の MPLS IPv4 コア ネットワークを活用することによってこの共存をうまく利用します。このアプローチを 6VPE と呼びます。

ここでは、基本的な IPv6 MPLS VPN 機能の概念について説明します。

IPv6 VPN アーキテクチャの概要

図 1 に、IPv6 VPN アーキテクチャの重要な特徴を示します。

図 1　簡単な IPv6 VPN アーキテクチャ



CE ルータは、PE ルータを使用して、プロバイダーのバックボーンに接続されます。PE ルータは、プロバイダー（図 1 の P1 および P2）ルータを使用して接続されます。プロバイダー（P）ルータは VPN ルートを認識せず、6VPE の場合は、IPv4 しかサポートしていないこともあります。PE ルータだけが VPN 固有の作業を実行します。6VPE の場合、PE ルータはデュアル スタック（IPv4 および IPv6）ルータになります。

VPN 動作のルーティング コンポーネントは、コア ルーティングとエッジ ルーティングに分けられます。PE ルータと P ルータを含むコア ルーティングは、一般的に Open Shortest Path First（OSPF）または Intermediate System-to-Intermediate System（IS-IS）などの IPv4 Interior Gateway Protocol（IGP; 内部ゲートウェイ プロトコル）によって実行されます。図 1 の場合、IGP は内部ルートだけをプロバイダーの自律システムに配布します。コア ルーティングにより、P および PE ルータ間の接続が可能になります。

エッジ ルーティングは、PE ペア間のルーティングと PE と CE 間のルーティングの 2 方向で発生します。PE ペア間のルーティングは、IPv6 VPN アドレス ファミリを使用したマルチプロトコル internal BGP（iBGP; 内部 BGP）を使用して実行されます。この方式は、入力 PE ルータでは適切なルート エクスポート ポリシーを、出力 PE ルータでは適切なルート インポート ポリシーを使用して、PE-CE ルーティングを介して CE から学習したルートを配布します。

CE とその PE 間のルーティングは、VRF 対応のルーティング プロトコルを使用して実行されます。スタティック ルート、external BGP（eBGP; 外部 BGP）、および Enhanced Interior Gateway Routing Protocol（EIGRP）は、VPN Routing and Forwarding（VRF; VPN ルーティングおよび転送）インスタンス対応です。図 1 の場合、CE（CE1）と PE（PE1）間で eBGP が使用されます。同時に、VPN サイト（図 1 のサイト 1）内では、CE によって IPv6 IGP（IPv6 用の OSPFv3 または IS-IS など）が実行されます。CE は、IGP ルートをマルチプロトコル eBGP アドレス ファミリ IPv6 に再配布します。これらのルートは、PE で vrf1 という名前の VRF にインストールされ、この VRF に定義されているエクスポート ポリシーに応じて、リモート PE（図 1 の PE2）に転送されます。

IPv6 VPN ネクストホップ

ルータが MP_REACH_NLRI アトリビュートを使用してプレフィクスをアナウンスすると、1 つの PE で稼動している MP-BGP が、リモート PE に送信されるアップデート メッセージ内に BGP ネクストホップを挿入します。このネクストホップは、受信されたアップデートから伝播されるか（たとえば、PE がルート リフレクタの場合など）、またはアップデート メッセージを送信する PE（出力 PE）のアドレスになります。

IPv6 VPN アドレス ファミリの場合、PE スピーカー間のネットワークの特性に関係なく、ネクストホップは IPv6 VPN アドレスである必要があります。RD は意味を持たないため（アドレスは VPN の一部ではない）、0 に設定されます。プロバイダー ネットワークがネイティブ IPv6 ネットワークの場

合、ネクストホップの残りの部分は出力 PE の IPv6 アドレスになります。それ以外の場合は、IPv6 マッピング アドレスとして使用される IPv4 アドレスになります（たとえば、::FFFF:*IPv4-address* など）。

IPv6 VPN ネクストホップの設定例については、「例：IPv4 ネクストホップを使用した IPv6 VPN の設定」（P.53）を参照してください。

MPLS 転送

1 つのカスタマー サイトから IPv6 トラフィックを受信すると、入力 PE ルータは MPLS を使用して、BGP ネクストホップとして識別された出力 PE ルータに向けて、バックボーンを介して IPv6 VPN パケットをトンネリングします。入力 PE ルータは、一般的に IPv6 パケットの先頭に外部ラベルおよび内部ラベルを付加してから、出力インターフェイスにパケットを配置します。

通常の動作では、転送パス上の P ルータは最初のラベルの先にあるフレームの内部を調べません。P ルータは着信ラベルを発信ラベルと交換するか、または次のルータが PE ルータの場合には着信ラベルを削除します。着信ラベルの削除は、最後から 2 番めのホップのポッピングと呼ばれます。残りのラベル（BGP ラベル）は、カスタマー サイトへの出力 PE インターフェイスを識別するために使用されます。また、このラベルは、プロトコル バージョン（IPv6）を最後の P ルータから隠します。このようにしなかった場合、最後の P ルータで IPv6 パケットを転送する必要があります。

P ルータは IPv6 VPN ルートを認識しません。IPv6 ヘッダーは 1 つ以上の MPLS ラベルの下に隠されたままになります。P ルータで、送達できない MPLS カプセル化 IPv6 パケットを受信した場合のオプションは 2 つあります。P ルータが IPv6 対応の場合、IPv6 ヘッダーを公開し、IPv6 メッセージ用の Internet Control Message Protocol（ICMP; インターネット制御メッセージ プロトコル）を構築して、MPLS カプセル化メッセージを元のパケットの送信元に送信します。P ルータが IPv6 対応ではない場合、パケットはドロップされます。

GRE トンネルを介した 6VPE

一部の Cisco IOS リリースでは、入力 PE ルータは、MPLS を介した 6VPE と組み合せた IPv4 Generic Routing Encapsulation（GRE; 総称ルーティング カプセル化）トンネルを使用して、BGP ネクストホップとして識別された出力 PE ルータに向けて、バックボーンを介して IPv6 VPN パケットをトンネリングします。

VRF の概念

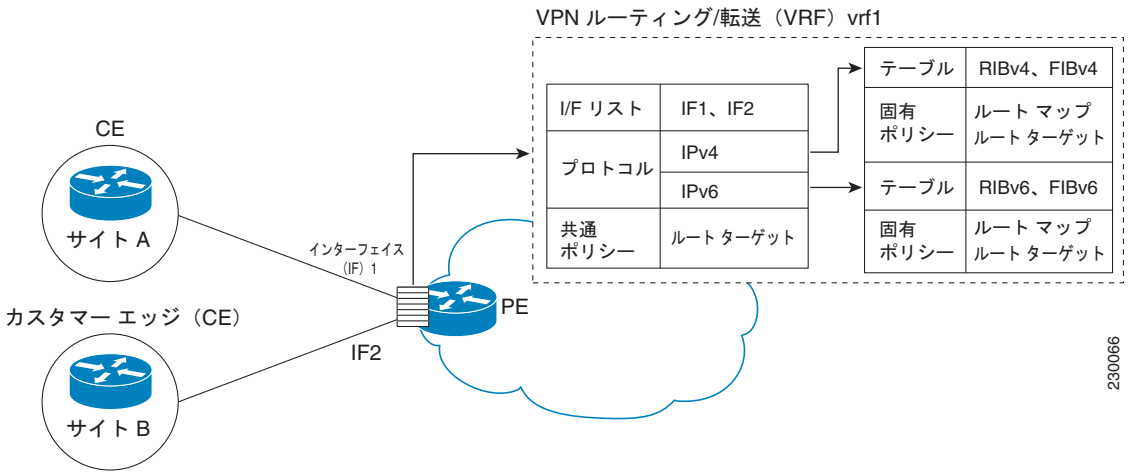
VRF は、プライベートなカスタマー固有の Routing Information Base（RIB; ルーティング情報ベース）および Forwarding Information Base（FIB; 転送情報ベース）とともに動作する仮想ルーティングおよび転送エンティティです。IPv4 ルーティング テーブルと IPv6 ルーティング テーブルは区別されますが、2 つのプロトコルで特定のカスタマーの同じ VRF を共有すると便利です。

IPv6 VPN カスタマーは、デュアル スタック ホストとルータを配置しているか、または IPv4 インフラストラクチャの一部を IPv6 ノードで覆っている既存の VPNv4 カスタマーである可能性があります。複数の配置モデルが可能です。一部のカスタマーは、IPv4 と IPv6 に別々の論理インターフェイスを使用して、それぞれに異なる VRF を定義しています。このアプローチでは IPv4 および IPv6 に別々のポリシーを設定できる柔軟性が提供されますが、同じポリシーを共有することはできなくなります。もう 1 つのアプローチのマルチプロトコル VRF では、PE-CE インターフェイス上で単一の VRF を保持し、IPv4、IPv6、または両方に対してイネーブルにします。これにより、共通または別々のポリシーを IP バージョンごとに定義できるようになります。このアプローチを使用すると、VRF は、PE で検出されるテーブル、インターフェイス、およびポリシーのセットとしてより適切に定義され、この PE に接続されている特定の VPN のサイトによって使用されます。

図 2 に、マルチプロトコル VRF を示します。ここでは、vrf1 という名前の VRF が IPv4 と IPv6 の両方に対してイネーブルになっており、2 つのインターフェイス（IF1、IF2）、2 つのテーブル セット（IPv4 RIB と FIB、IPv6 RIB と FIB）、および共通または個別のポリシー セットに関連付けられています。

IPv6 の VRF を設定する方法については、「IPv6 用の仮想ルーティングおよび転送インスタンスの設定」（P.10）を参照してください。

図 2　　マルチプロトコル VRF



IPv6 VPN スケーラビリティ

BGP-MPLS IPv6 VPN などの PE ベースの VPN は、CE ベースの VPN よりもスケーラビリティが高くなります。ネットワーク設計者は、ネットワークの設計時にスケーリングを考慮する必要があります。BGP-MPLS IPv6 VPN のスケーリングは、BGP-MPLS IPv4 VPN のスケーリングと似ています。次の点について考慮する必要があります。

- VRF テーブル サイズおよび BGP テーブル サイズなどのルーティング テーブル サイズ
- PE の平方数として増加する BGP セッション数

ルーティング テーブル サイズに関する問題は、多数のカスタマー サイトを処理する PE で発生します。これらの PE は、接続されているカスタマーごとに 1 つの RIB および FIB を持つだけではなく、PE の BGP テーブル（個々の VRF のすべてのエントリが統合される）もそれに応じて増加します。もう 1 つのスケーラビリティの問題は、プロバイダー ネットワーク内の PE の数が一定のレベル以上に増加したときに発生します。同じ VPN に属する多くのサイトが多数の PE に広がっていると想定した場合、マルチプロトコル BGP セッションの数は $(n–1) \times n/2$ のように急速に増加します。ここで、n は PE の数です。

IPv6 VPN over MPLS には、次の機能が含まれています。

- ルート リフレッシュおよび自動ルート フィルタリング：VRF にインポートされたルートだけがローカルに保持されるため、ルーティング テーブルのサイズが制限されます。インポート ポリシーが変更された場合は、ルート リフレッシュを送信して、ルーティング アップデートの再送信を照会できます。
- Outbound Route Filtering（ORF; アウトバウンド ルート フィルタリング）：アップデートがネットワーク上に不必要に送信されないように、入力 PE が出力 PE にフィルタをアドバタイズできるようにします。
- ルート リフレクタ：Route Reflector（RR; ルート リフレクタ）は、他の iBGP ピアから学習した iBGP ルートを伝播する iBGP ピアです。RR は iBGP セッションを集中させるために使用されます。

IPv6 MPLS VPN の高度な機能

IPv4 用の VPN からインターネットへのアクセス、マルチ自律システム バックボーン、CSC などの高度な MPLS 機能は、一般的に IPv6 でも IPv4 でも同じです。ただし、アドレッシングと、IPv4 バックボーンを介した 6VPE の動作方法には相違点があります。

