ジュニパーネットワークス EVPN 次世代データ センター アーキテクチャの実装

イーサネット VPN により、進化するデータ センター要件に対処する

目次

図一覧

概要

これまでのデータ センターでは、Spanning Tree Protocol（STP）、マルチシャーシ リンク アグリゲーション グループ（MC-LAG）、Transparent Interconnection of Lots of Links（TRILL）などのレイヤー 2 技術をコンピューティングとストレージの接続に使用してきました。データ センターの設計が進化してスケール アウト マルチテナント ネットワークになると、Virtual Extensible LAN（VXLAN）などの技術を使用して、アンダーレイ ネットワークをテナントのオーバーレイ ネットワークから切り離す、新しいデータ センター アーキテクチャが求められるようになりました。レイヤー 3 IP ベースのアンダーレイと VXLAN-EVPN オーバーレイを使用すると、データ センターとクラウドのオペレーターは、従来の L2 イーサネット ベース アーキテクチャで可能だったものよりもはるかに大規模なネットワークを導入できます。オーバーレイを使用すると、エンドポイント（サーバーや仮想マシン）をネットワークのどこにでも配置して、同じ論理 L2 ネットワークとの接続を維持できるため、仮想トポロジーを物理トポロジーから切り離すことができます。

はじめに

現在のデータ センター ネットワークには、さまざまなトレンド[1] によるプレッシャーがかかっています。

- 企業の IT 戦略におけるクラウド ベースのリソースとサービスの重要性が高まり、セキュリティーや性能で妥協しない高性能ネットワーク アーキテクチャが求められています。
- エンド ユーザーは時間と場所を問わずにアクセスできることや高レベルな応答性を必要としていますが、今日のネットワーク アーキテクチャでは、その達成がますます困難になっています。

このようなトレンドから、データ センター アーキテクチャでは、以下の 3 つの主要な目標を念頭に置いたネットワークのビジョン変更が推進されています。

- **拡張性**：企業のなかにはクラウド サービスの使用を増やして成長に対応しているところもあれば、独自のプライベート クラウドとハイブリッド クラウドを導入しているところもあります。サービス プロバイダは、需要を満たす十分な容量を確保するため、急速な拡張を求められます。多くの場合、今日のネットワークは、柔軟性が低く、変更しづらいため、大企業やサービス プロバイダの拡張ニーズに対応できません。クラウド データ センターに求められているのは、新しいテナント拡張方法です。そのような方法の 1 例が VXLAN です。VXLAN は、基盤となるネットワークをアンダーレイ ネットワーク上でトンネリングして、テナントの状態を基盤となるネットワークの状態から切り離すことで、クラウド データ センターのテナント数を 1,600 万まで拡張します。
- **効率的な運用**：地理的範囲を拡大した企業は、データ センターとユーザー間の物理的な距離、および 24 時間運用による保守の時間帯短縮に関する問題に直面します。新しいデータ センター ネットワークは、アプリケーションの移動をサポートし、ネットワーク管理者がデータ センター内とデータ センター間のアプリケーションを簡単に移行できるようにして、ビジネス継続性、ダウンタイムの発生しない保守、負荷分散を実現する必要があります。
- **高性能**：エンド ユーザーは応答時間の遅さに対してよく苦情を言いますが、帯域幅の制限や遅延の問題によって引き起こされるビジネスクリティカルなアプリケーションの停止すら苦情の対象となることがあります。よって、新しいデータ センターにはマルチパスのような技術と、コントロール プレーンの学習機能により、ネットワーク トラフィック フローを最適化し、ネットワークの障害を阻止し、帯域幅の利用率を最大化できるような仕組みが求められます。

今日のネットワークの主な問題は、アプリケーションが物理ネットワークのトポロジーと結び付いていることです。これが、次のような望ましくない結果を引き起こします。

- ネットワークを拡張できないために、アプリケーションを拡張できない。
- データセンター内または他のデータセンターとの間でアプリケーションを簡単に移動できない。
- アプリケーションと物理インフラ間の接続に柔軟性がないため、クラウド サービスを活用できない。

VXLAN と EVPN によるデータ センター ネットワークの変革

従来のデータ センターでは、ネットワーク設計者は、ユーザーとアプリケーションを分離してセキュリティーを担保するために、L2 の論理ネットワークを構成する手法として VLAN を使用してきました。VLAN を使用し、またネットワークの性能を高めるためにブロードキャスト通信の制限を行ってきました。

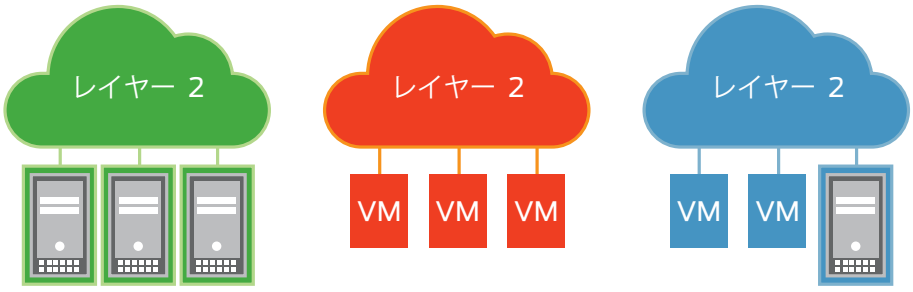


図 1：レイヤー 2 論理ネットワーク

[1] http://www.datacenterknowledge.com/archives/2014/12/22/dynamic-data-center-3-trends-driving-change-2015/

ただし、このアーキテクチャでは拡張が難しくなります。VLAN の仕様（IEEE 802.1ad）で提供されるアドレス空間は比較的小さいため、使用できる VLAN の最大数は 4,096 になります。VLAN と論理ネットワークの間には 1 対 1 のマッピングが存在するので、データ センターにおける論理ネットワークの数も 4,096 までに制限されます。通常、マルチテナント環境では多くのユーザー数をサポートし、それぞれに複数の論理ネットワークが必要な場合があります。したがって、比較的容易にこの制限に到達してしまいます。

VLAN のアプローチには、VLAN をホストする物理ハードウェア環境にしか仮想マシンを移動できないという別の問題があります（このため、VM に関連付けられたアプリケーションも移動できません）。同じデータ センター内や別のデータ センターへのアプリケーションの移動は、煩雑であり、間違いが起こりがちです。実際には、ほとんどのネットワーク管理者は絶対に必要な場合以外はこの作業を回避します。

VXLAN の概要

VXLAN（規格 IETF RFC7348）は、これらの問題解決に大きく貢献しました。VXLAN では、ネットワーク管理者が異なる L3 ネットワーク間に論理 L2 ネットワークを構築できます。VXLAN には 24 ビットの仮想ネットワーク ID（VNID）領域があり、1,600 万の論理ネットワークが構築可能です。VXLAN はハードウェアにて実装され、トンネルのカプセル化内のネイティブ イーサネット パケット転送をサポートします。VXLAN は、物理スイッチで終端するオーバーレイのデ ファクト スタンダードになっており、Juniper Networks® QFX5100 スイッチと QFX10000 スイッチ、EX9200 イーサネット スイッチ、MX シリーズ 3D ユニバーサル エッジ ルーターが対応しています。

VXLAN オーバーレイには次のような多くのメリットがあります。

- Spanning Tree Protocol（STP）の除去
- 拡張性の向上
- 耐障害性の向上
- 障害の封じ込め