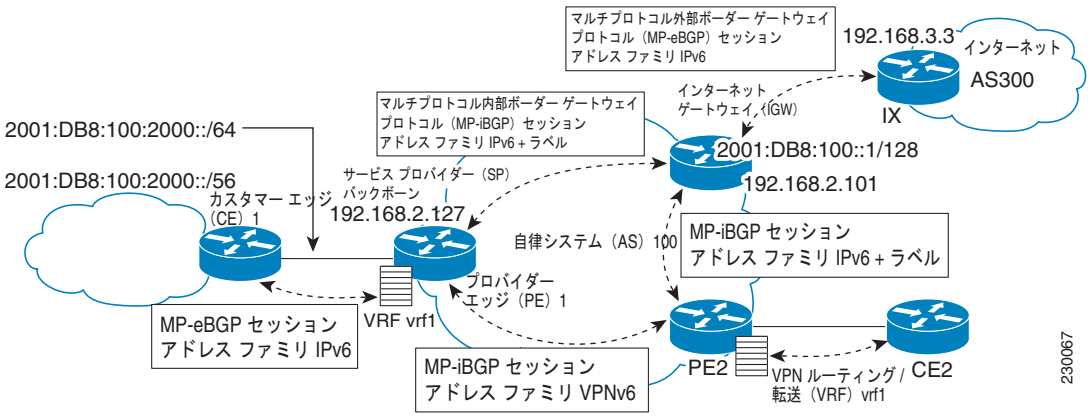
ここでは、高度な IPv6 MPLS VPN 機能の概念について説明します。

内部アクセス

大部分の VPN サイトでインターネットへのアクセスが必要になります。RFC 4364 には、インターネットへの VPN アクセスをイネーブルにするモデル セットが記載されています。これらすべてのモデルが IPv6 VPN にも適用されます。あるアプローチでは、1 つのインターフェイスがインターネットに接続するために CE によって使用され、別のインターフェイスが VRF に接続するために使用されます。別のモデルでは、すべてのインターネット ルートが VRF に再配布されます。このアプローチには、VRF ごとにインターネット ルートを複製する必要があるというデメリットがあります。

あるシナリオでは、IPv6 デフォルト テーブルで見つかったインターネット ゲートウェイを指すネクストホップとともに、スタティック ルートが VRF テーブルに挿入されます。図 3 に、このシナリオを示します。ここでは、インターネット アクセスが vrf1 という名前の VRF 内のカスタマーに提供されます。

図 3　　インターネット アクセス トポロジ



カスタマー サイト（図 3 のサイト 1）がインターネット経由でパブリック リソースにアクセスするには、このサイトがパブリック プレフィクスによって認識されている必要があります。IPv4 とは異なり、IPv6 では、サイト境界から発信されたときにプライベート アドレスをパブリック アドレスに変換できるようにする Network Address Translation（NAT; ネットワーク アドレス変換）メカニズムは提供されません。これは、サイト内のホストがパブリック アドレスを使用して発信することだけではなく、これらのアドレス（またはそれらが属するプレフィクス）がパブリック ドメイン内に表示される必要があることも意味します。

発信トラフィックの場合、入力 PE（PE1）の VRF テーブルに設定されているデフォルト ルートは、VPN 外の宛先に向かうトラフィックをインターネット ゲートウェイに誘導します。

着信トラフィックの場合、カスタマー サイトに向かうトラフィックを接続 PE（図 3 の PE1）経由で誘導するためのルートが、インターネット ゲートウェイに存在している必要があります。このルートは、入力 PE（PE1）により、（IPv6 アドレス ファミリ設定を含む）マルチプロトコル iBGP を使用して配布されます。そのため、インターネット ゲートウェイの VPN PE ごとに特別な設定を行う必要はありません。それでもなお、PE1 の着信トラフィックの場合は、サイトの VRF を指すカスタマー サイト グローバル プレフィクスのデフォルト テーブルに、ルートが存在している必要があります。

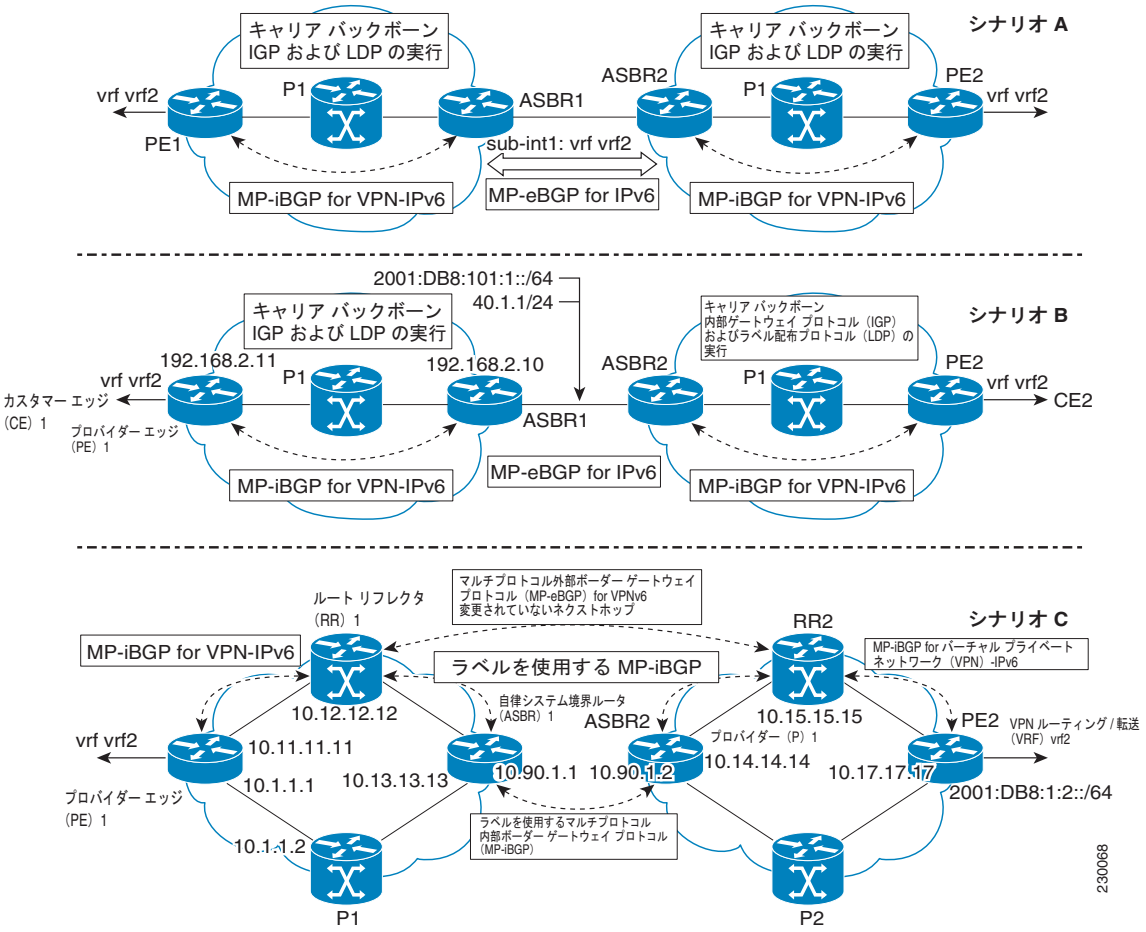
マルチ自律システム バックボーン

IPv4 が配置されていたすべての場所に IPv6 が配置されていると想定した場合、プロバイダー間 VPN の問題は、IPv6 および IPv4 で似ています。

自律システム境界を横断する IPv6 配置の場合、プロバイダーはピアリング モデルを入手するか、または VPNv4 用に設置されているピアリング モデルを使用する必要があることがあります。

図 4 に、IPv6 VPN のプロバイダー間シナリオを示します。

図 4　　プロバイダー間シナリオ



ASBR 間で使用されるネットワーク プロトコルに応じて、図 4 の 3 つのシナリオで複数の実装オプションを使用できます。たとえば、ASBR 間のマルチプロトコル eBGP IPv6 VPN ピアリングを提案しているシナリオ B では、IPv6 または IPv4 リンクのどちらでも使用できます。

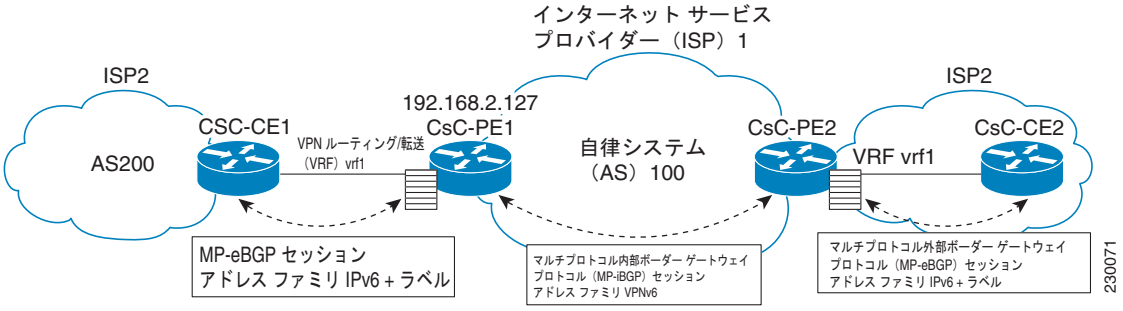
シナリオ C の場合、マルチホップ マルチプロトコル eBGP は、個々の自律システムのルート リフレクタ全体に IPv6 VPN ルートを再配布します。PE へのラベル付き IPv4 ルートは（6VPE の場合）、完全なラベル付きスイッチ パスがエンドツーエンドで設定されるように、ASBR 全体にアドバタイズされる必要があります。

Carrier Supporting Carrier

CSC 機能はカスタマー サービス プロバイダーに VPN アクセスを提供します。そのため、このサービスでは ISP MPLS バックボーンを介してルートを交換し、トラフィックを送信する必要があります。通常の PE との唯一の違いは、CSC-CE から CSC-PE へのインターフェイス上に、IP から MPLS への転送ではなく MPLS から MPLS への転送を提供することです。

図 5 に、2 つの ISP のインターフェイスの重要点を示します。

図 5　　CSC 6VPE の設定例



IPv6 用の BGP-MPLS VPN に CSC を設定する方法については、「IPv6 VPN 用の CSC の設定」（P.45）を参照してください。

IPv6 VPN over MPLS の実装方法

IPv6 用の仮想ルーティングおよび転送インスタンスの設定

VRF は、サポートされているアドレス ファミリごとにイネーブルにしたり設定したりできる、アドレス ファミリ非依存のオブジェクトです。VRF の設定は、次の 3 つの手順で構成されています。

1. VRF のアドレス ファミリ非依存部分の設定
2. VRF を使用するための IPv4 のイネーブル化および設定
3. VRF を使用するための IPv6 のイネーブル化および設定

VRF には名前および RD が与えられます。RD は特定の BGP アドレス ファミリのコンテキスト内にある重複するアドレスを区別するために使用されるものですが、アドレス ファミリのコンテキスト外で設定されます。IPv4 VPN アドレスと IPv6 VPN アドレスとで別々の RD を持っていても問題はありません。Cisco ルータでは、設定および VPN 管理を簡素化するために RD は同じになっています。

アドレス ファミリ コンテキストを使用していない場合、ユーザは IPv4 と IPv6 間で共通のポリシーを設定できます。この機能はルート ターゲット（インポートおよびエクスポート）共有であり、IPv4 ポリシーがすでに設定済みで、IPv6 ポリシーを IPv4 ポリシーと同じようにする必要がある移行シナリオで役立ちます。

IPv4 および IPv6 アドレス ファミリは、それぞれ個別にイネーブル化したり設定したりできます。このレベルで入力したルート ターゲット ポリシーは、アドレス ファミリ非依存の設定時に指定されている可能性のあるグローバル ポリシーに優先することに注意してください。

手順の概要

1. **enable**
2. **configure terminal**
3. **mls ipv6 vrf**
4. **vrf definition** *vrf-name*
5. **rd** *route-distinguisher*
6. **route-target** {**import** | **export** | **both**} *route-target-ext-community*
7. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*]
8. **route-target** {**import** | **export** | **both**} *route-target-ext-community*
9. **exit**
10. **address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast**]
11. **route-target** {**import** | **export** | **both**} *route-target-ext-community*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • 必要に応じてパスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	**mls ipv6 vrf** **例：** `Router(config)# mls ipv6 vrf`	VRF 内で IPv6 をグローバルにイネーブルにします。
ステップ 4	**vrf definition** *vrf-name* **例：** `Router(config)# vrf definition vrf1`	VPN VRF ルーティング テーブルを設定し、VRF コンフィギュレーション モードを開始します。
ステップ 5	**rd** *route-distinguisher* **例：** `Router(config-vrf)# rd 100:1`	VRF の RD を指定します。
ステップ 6	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **例：** `Router(config-vrf)# route target import 100:10`	IPv4 と IPv6 の両方のルート ターゲット VPN 拡張コミュニティを指定します。
ステップ 7	**address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] **例：** `Router(config)# address-family ipv4`	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、標準 IPv4 アドレス プレフィクスを使用するルーティング セッションを設定します。
ステップ 8	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **例：** `Router(config-vrf-af)# route target import 100:11`	IPv4 固有のルート ターゲット VPN 拡張コミュニティを指定します。
ステップ 9	**exit** **例：** `Router(config-vrf-af)# exit`	この VRF のアドレス 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 10	**address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast**] **例：** `Router(config-vrf)# address-family ipv6`	標準 IPv6 アドレス プレフィクスを使用する 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	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **例：** `Router(config-vrf-af)# route target import 100:12`	IPv6 固有のルート ターゲット VPN 拡張コミュニティを指定します。

インターフェイスへの VRF のバインド

次の作業では、VRF をインターフェイスにバインドする方法を示します。どのインターフェイスがどの VRF に属するかを指定するために、IPv4 と IPv6 の両方に対して **vrf forwarding** コマンドを使用します。インターフェイスは、複数の VRF に属すことはできません。インターフェイスが VRF にバインドされると、以前に設定したアドレス（IPv4 および IPv6）は削除されるため、再設定が必要になります。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **vrf forwarding** *vrf-name*
5. **ip address** *ip-address mask* [**secondary**]
6. **ipv6 address** {*ipv6-address*/*prefix-length* | *prefix-name sub-bits*/*prefix-length*}

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • 必要に応じてパスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *type number* **例：** `Router(config)# interface Ethernet 0/0`	インターフェイスのタイプと番号を指定し、ルータをインターフェイス コンフィギュレーション モードにします。
ステップ 4	**vrf forwarding** *vrf-name* **例：** `Router(config-if)# vrf forwarding vrf1`	VPN VRF をインターフェイスまたはサブインターフェイスに関連付けます。 このコマンドを入力する前に設定されていたアドレス（IPv4 または IPv6）はすべて削除されることに注意してください。
ステップ 5	**ip address** *ip-address mask* [**secondary**] **例：** `Router(config-if)# ip address 10.10.10.1 255.255.255.0`	インターフェイスに IPv4 アドレスを設定します。
ステップ 6	**ipv6 address** {*ipv6-address*/*prefix-length* | *prefix-name sub-bits*/*prefix-length*} **例：** `Router(config-if)# ipv6 address 2001:DB8:100:1::1/64`	インターフェイスに IPv6 アドレスを設定します。

PE から CE へのルーティングのためのスタティック ルートの設定

手順の概要

1. **enable**
2. **configure terminal**
3. **ipv6 route** [**vrf** *vrf-name*] *ipv6-prefix*/*prefix-length* {*ipv6-address* | *interface-type interface-number* [*ipv6-address*]} [**nexthop-vrf** [*vrf-name1* | **default**]] [*administrative-distance*] [*administrative-multicast-distance* | **unicast** | **multicast**] [*next-hop-address*] [**tag** *tag*]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • 必要に応じてパスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ipv6 route** [**vrf** *vrf-name*] *ipv6-prefix*/*prefix-length* {*ipv6-address* | *interface-type interface-number* [*ipv6-address*]} [**nexthop-vrf** [*vrf-name1* | **default**]] [*administrative-distance*] [*administrative-multicast-distance* | **unicast** | **multicast**] [*next-hop-address*] [**tag** *tag*] **例：** `Router(config)# ipv6 route vrf vrf1 ::/0 2001:DB8:200::1 nexthop-vrf default`	指定したネクストホップを使用して、指定した IPv6 スタティック ルートをインストールします。

eBGP の PE から CE へのルーティング セッションの設定

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast**]
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • 必要に応じてパスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router bgp** *autonomous-system-number* **例：** `Router(config)# router bgp 100`	BGP ルーティング プロセスを設定します。
ステップ 4	**address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast**] **例：** `Router(config-router)# address-family ipv6 vrf vrf1`	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* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **例：** `Router(config-router-af)# neighbor 2001:DB8:100:1::2 remote-as 200`	マルチプロトコル BGP ネイバーテーブルにエントリを追加します。
ステップ 6	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **例：** `Router(config-router-af)# neighbor 2001:DB8:100:1::2 activate`	このアドレス ファミリの情報を、指定した BGP ネイバーと交換できるようにします。

iBGP 用の IPv6 VPN アドレス ファミリの設定

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **address-family vpnv6** [**unicast**]
7. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**

8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
9. **exit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • 必要に応じてパスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router bgp** *autonomous-system-number* **例：** `Router(config)# router bgp 100`	BGP ルーティング プロセスを設定します。
ステップ 4	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **例：** `Router(config-router)# neighbor 192.168.2.11 remote-as 100`	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。 • IPv6 VPN の場合、BGP セッションを IPv4 ベースのコア ネットワークで転送できるようにするために、ピア アドレスは一般的に IPv4 アドレスになります。
ステップ 5	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **例：** `Router(config-router)# neighbor 192.168.2.11 update-source Loopback 0`	BGP セッションで、指定したインターフェイスの送信元アドレスを使用できるようにします。
ステップ 6	**address-family vpnv6** [**unicast**] **例：** `Router(config-router)# address-family vpnv6`	ルーティング セッションを設定するために、ルータをアドレス ファミリ コンフィギュレーション モードに設定します。
ステップ 7	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **例：** `Router(config-router-af)# neighbor 192.168.2.11 activate`	このアドレス ファミリの情報を、指定した BGP ネイバーと交換できるようにします。

	コマンドまたはアクション	目的
ステップ 8	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**] **例：** `Router(config-router-af)# neighbor 192.168.2.11 send-community extended`	コミュニティ アトリビュートを 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 9	**exit** **例：** `Router(config-router-af)# exit`	アドレス ファミリ コンフィギュレーション モードを終了します。

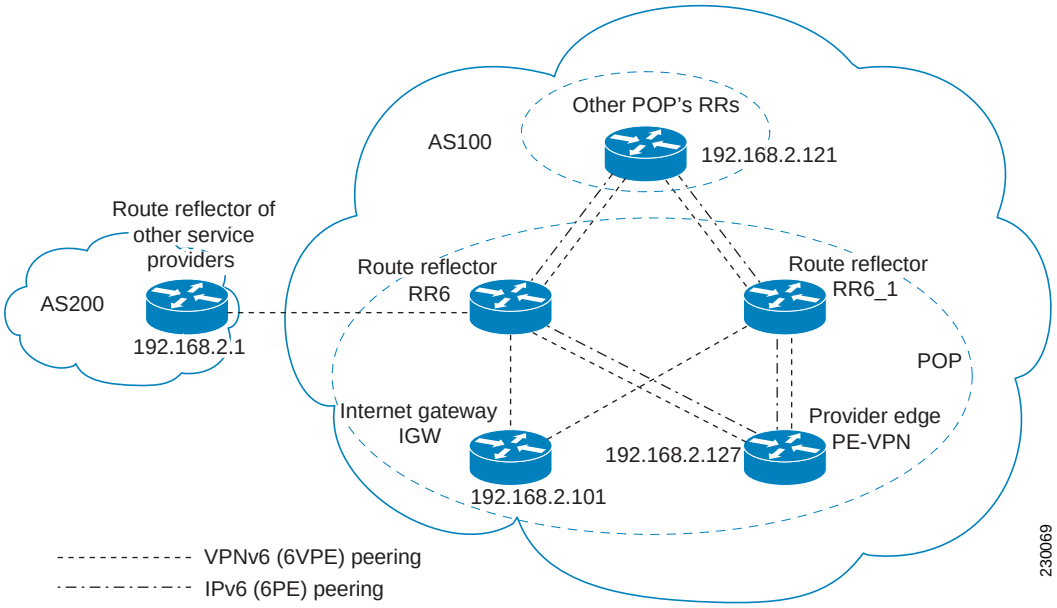
スケーラビリティ向上のためのルート リフレクタの設定

RR を配置すると、BGP セッションの数が大幅に削減され、これによりスケーラビリティが向上します。通常、1 つの RR が多数の iBGP スピーカーとピアリングして、BGP セッションの完全メッシュが防止されます。

MPLS ベースのコアの場合、RR はラベル スイッチ パスの一部ではないため、ネットワーク内の任意の場所に配置できます。たとえば、フラットな RR 設計では、RR はレベル 1 の Point of Presence（POP）に配置でき、完全メッシュ トポロジで互いにピアリングします。階層型の RR 設計では、RR はレベル 1 とレベル 2 の POP に配置でき、レベル 1 POP で互いにピアリングし、レベル 2 の RR ともピアリングします。

既存の MPLS ネットワーク（つまり、VPNv4 サービスを提供するネットワーク）に 6VPE を配置する一般的なケースでは、一部の RR 設計がすでに実施されている可能性が高く、IPv6 VPN サービス用に同様の RR インフラストラクチャを配置できます。図 6 に、ISP POP 内の RR とその RR クライアント セット間の主なピアリング ポイントを示します。

図 6　　ルート リフレクタのピアリング設計



この作業では、冗長性の理由から 2 つの RR が設定されていることに注意してください。

次のリストの BGP RR クライアントを、各 POP の IPv6 RR（図 6 の RR6 および RR6_1）ルータごとに設定する必要があります。

- ISP カスタマーに IPv6 VPN アクセスを提供する POP の PE ルータ（PE-VPN）。これには、カスタマー サイトを相互接続するための IPv6 VPN（6VPE）ピアリングと、VPN カスタマーにインターネット アクセスを提供するための IPv6 ピアリング（6PE）の両方が含まれます（「インターネット アクセスの設定」（P.22）を参照）。
- PE カスタマーに IPv6 インターネットへのアクセスを提供するために、POP 内に配置される Internet Gateway（IGW; インターネット ゲートウェイ）（「インターネット アクセスの設定」（P.22）を参照）。
- 他のサービス プロバイダーの RR。この機能は、相互自律システムの接続を提供するために使用されるもので、IPv6 と IPv6 VPN ピアリングの両方が含まれます。このサービスについては、「IPv6 VPN 用のマルチ自律システム バックボーンの設定」（P.30）で説明しています。
- 他の POP 内の RR。IPv6 と IPv6 VPN の両方のアドレス ファミリがイネーブルになっている場合、すべての RR が互いにピアリングします。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
9. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
10. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
11. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
12. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **ebgp-multihop** [*ttl*]
13. **address-family ipv6**
14. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
15. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**
16. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client**
17. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
18. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**
19. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client**
20. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
21. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**
22. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client**

23. **exit**
24. **address-family vpnv6** [**unicast**]
25. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
26. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
27. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client**
28. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
29. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
30. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client**
31. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
32. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
33. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client**
34. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **next-hop-unchanged** [**allpaths**]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • 必要に応じてパスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router bgp** *autonomous-system-number* **例：** `Router(config)# router bgp 100`	BGP ルーティング プロセスを設定します。
ステップ 4	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **例：** `Router(config-router)# neighbor 192.168.2.101 remote-as 100`	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て、インターネット アクセスを提供するためにインターネット ゲートウェイとのピアリングを提供します。
ステップ 5	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **例：** `Router(config-router)# neighbor 192.168.2.101 update-source Loopback 0`	BGP セッションで、指定したインターフェイスの送信元アドレスを使用できるようにします。

	コマンドまたはアクション	目的
ステップ 6	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **例：** `Router(config-router)# neighbor 192.168.2.121 remote-as 100`	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て、他の POP の RR とのピアリングを提供します。
ステップ 7	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **例：** `Router(config-router)# neighbor 192.168.2.121 update-source Loopback 0`	BGP セッションで、指定したインターフェイスの送信元アドレスを使用できるようにします。
ステップ 8	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **例：** `Router(config-router)# neighbor 192.168.2.127 remote-as 100`	マルチプロトコル BGP ネイバーテーブルにエントリを追加します。
ステップ 9	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **例：** `Router(config-router)# neighbor 192.168.2.127 update-source Loopback 0`	BGP セッションで、指定したインターフェイスの送信元アドレスを使用できるようにします。
ステップ 10	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **例：** `Router(config-router)# neighbor 192.168.2.1 remote-as 200`	（任意）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て、VPN 間サービスを提供するためにピア ISP の RR とのピアリングを提供します。
ステップ 11	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **例：** `Router(config-router)# neighbor 192.168.2.1 update-source Loopback 0`	（任意）BGP セッションで、指定したインターフェイスの送信元アドレスを使用できるようにします。
ステップ 12	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **ebgp-multihop** [*ttl*] **例：** `Router(config-router)# neighbor 192.168.2.1 ebgp-multihop`	（任意）直接接続されていないネットワーク上の外部ピアからの BGP 接続を受け入れ、またそのピアへの BGP 接続を試みます。
ステップ 13	**address-family ipv6** **例：** `Router(config-router)# address-family ipv6`	（任意）インターネット アクセス サービスを提供するために、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。

	コマンドまたはアクション	目的
ステップ 14	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **例：** `Router(config-router-af)# neighbor 192.168.2.101 activate`	（任意）このアドレス ファミリの情報を、指定したネイバーと交換できるようにします。
ステップ 15	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label** **例：** `Router(config-router-af)# neighbor 192.168.2.101 send-label`	（任意）BGP ルートとともに MPLS ラベルをネイバー BGP ルータに送信できるように BGP ルータを設定します。
ステップ 16	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client** **例：** `Router(config-router-af)# neighbor 192.168.2.101 route-reflector-client`	（任意）ルータを BGP ルート リフレクタとして設定し、指定したネイバーをそのクライアントとして設定します。
ステップ 17	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **例：** `Router(config-router-af)# neighbor 192.168.2.121 activate`	（任意）このアドレス ファミリの情報を、指定した BGP ネイバーと交換できるようにします。
ステップ 18	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label** **例：** `Router(config-router-af)# neighbor 192.168.2.121 send-label`	（任意）BGP ルートとともに MPLS ラベルをネイバー BGP ルータに送信できるように BGP ルータを設定します。
ステップ 19	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client** **例：** `Router(config-router-af)# neighbor 192.168.2.121 route-reflector-client`	（任意）指定したネイバーをルート リフレクタ クライアントとして設定します。
ステップ 20	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **例：** `Router(config-router-af)# neighbor 192.168.2.127 activate`	（任意）このアドレス ファミリの情報を、指定した BGP ネイバーと交換できるようにします。
ステップ 21	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label** **例：** `Router(config-router-af)# neighbor 192.168.2.127 send-label`	（任意）BGP ルートとともに MPLS ラベルをネイバー BGP ルータに送信できるように BGP ルータを設定します。

	コマンドまたはアクション	目的
ステップ 22	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client** **例：** `Router(config-router-af)# neighbor 192.168.2.127 route-reflector-client`	（任意）指定したネイバーをルート リフレクタ クライアントとして設定します。
ステップ 23	**exit** **例：** `Router(config-router-af)# exit`	（任意）アドレス ファミリ コンフィギュレーション モードを終了します。
ステップ 24	**address-family vpnv6** [**unicast**] **例：** `Router(config-router)# address-family vpnv6`	ルーティング セッションを設定するために、ルータをアドレス ファミリ コンフィギュレーション モードに設定します。
ステップ 25	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **例：** `Router(config-router-af)# neighbor 192.168.2.121 activate`	このアドレス ファミリの情報を、指定した BGP ネイバーと交換できるようにします。
ステップ 26	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**] **例：** `Router(config-router-af)# neighbor 192.168.2.21 send-community extended`	コミュニティ アトリビュートを 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 27	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client** **例：** `Router(config-router-af)# neighbor 192.168.2.121 route-reflector-client`	指定したネイバーをルート リフレクタ クライアントとして設定します。
ステップ 28	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **例：** `Router(config-router-af)# neighbor 192.168.2.127 activate`	このアドレス ファミリの情報を、指定した BGP ネイバーと交換できるようにします。
ステップ 29	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**] **例：** `Router(config-router-af)# neighbor 192.168.2.127 send-community extended`	コミュニティ アトリビュートを 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 30	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client** **例：** `Router(config-router-af)# neighbor 192.168.2.127 route-reflector-client`	指定したネイバーをルート リフレクタ クライアントとして設定します。

	コマンドまたはアクション	目的
ステップ 31	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **例：** `Router(config-router-af)# neighbor 192.168.2.1 activate`	このアドレス ファミリの情報を、指定した BGP ネイバーと交換できるようにします。
ステップ 32	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**] **例：** `Router(config-router-af)# neighbor 192.168.2.1 send-community extended`	コミュニティ アトリビュートを 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 33	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client** **例：** `Router(config-router-af)# neighbor 192.168.2.1 route-reflector-client`	指定したネイバーをルート リフレクタ クライアントとして設定します。
ステップ 34	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **next-hop-unchanged** [**allpaths**] **例：** `Router(config-router-af)# neighbor 192.168.2.1 next-hop-unchanged allpaths`	EBGP マルチホップ ピアで、パスのネクストホップを変更せずに伝播できるようにします。

インターネット アクセスの設定

大部分の VPN カスタマーは IPv4 インターネットにアクセスできます。IPv6 VPN にアクセスするカスタマーの場合は、IPv6 インターネットにアクセスできる必要があります。このサービスの設計は、グローバル インターネット アクセス サービスと似ています。レベル 1 POP に配置されている 6VPE ルータ（IGW ルータと共存）はネイティブに IGW にアクセスできますが、レベル 2 およびレベル 3 POP に配置されている、IGW に直接アクセスできない 6VPE ルータは、6PE を介して最も近いレベル 1 POP の IGW にアクセスできます。

このような 6VPE ルータで VPN インターネット アクセスを設定するには、IGW との BGP ピアリングの設定が必要になります（多くの場合、「スケーラビリティ向上のためのルート リフレクタの設定」の項で説明したように、IPv6 RR を使用します）。次に、ユーザは、プライベート ドメイン（VRF）とパブリック ドメイン（インターネット）間の通信をイネーブルにするように、相互テーブル ルーティングを設定する必要があります。

図 3 に、次の設定作業が示されています。

- 「インターネット ゲートウェイの設定」（P.22）
- 「IPv6 VPN PE の設定」（P.26）

インターネット ゲートウェイの設定

インターネット アクセス用のインターネット ゲートウェイの設定は、次の作業で構成されます。

- 「VPN PE への iBGP 6PE ピアリングの設定」（P.23）

- 「パブリック ドメインへのゲートウェイとしてのインターネット ゲートウェイの設定」（P.24）
- 「インターネットへの eBGP ピアリングの設定」（P.25）

VPN PE への iBGP 6PE ピアリングの設定

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **address-family ipv6**
7. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • 必要に応じてパスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router bgp** *autonomous-system-number* **例：** `Router(config)# router bgp 100`	BGP ルーティング プロセスを設定します。
ステップ 4	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **例：** `Router(config-router)# neighbor 192.168.2.127 remote-as 100`	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て、VPN PE とのピアリングを提供します。
ステップ 5	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **例：** `Router(config-router)# neighbor 192.168.2.127 update-source Loopback 0`	BGP セッションで、指定したインターフェイスの送信元アドレスを使用できるようにします。

	コマンドまたはアクション	目的
ステップ 6	**address-family ipv6** **例：** `Router(config-router)# address-family ipv6`	グローバル テーブルの到達可能性を交換するために、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 7	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **例：** `Router(config-router-af)# neighbor 192.168.2.127 activate`	このアドレス ファミリの情報を、指定した BGP ネイバーと交換できるようにします。
ステップ 8	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label** **例：** `Router(config-router-af)# neighbor 192.168.2.127 send-label`	BGP ルートとともに MPLS ラベルをネイバー BGP ルータに送信できるように BGP ルータを設定して、PE VPN が MPLS を介してインターネット ゲートウェイに到達できるようにします。

パブリック ドメインへのゲートウェイとしてのインターネット ゲートウェイの設定

次の作業は、「VPN PE への iBGP 6PE ピアリングの設定」（P.23）で確立した 6PE ピアリング設定を使用して、パブリック ドメインへのゲートウェイになるようにゲートウェイを設定する方法を示しています。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv6**
5. **network** *ipv6-address*/*prefix-length*
6. **exit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • 必要に応じてパスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router bgp** *autonomous-system-number* **例：** `Router(config)# router bgp 100`	BGP ルーティング プロセスを設定します。

	コマンドまたはアクション	目的
ステップ 4	**address-family ipv6** **例：** `Router(config-router)# address-family ipv6`	グローバル テーブルの到達可能性を交換するために、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 5	**network** *ipv6-address*/*prefix-length* **例：** `Router(config-router-af)# network 2001:DB8:100::1/128`	PE VPN によって使用されるネクストホップのネットワーク ソースを設定します。
ステップ 6	**exit** **例：** `Router(config-router-af)# exit`	アドレス ファミリ コンフィギュレーション モードを終了します。

インターネットへの eBGP ピアリングの設定

次の作業では、グローバル テーブルに値を格納するようにインターネットへの eBGP ピアリングを設定する方法を示します。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **address-family ipv6**
6. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
7. **aggregate-address** *address mask* [**as-set**] [**summary-only**] [**suppress-map** *map-name*] [**advertise-map** *map-name*] [**attribute-map** *map-name*]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • 必要に応じてパスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router bgp** *autonomous-system-number* **例：** `Router(config)# router bgp 100`	BGP ルーティング プロセスを設定します。

	コマンドまたはアクション	目的
ステップ 4	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **例：** `Router(config-router)# neighbor FE80::300::1%Ethernet0/0 remote-as 300`	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て、PE（PE-VPN）とのピアリングを提供します。 • ピアリングは、リンクローカル アドレス経由で行われることに注意してください。
ステップ 5	**address-family ipv6** **例：** `Router(config-router)# address-family ipv6`	グローバル テーブルの到達可能性を交換するために、アドレス ファミリ コンフィギュレーション モードを開始します。
ステップ 6	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **例：** `Router(config-router-af)# neighbor FE80::300::1%Ethernet0/0 activate`	このアドレス ファミリの情報を、指定した BGP ネイバーと交換できるようにします。
ステップ 7	**aggregate-address** *address mask* [**as-set**] [**summary-only**] [**suppress-map** *map-name*] [**advertise-map** *map-name*] [**attribute-map** *map-name*] **例：** `Router(config-router-af)# aggregate-address 2001:DB8::/32 summary-only`	集約プレフィクスを作成してから、インターネットにアドバタイズします。

IPv6 VPN PE の設定

インターネット アクセス用の IPv6 VPN PE の設定は、次の作業で構成されます。

- 「VRF からインターネット ゲートウェイへのデフォルト スタティック ルートの設定」（P.26）
- 「デフォルト テーブルから VRF へのスタティック ルートの設定」（P.27）
- 「インターネット ゲートウェイへの iBGP 6PE ピアリングの設定」（P.28）

VRF からインターネット ゲートウェイへのデフォルト スタティック ルートの設定

手順の概要

1. **enable**
2. **configure terminal**
3. **ipv6 route** [**vrf** *vrf-name*] *ipv6-prefix/prefix-length* {*ipv6-address* | *interface-type interface-number* [*ipv6-address*]} [**nexthop-vrf** [*vrf-name1* | **default**]] [*administrative-distance*] [*administrative-multicast-distance* | **unicast** | **multicast**] [*next-hop-address*] [**tag** *tag*]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • 必要に応じてパスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ipv6 route** [**vrf** *vrf-name*] *ipv6-prefix*/*prefix-length* {*ipv6-address* | *interface-type interface-number* [*ipv6-address*]} [**nexthop-vrf** [*vrf-name1* | **default**]] [*administrative-distance*] [*administrative-multicast-distance* | **unicast** | **multicast**] [*next-hop-address*] [**tag** *tag*] **例：** `Router(config)# ipv6 route vrf vrf1 ::/0 2001:DB8:100::1 nexthop-vrf default`	発信トラフィックを VRF から発信できるようにするために、VRF からインターネット ゲートウェイへのデフォルト スタティック ルートを設定します。

デフォルト テーブルから VRF へのスタティック ルートの設定

手順の概要

1. **enable**
2. **configure terminal**
3. **ipv6 route** [**vrf** *vrf-name*] *ipv6-prefix*/*prefix-length* {*ipv6-address* | *interface-type interface-number* [*ipv6-address*]} [**nexthop-vrf** [*vrf-name1* | **default**]] [*administrative-distance*] [*administrative-multicast-distance* | **unicast** | **multicast**] [*next-hop-address*] [**tag** *tag*]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • 必要に応じてパスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ipv6 route** [**vrf** *vrf-name*] *ipv6-prefix*/*prefix-length* {*ipv6-address* | *interface-type interface-number* [*ipv6-address*]} [**nexthop-vrf** [*vrf-name1* | **default**]] [*administrative-distance*] [*administrative-multicast-distance* | **unicast** | **multicast**] [*next-hop-address*] [**tag** *tag*] **例：** `Router(config)# ipv6 route 2001:DB8:100:2000::/64 nexthop-vrf vrf1`	着信トラフィックが VRF に到達できるようにするために、デフォルト テーブルから VRF へのスタティック ルートを設定します。

インターネット ゲートウェイへの iBGP 6PE ピアリングの設定

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast**]
7. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**
9. **network** *ipv6-address*/*prefix-length*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • 必要に応じてパスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router bgp** *autonomous-system-number* **例：** `Router(config)# router bgp 100`	BGP ルーティング プロセスを設定します。
ステップ 4	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **例：** `Router(config-router)# neighbor 192.168.2.101 remote-as 100`	インターネット ゲートウェイとピアリングするために、マルチプロトコル BGP ネイバーテーブルにエントリを追加します。
ステップ 5	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **例：** `Router(config-router)# neighbor 192.168.2.101 update-source Loopback 0`	BGP セッションで、指定したインターフェイスの送信元アドレスを使用できるようにします。
ステップ 6	**address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast**] **例：** `Router(config-router)# address-family ipv6`	グローバル テーブルの到達可能性を交換するために、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 7	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **例：** `Router(config-router-af)# neighbor 192.168.2.101 activate`	このアドレス ファミリの情報を、指定した BGP ネイバーと交換できるようにします。
ステップ 8	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label** **例：** `Router(config-router-af)# neighbor 192.168.2.101 send-label`	VPN PE が MPLS を介してインターネット ゲートウェイに到達できるようにするために、このネイバーに対するこのアドレス ファミリのラベル交換をイネーブルにします。
ステップ 9	**network** *ipv6-address*/*prefix-length* **例：** `Router(config-router-af)# network 2001:DB8:100:2000::/64`	VRF プレフィクスをインターネット ゲートウェイに提供します。

IPv6 VPN 用のマルチ自律システム バックボーンの設定

たとえば、VPN の 2 つのサイトが異なるサービス プロバイダーに接続されているために、それぞれ別の自律システムに接続されることがあります。この場合、その VPN に接続されている PE ルータは、iBGP 接続を互いに維持したり、共通のルート リフレクタを使用して維持したりすることはできません。このような状況では、eBGP を使用して VPN-IPv6 アドレスを配布するには、何らかの方法が必要となります。

次に、2 つのシナリオでの設定例を示します。1 つは、ASBR 間のマルチプロトコル eBGP-IPv6 VPN ピアリングで IPv4 リンクを使用し、もう 1 つは同じピアリングで IPv6 リンクを使用します。ASBR 間のピアリングが IPv4 リンク経由で実行される場合、ASBR1 の BGP 設定は次のようになります。

```
router bgp 1001
 no bgp default ipv4-unicast
 no bgp default route-target filter
 neighbor 192.1.1.1 remote-as 1002
 neighbor 192.168.2.11 remote-as 1001
 neighbor 192.168.2.11 update-source Loopback1
 !
 address-family vpnv6
!Peering to ASBR2 over an IPv4 link
 neighbor 192.1.1.1 activate
 neighbor 192.1.1.1 send-community extended
!Peering to PE1 over an IPv4 link
 neighbor 192.168.2.11 activate
 neighbor 192.168.2.11 next-hop-self
 neighbor 192.168.2.11 send-community extended
```

ASBR 間のピアリングが IPv6 リンク経由で実行される場合、ASBR1 の BGP 設定は次のようになります。

```
router bgp 1001
neighbor 2001:DB8:101::72d remote-as 1002
!
 address-family vpnv6
!Peering to ASBR2 over an IPv6 link
 neighbor 2001:DB8:101::72d activate
 neighbor 2001:DB8:101::72d send-community extended
```

次の複数の作業は、マルチホップ マルチプロトコル eBGP を使用して個々の自律システムの RR 全体に VPN ルートを再配布する、マルチ自律システム バックボーン用の PE VPN を設定する方法を示しています。PE へのラベル付き IPv4 ルートは、完全な Label Switch Path（LSP; ラベル スイッチ パス）がエンドツーエンドで設定されるように、ASBR 全体にアドバタイズされます。

このシナリオでは、ASBR は VPN 対応ではなく、RR だけが VPN 対応になっています。次の設定を有効にし、かつ理解しておく必要があります。

- ASBR では、ピアリングするサービス プロバイダーに PE のループバック アドレスを提供しています。提供される内容は次のとおりです。
  - リモート サービス プロバイダーのロケーションでネクストホップ解決をイネーブルにするための、VPN PE の IPv4 ループバック アドレス（/32）
  - プロバイダー間（RR 間）eBGP ピアリングをイネーブルにするための、VPN RR の IPv4 ループバック アドレス（/32）
- VPN PE の IPv4 ループバック アドレスの場合、ラベルがエンドツーエンド LSP を確立するように、アドレス提供は、ラベルとともにリモート PE までマルチプロトコル BGP を介して実行されます。そのため、次の MP-BGP ピアリングが VPNv4 用に設定されています。
  - VPN PE は VPN RR と iBGP ピアリングする。
  - ASBR は VPN RR と iBGP ピアリングする。

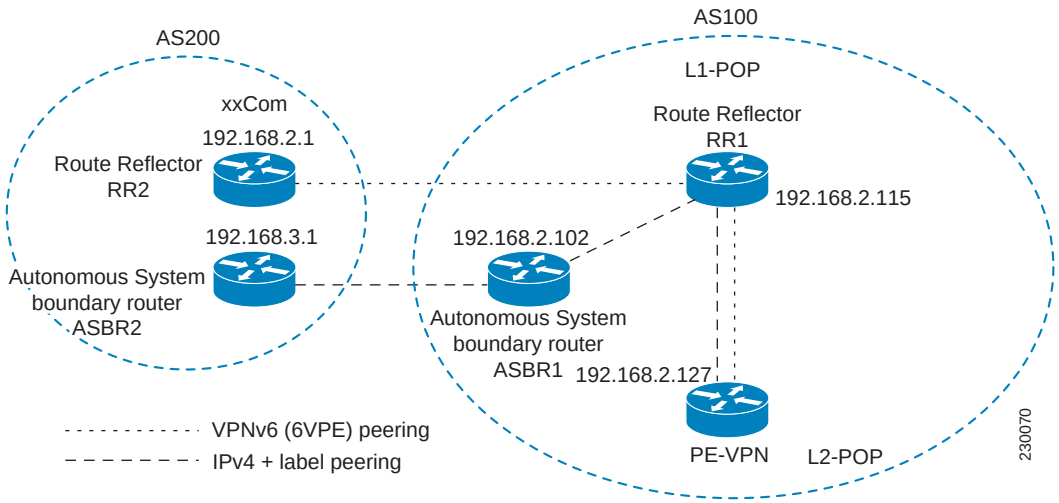
- ASBR はリモート サービス プロバイダーの ASBR と eBGP ピアリングする。

- 各サービス プロバイダーの VPN RR は、eBGP を介して互いにピアリングして、VPN ルートを交換します。エンドツーエンド LSP が RR 経由にならないように、ネクストホップは変更せずに転送されます。

このシナリオで IPv6 VPN 相互自律システム アクセスをイネーブルにするには、ISP 側で PE VPN および RR での設定を変更する必要があります。同様のサービスを VPNv4 に提供するには、同じ RR を設定します。この場合、RR と ASBR 間のピアリングおよび ASBR 間のピアリングは IPv4 VPN と IPv6 VPN の両方で使用される IPv4 ネクストホップのラベルを交換するだけなので、ASBR は完全に IPv6 非対応のままであり、ここで必要な設定変更はありません。

図 7 に、PE-VPN ルータ（IPv6 VPN アクセスを提供）から xxCom ネットワークへの IPv6 プロバイダー間接続をイネーブルにするために必要な BGP ピアリング ポイントを示します。

図 7　InterAS シナリオ C をイネーブルにするための BGP ピアリング ポイント



次に、レベル 2 POP に配置されている IPv6 VPN PE からの相互自律システム通信をイネーブルにするために必要となる、その他の BGP ピアリングのリストを示します。

- PE VPN から RR1 という名前のルート リフレクタへの、ラベルを伴う IPv4 ピアリング（VPNv4 interAS が、同じ LSP を使用して同じノードに配置されている場合は、すでに設定済み）
- RR1 から ASBR1 への、ラベルを伴う IPv4 ピアリング
- ASBR1 と ASBR1 間の、ラベルを伴う IPv4 ピアリング
- IPv6 VPN ルートを交換するための、RR1 と RR2（他の自律システムのルート リフレクタ）間の IPv6 VPN ピアリング
- RR1 との IPv6 VPN ピアリング IPv6 VPN サービスを拡張するために使用されているそのルート リフレクタが自律システム機能に使用されている場合、この機能もまたすでに設定済みである可能性があります（「スケーラビリティ向上のためのルート リフレクタの設定」（P.16）を参照）。

IPv6 VPN 用のマルチ自律システム バックボーンの設定は、次の手順で構成されます。

1. 「マルチ自律システム バックボーン用の PE VPN の設定」（P.32）
2. 「マルチ自律システム バックボーン用のルート リフレクタの設定」（P.34）
3. 「ASBR の設定」（P.42）

マルチ自律システム バックボーン用の PE VPN の設定

マルチ自律システム バックボーン用の PE VPN の設定は、次の作業で構成されます。

- 「ルート リフレクタへの iBGP IPv6 VPN ピアリングの設定」（P.32）
- 「ルート リフレクタへの IPv4 とラベルの iBGP ピアリングの設定」（P.33）

ルート リフレクタへの iBGP IPv6 VPN ピアリングの設定

次の作業では、RR1 という名前のルート リフレクタへの iBGP IPv6 VPN ピアリングを設定する方法を示します。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **address-family vpnv6** [**unicast**]
7. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address* | **peer-group-name**} **send-community** [**both** | **standard** | **extended**]
9. **exit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • 必要に応じてパスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router bgp** *autonomous-system-number* **例：** `Router(config)# router bgp 100`	BGP ルーティング プロセスを設定します。
ステップ 4	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **例：** `Router(config-router)# neighbor 192.168.2.115 remote-as 100`	相互自律システム機能を備えたルート リフレクタとピアリングするために、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。

	コマンドまたはアクション	目的
ステップ 5	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **例：** `Router(config-router)# neighbor 192.168.2.115 update-source Loopback 0`	BGP セッションで、指定したインターフェイスの送信元アドレスを使用できるようにします。
ステップ 6	**address-family vpnv6** [**unicast**] **例：** `Router(config-router)# address-family vpnv6`	（任意）ルーティング セッションを設定するために、ルータをアドレス ファミリ コンフィギュレーション モードに設定します。
ステップ 7	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **例：** `Router(config-router-af)# neighbor 192.168.2.115 activate`	このアドレス ファミリの情報を、指定した BGP ネイバーと交換できるようにします。
ステップ 8	**neighbor** {*ip-address* | *ipv6-address* | ***peer-group-name***} **send-community** [**both** | **standard** | **extended**] **例：** `Router(config-router-af)# neighbor 192.168.2.115 send-community extended`	コミュニティ アトリビュートを 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 9	**exit** **例：** `Router(config-router-af)# exit`	アドレス ファミリ コンフィギュレーション モードを終了します。

ルート リフレクタへの IPv4 とラベルの iBGP ピアリングの設定

次の作業では、RR1 という名前のルート リフレクタへの IPv4 とラベルの iBGP ピアリングを設定する方法を示します。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*]
5. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
6. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • 必要に応じてパスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router bgp** *autonomous-system-number* **例：** `Router(config)# router bgp 100`	BGP ルーティング プロセスを設定します。
ステップ 4	**address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] **例：** `Router(config-router)# address-family ipv4`	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、標準 IPv4 アドレス プレフィクスを使用するルーティング セッションを設定します。
ステップ 5	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **例：** `Router(config-router-af)# neighbor 192.168.2.115 activate`	このアドレス ファミリの情報を、指定した BGP ネイバーと交換できるようにします。
ステップ 6	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label** **例：** `Router(config-router-af)# neighbor 192.168.2.115 send-label`	エンドツーエンド LSP を設定するためのラベルとともにリモート PE ピア IPv4 ループバックを RR1 経由で受信するために、このネイバーに対するこのアドレス ファミリのラベル交換をイネーブルにします。

マルチ自律システム バックボーン用のルート リフレクタの設定

- 「PE VPN へのピアリングの設定」（P.34）
- 「ルート リフレクタの設定」（P.36）
- 「自律システム境界ルータへのピアリングの設定」（P.38）
- 「別の ISP のルート リフレクタへのピアリングの設定」（P.40）

PE VPN へのピアリングの設定

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*

5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **address-family vpnv6** [**unicast**]
7. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
9. **exit**
10. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*]
11. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
12. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**
13. **exit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • 必要に応じてパスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router bgp** *autonomous-system-number* **例：** `Router(config)# router bgp 100`	BGP ルーティング プロセスを設定します。
ステップ 4	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **例：** `Router(config-router)# neighbor 192.168.2.115 remote-as 100`	InterAS 用のルート リフレクタとピアリングするために、マルチプロトコル BGP ネイバーテーブルにエントリを追加します。
ステップ 5	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **例：** `Router(config-router)# neighbor 192.168.2.115 update-source Loopback 0`	BGP セッションで、指定したインターフェイスの送信元アドレスを使用できるようにします。
ステップ 6	**address-family vpnv6** [**unicast**] **例：** `Router(config-router)# address-family vpnv6`	（任意）ルータをアドレス ファミリ コンフィギュレーション モードに設定します。

	コマンドまたはアクション	目的
ステップ 7	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **例：** `Router(config-router-af)# neighbor 192.168.2.115 activate`	このアドレス ファミリの情報を、指定した BGP ネイバーと交換できるようにします。
ステップ 8	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**] **例：** `Router(config-router-af)# neighbor 192.168.2.115 send-community extended`	コミュニティ アトリビュートを 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 9	**exit** **例：** `Router(config-router-af)# exit`	アドレス ファミリ コンフィギュレーション モードを終了します。
ステップ 10	**address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] **例：** `Router(config-router)# address-family ipv4`	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、標準 IPv4 アドレス プレフィクスを使用するルーティング セッションを設定します。
ステップ 11	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **例：** `Router(config-router-af)# neighbor 192.168.2.115 activate`	このアドレス ファミリの情報を、指定した BGP ネイバーと交換できるようにします。
ステップ 12	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label** **例：** `Router(config-router-af)# neighbor 192.168.2.115 send-label`	エンドツーエンド LSP を設定するためのラベルとともにリモート PE IPv4 ループバックをローカル PE に送信するために、このネイバーに対するこのアドレス ファミリのラベル交換をイネーブルにします。
ステップ 13	**exit** **例：** `Router(config-router-af)# exit`	アドレス ファミリ コンフィギュレーション モードを終了します。

ルート リフレクタの設定

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*

6. **address-family vpnv6** [**unicast**]
7. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
9. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client**
10. **exit**
11. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*]
12. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
13. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**
14. **exit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • 必要に応じてパスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router bgp** *autonomous-system-number* **例：** `Router(config)# router bgp 100`	BGP ルーティング プロセスを設定します。
ステップ 4	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **例：** `Router(config-router)# neighbor 192.168.2.127 remote-as 100`	InterAS 用の VPN PE とピアリングするために、マルチプロトコル BGP ネイバーテーブルにエントリを追加します。
ステップ 5	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **例：** `Router(config-router)# neighbor 192.168.2.127 update-source Loopback 0`	BGP セッションで、指定したインターフェイスの送信元アドレスを使用できるようにします。
ステップ 6	**address-family vpnv6** [**unicast**] **例：** `Router(config-router)# address-family vpnv6`	（任意）ルータをアドレス ファミリ コンフィギュレーション モードに設定します。

	コマンドまたはアクション	目的
ステップ 7	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **例：** `Router(config-router-af)# neighbor 192.168.2.127 activate`	このアドレス ファミリの情報を、指定したネイバーと交換できるようにします。
ステップ 8	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**] **例：** `Router(config-router-af)# neighbor 192.168.2.127 send-community extended`	コミュニティ アトリビュートを 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 9	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client** **例：** `Router(config-router-af)# neighbor 192.168.2.127 route-reflector-client`	指定したネイバーをルート リフレクタ クライアントとして設定します。
ステップ 10	**exit** **例：** `Router(config-router-af)# exit`	アドレス ファミリ コンフィギュレーション モードを終了します。
ステップ 11	**address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] **例：** `Router(config-router)# address-family ipv4`	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、標準 IPv4 アドレス プレフィクスを使用するルーティング セッションを設定します。
ステップ 12	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **例：** `Router(config-router-af)# neighbor 192.168.2.127 activate`	このアドレス ファミリの情報を、指定したネイバーと交換できるようにします。
ステップ 13	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label** **例：** `Router(config-router-af)# neighbor 192.168.2.127 send-label`	エンドツーエンド LSP を設定するためのラベルとともにリモート PE IPv4 ループバックをローカル PE に送信するために、このネイバーに対するこのアドレス ファミリのラベル交換をイネーブルにします。
ステップ 14	**exit** **例：** `Router(config-router-af)# exit`	アドレス ファミリ コンフィギュレーション モードを終了します。

自律システム境界ルータへのピアリングの設定

次の作業では、ASBR1 という名前の Autonomous System Boundary Router（ASBR; 自律システム境界ルータ）へのピアリングを設定する方法を示します。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*]
7. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**
9. **exit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • 必要に応じてパスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router bgp** *autonomous-system-number* **例：** `Router(config)# router bgp 100`	BGP ルーティング プロセスを設定します。
ステップ 4	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **例：** `Router(config-router)# neighbor 192.168.2.102 remote-as 100`	ASBR1 とピアリングするために、マルチプロトコル BGP ネイバーテーブルにエントリを追加します。
ステップ 5	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **例：** `Router(config-router)# neighbor 192.168.2.102 update-source Loopback 0`	BGP セッションで、指定したインターフェイスの送信元アドレスを使用できるようにします。
ステップ 6	**address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] **例：** `Router(config-router)# address-family ipv4`	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、標準 IPv4 アドレス プレフィクスを使用するルーティング セッションを設定します。

	コマンドまたはアクション	目的
ステップ 7	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **例：** `Router(config-router-af)# neighbor 192.168.2.102 activate`	このアドレス ファミリの情報を、指定した BGP ネイバーと交換できるようにします。
ステップ 8	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label** **例：** `Router(config-router-af)# neighbor 192.168.2.102 send-label`	エンドツーエンド LSP を設定するラベルとともにリモート PE IPv4 ループバックを受信するために、このネイバーに対するこのアドレス ファミリのラベル交換をイネーブルにします。
ステップ 9	**exit** **例：** `Router(config-router-af)# exit`	アドレス ファミリ コンフィギュレーション モードを終了します。

別の ISP のルート リフレクタへのピアリングの設定

次の作業では、RR2 という名前の別の ISP のルート リフレクタへのピアリングを設定する方法を示します。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **ebgp-multihop** [*ttl*]
7. **address-family vpnv6** [**unicast**]
8. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
9. **neighbor** {*ip-address* | *ipv6-address* | **peer-group-name**} **send-community** [**both** | **standard** | **extended**]
10. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **next-hop-unchanged** [**allpaths**]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • 必要に応じてパスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router bgp** *autonomous-system-number* **例：** `Router(config)# router bgp 100`	BGP ルーティング プロセスを設定します。
ステップ 4	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **例：** `Router(config-router)# neighbor 192.168.2.1 remote-as 100`	RR2 と eBGP ピアリングするために、マルチプロトコル BGP ネイバーテーブルにエントリを追加します。
ステップ 5	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **例：** `Router(config-router)# neighbor 192.168.2.1 update-source Loopback 0`	BGP セッションで、指定したインターフェイスの送信元アドレスを使用できるようにします。
ステップ 6	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **ebgp-multihop** [*ttl*] **例：** `Router(config-router)# neighbor 192.168.2.1 ebgp-multihop`	（任意）直接接続されていないネットワーク上の外部ピアからの BGP 接続を受け入れ、またそのピアへの BGP 接続を試みます。
ステップ 7	**address-family vpnv6** [**unicast**] **例：** `Router(config-router)# address-family vpnv6`	（任意）ルーティング セッションを設定するために、ルータをアドレス ファミリ コンフィギュレーション モードに設定します。
ステップ 8	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **例：** `Router(config-router-af)# neighbor 192.168.2.1 activate`	このアドレス ファミリの情報を、指定した BGP ネイバーと交換できるようにします。

	コマンドまたはアクション	目的
ステップ 9	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**] **例：** `Router(config-router-af)# neighbor 192.168.2.1 send-community extended`	コミュニティ アトリビュートを 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 10	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **next-hop-unchanged** [**allpaths**] **例：** `Router(config-router-af)# neighbor 192.168.2.1 next-hop-unchanged allpaths`	eBGP マルチホップ ピアで、パスのネクストホップを変更せずに伝播できるようにします。

ASBR の設定

MultiAS バックボーン用の ASBR の設定は、次の作業で構成されます。

- 「ルート リフレクタ RR1 とのピアリングの設定」（P.42）
- 「他の ISP の ASBR2 とのピアリングの設定」（P.44）

ルート リフレクタ RR1 とのピアリングの設定

次の作業では、RR1 という名前のルート リフレクタとのピアリングを設定する方法を示します。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*]
7. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**
9. **exit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • 必要に応じてパスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router bgp** *autonomous-system-number* **例：** `Router(config)# router bgp 100`	BGP ルーティング プロセスを設定します。
ステップ 4	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **例：** `Router(config-router)# neighbor 192.168.2.115 remote-as 100`	RR1 とピアリングするために、マルチプロトコル BGP ネイバーテーブルにエントリを追加します。
ステップ 5	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **例：** `Router(config-router)# neighbor 192.168.2.115 update-source Loopback 0`	BGP セッションで、指定したインターフェイスの送信元アドレスを使用できるようにします。
ステップ 6	**address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] **例：** `Router(config-router)# address-family ipv4`	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、標準 IPv4 アドレス プレフィクスを使用するルーティング セッションを設定します。
ステップ 7	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **例：** `Router(config-router-af)# neighbor 192.168.2.115 activate`	このアドレス ファミリの情報を、指定した BGP ネイバーと交換できるようにします。
ステップ 8	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label** **例：** `Router(config-router-af)# neighbor 192.168.2.115 send-label`	エンドツーエンド LSP を設定するためのラベルとともにリモート PE IPv4 ループバックをローカル PE に送信するために、このネイバーに対するこのアドレス ファミリのラベル交換をイネーブルにします。
ステップ 9	exit **例：** `Router(config-router-af)# exit`	アドレス ファミリ コンフィギュレーション モードを終了します。

他の ISP の ASBR2 とのピアリングの設定

次の作業では、他の ISP の ASBR（ASBR2）とのピアリングを設定する方法を示します。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **ebgp-multihop** [*ttl*]
7. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*]
8. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
9. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**
10. **network** {*network-number* [**mask** *network-mask*] | *nsap-prefix*} [**route-map** *map-tag*]
11. **network** {*network-number* [**mask** *network-mask*] | *nsap-prefix*} [**route-map** *map-tag*]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • 必要に応じてパスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**router bgp** *autonomous-system-number* **例：** `Router(config)# router bgp 100`	BGP ルーティング プロセスを設定します。
ステップ 4	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **例：** `Router(config-router)# neighbor 192.168.3.1 remote-as 100`	ASBR2 とピアリングするために、マルチプロトコル BGP ネイバーテーブルにエントリを追加します。
ステップ 5	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **例：** `Router(config-router)# neighbor 192.168.3.1 update-source Loopback 0`	BGP セッションで、指定したインターフェイスの送信元アドレスを使用できるようにします。

	コマンドまたはアクション	目的
ステップ 6	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **ebgp-multihop** [*ttl*] **例：** `Router(config-router)# neighbor 192.168.3.1 ebgp-multihop`	直接接続されていないネットワーク上の外部ピアからの BGP 接続を受け入れ、またそのピアへの BGP 接続を試みます。
ステップ 7	**address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] **例：** `Router(config-router)# address-family ipv4`	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、標準 IPv4 アドレス プレフィクスを使用するルーティング セッションを設定します。
ステップ 8	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **例：** `Router(config-router-af)# neighbor 192.168.3.1 activate`	このアドレス ファミリの情報を、指定した BGP ネイバーと交換できるようにします。
ステップ 9	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label** **例：** `Router(config-router-af)# neighbor 192.168.3.1 send-label`	エンドツーエンド LSP を設定するためのラベルとともにリモート PE IPv4 ループバックを受信するために、このネイバーに対するこのアドレス ファミリのラベル交換をイネーブルにします。
ステップ 10	**network** {*network-number* [**mask** *network-mask*] | *nsap-prefix*} [**route-map** *map-tag*] **例：** `Router(config-router-af)# network 192.168.2.27 mask 255.255.255.255`	ネットワークをこの自律システムにローカルとしてフラグして、ネットワークを BGP テーブルに入力します。この設定は PE VPN ループバック用です。
ステップ 11	**network** {*network-number* [**mask** *network-mask*] | *nsap-prefix*} [**route-map** *map-tag*] **例：** `Router(config-router-af)# network 192.168.2.15 mask 255.255.255.255`	ネットワークをこの自律システムにローカルとしてフラグして、ネットワークを BGP テーブルに入力します。この設定は RR1 ループバック用です。

IPv6 VPN 用の CSC の設定

次の作業は、CsC-PE1 の CsC-CE1 とのピアリング設定を指定する方法を示します。

手順の概要

1. **enable**
2. **configure terminal**
3. **hostname** *name*
4. **router bgp** *autonomous-system-number*
5. **address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast**]
6. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*

7. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • 必要に応じてパスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**hostname** *name* **例：** `Router(config)# hostname CSC-PE1`	ネットワーク サーバのホスト名を指定または変更します。
ステップ 4	**router bgp** *autonomous-system-number* **例：** `Router(config)# router bgp 100`	BGP ルーティング プロセスを設定します。
ステップ 5	**address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast**] **例：** `Router(config-router)# address-family ipv6 vrf ISP2`	アドレス ファミリ コンフィギュレーション モードを開始します。
ステップ 6	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **例：** `Router(config-router-af)# neighbor FE80::866C:99%Serial0/0 remote-as 200`	マルチプロトコル BGP ネイバーテーブルにエントリを追加します。
ステップ 7	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **例：** `Router(config-router-af)# neighbor FE80::866C:99%Serial0/0 activate`	このアドレス ファミリの情報を、指定した BGP ネイバーと交換できるようにします。
ステップ 8	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label** **例：** `Router(config-router-af)# neighbor FE80::866C:99%Serial0/0 send-label`	このネイバーに対するこのアドレス ファミリのラベル交換をイネーブルにします。

IPv6 VPN の確認とトラブルシューティング

ユーザが IPv6 をトラブルシューティングする場合、VPNv4 と同様の働きをする機能は、IPv6 でも機能する可能性が高いため、新しい IPv6 ユーザの学習曲線は最小限に抑えられます。6PE および 6VPE のトラブルシューティングに使用される一部のツールおよびコマンドだけが、IPv6 に固有です。より正確に言うと、トラブルシューティング方法論は IPv4 も IPv6 も同じであり、多くの場合、コマンドおよびツールで異なるのは 1 つのキーワードだけです。

次の作業は、特定のシナリオで IPv6 VPN を確認して問題をトラブルシューティングする方法を示しています。

- 「ルーティングの確認とトラブルシューティング」（P.47）
- 「転送の確認とトラブルシューティング」（P.48）
- 「ルーティングおよび転送のデバッグ」（P.53）

ルーティングの確認とトラブルシューティング

6PE および 6VPE の配置には、主として BGP が関係します。VPNv4 に使用されているコマンド セットと同じコマンドセットを IPv6 にも使用可能であり（引数セットは異なる）、また、同様の出力が得られます。

次の例を使用すると、BGP 配置を確認およびトラブルシューティングできます。

- 「BGP IPv6 アクティビティ サマリー」（P.47）
- 「BGP IPv6 テーブルのダンプ」（P.47）
- 「IPv6 ルーティング テーブルのダンプ」（P.48）

BGP IPv6 アクティビティ サマリー

次に、BGP IPv6 アクティビティのサマリーを表示する例を示します。

```
Router# show bgp ipv6 summary

For address family: IPv6 Unicast
BGP router identifier 192.168.2.126, local AS number 33751
BGP table version is 15, main routing table version 15
12 network entries using 1692 bytes of memory
22 path entries using 1672 bytes of memory
5/4 BGP path/bestpath attribute entries using 580 bytes of memory
14 BGP rrinfo entries using 336 bytes of memory
2 BGP AS-PATH entries using 48 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 4328 total bytes of memory
Dampening enabled. 0 history paths, 0 dampened paths
BGP activity 13/1 prefixes, 23/1 paths, scan interval 60 secs
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.2.146     4 33751     991     983       15    0    0 16:26:21        10
192.168.2.147     4 33751     991     983       15    0    0 16:26:22        10
FE80::4F6B:44%Serial1/0
                4 20331     982     987       15    0    0 14:55:52         1
```

BGP IPv6 テーブルのダンプ

次の例に示すように、各テーブル（BGP IPv6、BGP IPv6 VPN など）を個別に確認できます。

```
Router# show bgp ipv6 unicast
```

```
BGP table version is 15, local router ID is 192.168.2.126
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric    LocPrf Weight Path
* i2001:DB8:100::/48 ::FFFF:192.168.2.101     0    100      0 10000 ?
*>i                  ::FFFF:192.168.2.101     0    100      0 10000 ?
* i2001:DB8::1/128   ::FFFF:192.168.2.101     0    100      0  i
*>i                  ::FFFF:192.168.2.101     0    100      0  i
```

IPv6 ルーティング テーブルのダンプ

次の例に示すように、IPv6 ルーティング テーブルを表示して、ルーティング可能なエントリを導出した各ルーティング プロトコルを確認できます。

```
Router# show ipv6 route

IPv6 Routing Table - default - 13 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, R - RIP, I1 - ISIS L1, I2 - ISIS L2
       IA - ISIS interarea, IS - ISIS summary
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
B   2001:DB8:100::/48 [200/0]
     via 192.168.2.101%Default-IP-Routing-Table, indirectly connected
B   2001:DB8::1/128 [200/0]
     via 192.168.2.101%Default-IP-Routing-Table, c
LC  2001:DB8::26/128 [0/0]
     via Loopback0, receive
```

IPv6 ルーティングの観点から見ると、MPLS バックボーンを介して到達可能なエントリが、間接的に接続されているものとしてリストされることに注意してください。これは、MPLS がレイヤ 2 トンネル メカニズムを提供しているためです。

転送の確認とトラブルシューティング

ユーザがトラブルシューティングを実行できるように、転送の異常を検出して、理解しておく必要があります。**ping ipv6** および **traceroute ipv6** などのコマンドを使用して、データプレーン接続を検証し、トラフィックのブラックホール化を検出します。**traceroute mpls** および **show mpls forwarding** などのコマンドでは、障害の発生しているノード、インターフェイス、および Forwarding Error Correction（FEC; 転送エラー訂正）を特定できます。エッジでの特定の IPv6 宛先の転送障害のトラブルシューティングでは、一般的に、再帰的解決が基本構成要素に分割されます。この作業は、IPv6 ルーティング（iBGP または eBGP）、IP ルーティング（IS-IS または OSPF）、ラベル配布（BGP、LDP、または RSVP）、および解決の中断を検出する隣接解決の分析を組み合せて実行する必要があります。

次の例では、IPv6 VPN を確認して、さまざまな IPv6 VPN 転送状況をトラブルシューティングする方法を示します。

- 「PE-CE 接続」（P.48）
- 「PE インポジション パス」（P.50）
- 「PE ディスポジション パス」（P.51）
- 「ラベル スイッチ パス」（P.51）

PE-CE 接続

ipv6 ping および **traceroute** コマンドは、ローカルに接続されている場合でも、MPLS バックボーンを介してリモートで接続されている場合でも、PE から CE への接続を確認するのに役立ちます。

ルータがローカルに接続されている場合は、次の例に示すように、CE のリンクローカル アドレス（eBGP ピアリングに使用される）を指定して、**ipv6 ping** コマンドを使用できます。

```
Router# ping FE80::4F6B:44%Serial1/0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FE80::4F6B:44, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/33/48 ms
```

また、**ipv6 ping** コマンドを使用すると、リモート PE または CE の到達可能性もテストできますが、使用できるのは IPv6 グローバル アドレスだけです（リンクローカル アドレスはリンクの向こう側にはアドバタイズされません）。

```
Router# ping 2001:DB8:1120:1::44

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:1120:1:44::1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/33/48 ms
```

MPLS を介した **ping ipv6** および **traceroute** コマンド機能は、PE および CE に対して 1 つの IPv6 グローバル プレフィクスをアナウンスするように要求することに注意してください。各 6PE ルータは、自律システムのエッジでフィルタリングされる 2001:DB8::PE#/128 をアナウンスします。各 IPv6 CE は 2001:DB8:*prefix*:CE#/128 を設定し、これを特定的でないプレフィクスの一部としてアナウンスします（2001:DB8:*prefix*::/n）。

リモート PE および CE の到達可能性は、**traceroute** コマンドを使用してテストできます。すべての PE を **no mpls ip propagate-ttl forwarded** コマンドで設定してある場合、**traceroute** コマンドを CE から実行すると、その出力には IPv6 ノードだけが表示されます。

```
Router# traceroute 2001:DB8::1

Type escape sequence to abort.
Tracing the route to 2001:DB8::1
  1 2001:DB8::26 [AS 33751] 32 msec 32 msec 20 msec
  2 2001:DB8::1 [AS 33751] [MPLS: Label 73 Exp 0] 20 msec 20 msec 20 msec
  3 2001:DB8::1 [AS 33751] 28 msec 20 msec 20 msec
```

P ルータが ICMPv6 をサポートしているイメージでアップグレードされたあとに PE ルータで **traceroute** コマンドを実行すると（このとき、Time to Live（TTL; 存続可能時間）が伝播される）、次の例に示すように、P ルータの応答も表示されます。

```
Router# traceroute 2001:DB8::1

Type escape sequence to abort.
Tracing the route to 2001:DB8::1
  1 ::FFFF:172.20.25.1 [MPLS: Labels 38/73 Exp 0] 40 msec 32 msec 32 msec
  2 ::FFFF:172.20.10.1 [MPLS: Labels 30/73 Exp 0] 60 msec 32 msec 32 msec
  3 2001:DB8::1 [MPLS: Label 73 Exp 0] 32 msec 32 msec 16 msec
```

ping ipv6 コマンドと **traceroute** コマンドを 6VPE ルータから実行すると、どちらのコマンドも VPNv4 の場合とまったく同様に *vrf* 引数を受け付けます。

traceroute コマンドは MPLS バックボーン全体のパスの評価には役立ちますが、データプレーン障害のトラブルシューティングには役立たないことに注意してください。P ルータは IPv6 対応ではないため（VPNv4 対応でもない）、**traceroute** コマンドに応答して生成された ICMPv6 メッセージは、受信されたラベル スタックを使用して出力 PE に転送されます。出力 PE は ICMPv6 メッセージを traceroute の送信元にルーティングできます。MPLS パスが切断されている場合は ICMP メッセージからも切断されるため、ICMP メッセージは出力 PE に到達できません。

PE インポジション パス

Cisco ルータで、IPv6 のインポジション パスのトラブルシューティングに最も役立つツールは、**show ipv6 cef** コマンドです。

IPv6 転送テーブルのダンプ

次の例に示すように、**show ipv6 cef** コマンドを使用すると、各宛先プレフィクスに使用される転送テーブルとラベル スタックが表示されます。

```
Router# show ipv6 cef

2001:DB8:100::/48
  nexthop 172.20.25.1 Serial0/0 label 38 72
2001:DB8::1/128
  nexthop 172.20.25.1 Serial0/0 label 38 73
2001:DB8::26/128
  attached to Loopback0, receive
```

転送テーブル内の IPv6 エントリの詳細

次の例に示すように、**show ipv6 cef** コマンドを使用すると、特定のエントリの詳細を表示したり、宛先の解決方法やラベル スタックの計算方法を分析したりできます。

```
Router# show ipv6 cef 2001:DB8:100::/48 internal

2001:DB8:100::/48, epoch 0, RIB[B], refcount 4
  sources: RIB
..
  recursive via 192.168.2.101[IPv4:Default] label 72, fib 0252B1F8, 1 terminal fib
    path 024F56A8, path list 024F0BA8, share 0/1, type attached nexthop
    ifnums: (none)
     path_list contains at least one resolved destination(s). HW IPv4 notified.
    nexthop 172.20.25.1 Serial0/0 label 38, adjacency IP adj out of Serial0/0 0289BEF0
  output chain: label 72 label 38 TAG adj out of Serial0/0 0289BD80
```

BGP テーブル内の BGP エントリの詳細

前述の例の詳細出力には、ラベル スタックを構成している各ラベルに、個別に追跡できる発信元がそれぞれ含まれていることが示されています。次の例に示すように、BGP テーブルには一番下のラベルが格納されています。

```
Router# show bgp ipv6 unicast 2001:DB8:100::/48

BGP routing table entry for 2001:DB8:100::/48, version 2
Paths: (2 available, best #2, table default)
  Advertised to update-groups:
     1
  10000
    ::FFFF:192.168.2.101 (metric 30) from 192.168.2.147 (192.168.2.147)
      Origin incomplete, metric 0, localpref 100, valid, internal
      Originator: 192.168.2.101, Cluster list: 192.168.2.147,
      mpls labels in/out nolabel/72
  10000
    ::FFFF:192.168.2.101 (metric 30) from 192.168.2.146 (192.168.2.146)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Originator: 192.168.2.101, Cluster list: 192.168.2.146,
      mpls labels in/out nolabel/72
```

次の例に示すように、LDP ではその他のラベルが表示されます。

```
Router# show mpls ldp bindings 192.168.2.101 32

  lib entry: 192.168.2.101/32, rev 56
```

```
        local binding:  label: 40
        remote binding: lsr: 192.168.2.119:0, label: 38

Router# show mpls ldp bindings 172.20.25.0 24

  lib entry: 172.20.25.0/24, rev 2
        local binding: label: imp-null
        remote binding: lsr: 192.168.2.119:0, label: imp-null
```

PE ディスポジション パス

次の例を使用して、ディスポジション パスをトラブルシューティングします。

- 「MPLS 転送テーブルのダンプ」（P.51）
- 「BGP ラベル分析」（P.51）

MPLS 転送テーブルのダンプ

次に、ディスポジション パスをトラブルシューティングするための MPLS 転送テーブル情報の例を示します。

```
Router# show mpls forwarding-table

Local  Outgoing      Prefix            Bytes Label   Outgoing   Next Hop
Label  Label or VC   or Tunnel Id      Switched      interface
16     Pop Label     192.168.2.114/32    0             Se0/0      point2point
17     26            192.168.2.146/32    0             Se0/0      point2point
..
72     No Label      2001:DB8:100::/48     63121         Se1/0      point2point
73     Aggregate     2001:DB8::1/128   24123
```

BGP ラベル分析

次に、スイッチングに使用されるラベルの例を示します。ラベルは iBGP（この例では 6PE）によってアナウンスされており、確認が可能です。

```
Router# show bgp ipv6 2001:DB8:100::/48

BGP routing table entry for 2001:DB8:100::/48, version 2
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
     2
  10000
    FE80::2710:2 (FE80::2710:2) from FE80::2710:2%Serial1/0 (192.168.2.103)
      Origin incomplete, metric 0, localpref 100, valid, external, best,
```

ラベル スイッチ パス

6PE および 6VPE LSP エンドポイントは IPv4 アドレスであるため、LSP をトラブルシューティングする IPv4 ツールが、IPv6 トラフィックのブラックホール化につながるデータプレーン障害の検出に役立ちます。

ラベル スイッチ パスの分析

次に、LSP IPv4 エンドを表示する例を示します。

```
Router# show ipv6 route 2001:DB8::1/128

Routing entry for 2001:DB8::1/128
  Known via "bgp 33751", distance 200, metric 0, type internal
  Route count is 1/1, share count 0
  Routing paths:
    192.168.2.101%Default-IP-Routing-Table indirectly connected
```

```
      MPLS Required
      Last updated 02:42:12 ago
```

traceroute LSP の例

次に、traceroute LSP の例を示します。

```
Router# traceroute mpls ipv4 192.168.2.101/32 verbose

Tracing MPLS Label Switched Path to 192.168.2.101/32, timeout is 2 seconds
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target,
       'M' - malformed request
Type escape sequence to abort.
  0 172.20.25.2 0.0.0.0 MRU 1500 [Labels: 38 Exp: 0]
R 1 172.20.25.1 0.0.0.0 MRU 1500 [Labels: 30 Exp: 0] 40 ms, ret code 6
R 2 172.20.10.1 0.0.0.0 MRU 1504 [Labels: implicit-null Exp: 0] 60 ms, ret code 6
! 3 172.20.40.1 48 ms
```

VRF 情報

次のコマンド入力では、6VPE の VRF 情報が表示されます。

show ipv6 cef vrf

次に、cisco1 という名前の VRF に関連付けられているシスコ エクスプレス フォワーディング FIB からの出力例を示します。

```
Router# show ipv6 cef vrf cisco1

 2001:8::/64
  attached to FastEthernet0/0
 2001:8::3/128
  receive
 2002:8::/64
  nexthop 10.1.1.2 POS4/0 label 22 19
 2010::/64
  nexthop 2001:8::1 FastEthernet0/0
 2012::/64
  attached to Loopback1
 2012::1/128
  receive
```

show ipv6 route vrf

次に、cisco1 という名前の VRF に関連付けられている IPv6 ルーティング テーブルに関する出力例を示します。

```
Router# show ipv6 route vrf cisco1

IPv6 Routing Table cisco1 - 6 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
C   2001:8::/64 [0/0]
     via ::, FastEthernet0/0
L   2001:8::3/128 [0/0]
     via ::, FastEthernet0/0
B   2002:8::/64 [200/0]
     via ::FFFF:192.168.1.4,
B   2010::/64 [20/1]
     via 2001:8::1,
```

```
C   2012::/64 [0/0]
     via ::, Loopback1
L   2012::1/128 [0/0]
     via ::, Loopback1
```

ルーティングおよび転送のデバッグ

ルーティングおよび転送の異常をトラブルシューティングする場合、デバッグ コマンドをイネーブルにすると役立つ可能性がありますが、いくつかのデバッグ メッセージはルータの動作を遅くし、このようなツールの有用性を損なう可能性があります。そのため、**debug** コマンドは注意して使用する必要があります。**debug ipv6 cef**、**debug mpls packet**、および **debug ipv6 packet** コマンドは、転送パスのトラブルシューティングに役立ち、**debug bgp ipv6** および **debug bgp vpnv6** コマンドはコントロール プレーンのトラブルシューティングに役立ちます。

IPv6 VPN over MPLS を実装するための設定例

例：IPv4 ネクストホップを使用した IPv6 VPN の設定

次に、6VPE ネクストホップの例を示します。

```
interface Loopback0
 ip address 192.168.2.11 255.255.255.255
!
router bgp 100
 neighbor 192.168.2.10 remote-as 100
 neighbor 192.168.2.10 update-source Loopback0
!
 address-family vpnv6
 neighbor 192.168.2.10 activate
 neighbor 192.168.2.10 send-community extended
 exit-address-family
```

デフォルトでは、アドバタイズされるネクストホップは IPv6 VPN アドレスになります。

```
[0:0]::FFFF:192.168.2.10
```

[*RD*]::FFFF:*IPv4-address* の形式の 192 ビット アドレスであることに注意してください。

BGP IPv6 VPN ピアが共通サブネットを共有する場合、MP_REACH_NLRI アトリビュートには、グローバル アドレス ネクストホップに加えてリンクローカル アドレス ネクストホップも含まれます。この状況は、一般的に、ASBR が互いに向き合っている相互自律システム トポロジで発生します。この場合、リンクローカル ネクストホップがローカルに使用され、グローバル ネクストホップは BGP によって再アドバタイズされます。

BGP ネクストホップは、ラベル スタックを構築する場合の中心要素です。内部ラベルは BGP NLRI から取得され、外部ラベルは BGP ネクストホップに埋め込まれた IPv4 アドレスに到達する Label Distribution Protocol（LDP; ラベル配布プロトコル）ラベルになります。

その他の関連資料

関連資料

関連項目	参照先
IPv6 マルチプロトコル BGP	『*Implementing Multiprotocol BGP for IPv6*』
IPv6 EIGRP	『*Implementing EIGRP for IPv6*』
IPv6 MPLS	『*Implementing IPv6 over MPLS*』
IPv6 スタティック ルート	『*Implementing Static Routes for IPv6*』
IPv6 コマンド	『*Cisco IOS IPv6 Command Reference*』

規格

規格	タイトル
draft-bonica-internet-icmp	『*ICMP Extensions for Multiprotocol Label Switching*』
draft-ietf-idr-bgp-ext-communities-0*x*.txt	『*Cooperative Route Filtering Capability for BGP-4*』

MIB

MIB	MIB リンク
•	選択したプラットフォーム、Cisco ソフトウェア リリース、および機能セットの MIB の場所を検索しダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

RFC

RFC	タイトル
RFC 1267	『*A Border Gateway Protocol 3 (BGP-3)*』
RFC 1772	『*Application of the Border Gateway Protocol in the Internet*』
RFC 1918	『*Address Allocation for Private Internets*』
RFC 2858	『*Multiprotocol Extensions for BGP-4*』
RFC 3107	『*Carrying Label Information in BGP-4*』
RFC 3392	『*Capabilities Advertisement with BGP-4*』
RFC 3513	『*Internet Protocol Version 6 (IPv6) Addressing Architecture*』
RFC 4007	『*IPv6 Scoped Address Architecture*』
RFC 4193	『*Unique Local IPv6 Unicast Addresses*』
RFC 4364	『*BGP MPLS/IP Virtual Private Networks (VPNs)*』

RFC	タイトル
RFC 4382	『*MPLS/BGP Layer 3 Virtual Private Network (VPN) Management Information Base*』
RFC 4659	『*BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN*』

シスコのテクニカル サポート

説明	リンク
右の URL にアクセスして、シスコのテクニカル サポートを最大限に活用してください。 以下を含むさまざまな作業にこの Web サイトが役立ちます。 ・テクニカル サポートを受ける ・ソフトウェアをダウンロードする ・セキュリティの脆弱性を報告する、またはシスコ製品のセキュリティ問題に対する支援を受ける ・ツールおよびリソースへアクセスする - Product Alert の受信登録 - Field Notice の受信登録 - Bug Toolkit を使用した既知の問題の検索 ・Networking Professionals（NetPro）コミュニティで、技術関連のディスカッションに参加する ・トレーニング リソースへアクセスする ・TAC Case Collection ツールを使用して、ハードウェアや設定、パフォーマンスに関する一般的な問題をインタラクティブに特定および解決する この Web サイト上のツールにアクセスする際は、Cisco.com のログイン ID およびパスワードが必要です。	http://www.cisco.com/en/US/support/index.html

IPv6 VPN over MPLS の実装の機能情報

表 1 に、このモジュールで説明した機能をリストし、特定の設定情報へのリンクを示します。

プラットフォームのサポートおよびソフトウェア イメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator により、どのソフトウェア イメージが特定のソフトウェア リリース、機能セット、またはプラットフォームをサポートするか調べることができます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn からアクセスします。Cisco.com のアカウントは必要ありません。

（注） 表 1 には、一連のソフトウェア リリースのうち、特定の機能が初めて導入されたソフトウェア リリースだけが記載されています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

表 1　IPv6 VPN over MPLS の実装の機能情報

機能名	リリース	機能情報
IPv6 VPN over MPLS（6VPE）	12.2(28)SB 12.2(33)SRB 12.2(33)SXI 12.4(20)T 15.0(1)S	MPLS を介した IPv6 VPN（6VPE）の IPv4 コア インフラストラクチャ機能を使用すると、ISP はカスタマーに IPv6 VPN サービスを提供できます。 このマニュアルでは、この機能について説明しています。
IP トンネルを介した MPLS VPN 6VPE サポート	12.2(33)SRB1 12.2(33)SXI	この機能では、IPv4 GRE トンネルを使用して、BGP ネクストホップに到達するための IPv6 VPN over MPLS 機能を提供できます。 この機能に関する詳細については、次の各項を参照してください。 • 「MPLS 転送」（P.5）

用語集

- **6VPE ルータ**：IPv4 ベースの MPLS コア上に BGP-MPLS IPv6 VPN サービスを提供するプロバイダー エッジ ルータ。コア方向のインターフェイスで 6PE 概念を実装する IPv6 VPN PE のデュアル スタック ルータです。
- **Customer Edge（CE; カスタマー エッジ）ルータ**：VPN カスタマー サイトに接続するサービス プロバイダー ルータ。
- **Forwarding Information Base（FIB; 転送情報ベース）**：IP データグラムの転送に必要な情報を含むテーブル。FIB には、少なくとも、到達可能な宛先ネットワーク プレフィクスごとにインターフェイス識別子とネクストホップ情報が格納されます。
- **Inbound Route Filtering（IRF; インバウンド ルート フィルタリング）**：受信側 PE ルータによってインポートされない着信 BGP アップデートをフィルタリングするために使用される BGP 機能。
- **IPv6 Provider Edge（6PE; IPv6 プロバイダー エッジ）ルータ**：BGP ベースのメカニズムを実行して、MPLS 対応の IPv4 クラウド上で IPv6 アイランドを相互接続するルータ。
- **IPv6 VPN アドレス**：IPv6 VPN アドレスは、8 バイトの Route Distinguisher（RD; ルート識別子）で始まり、16 バイトの IPv6 アドレスで終わる、24 バイトの識別子です。IPv6 VPN アドレスと呼ばれることもあります。
- **IPv6 VPN アドレス ファミリ**：Address-Family Identifier（AFI; アドレス ファミリ識別子）は特定のネットワーク レイヤ プロトコルを識別し、Subsequent AFI（SAFI; 後続 AFI）は追加情報を提供します。AFI IPv6 SAFI VPN（AFI=2、SAFI=128）は、IPv6 VPN アドレス ファミリと呼ばれます。IPv6 VPN アドレス ファミリと呼ばれることもあります。同様に、AFI IPv4 SAFI VPN は VPNv4 アドレス ファミリと呼ばれます。
- **Network Layer Reachability Information（NLRI; ネットワーク レイヤ到達可能性情報）**：BGP では、ルートおよびそのルートへのアクセス方法を記述した NLRI を含むルーティング アップデート メッセージが送信されます。この場合、NLRI がプレフィクスとなります。BGP アップデート メッセージでは、1 つ以上の NLRI プレフィクス、および NLRI プレフィクスのルートのアトリビュートが伝送されます。ルート アトリビュートには、BGP ネクストホップ ゲートウェイ アドレスおよびコミュニティ値が含まれています。
- **Outbound Route Filtering（ORF; アウトバウンド ルート フィルタリング）**：発信 BGP ルーティング アップデートのフィルタリングに使用される BGP 機能。
- **Point of Presence（POP）**：装置にインストールされている中継キャリアが、地域通信事業者と相互接続する物理的なロケーション。
- **Provider Edge（PE; プロバイダー エッジ）ルータ**：VPN カスタマー サイトに接続されるサービス プロバイダー ルータ。
- **Route Distinguisher（RD; ルート識別子）**：グローバルに一意な IPv6 VPN アドレスを形成するために、IPv6 プレフィクスの先頭に付加される 64 ビットの値。
- **Routing Information Base（RIB; ルーティング情報ベース）**：ルーティング テーブルとも呼ばれます。
- **Virtual routing and forwarding（VRF; 仮想ルーティングおよび転送）**：PE 内の VPN ルーティングおよび転送インスタンス。
- **VRF テーブル**：VRF に関連付けられているルーティングおよび転送テーブル。これは、PE ルータがカスタマーごとに独立したルーティング状態を維持できるようにするカスタマー固有のテーブルです。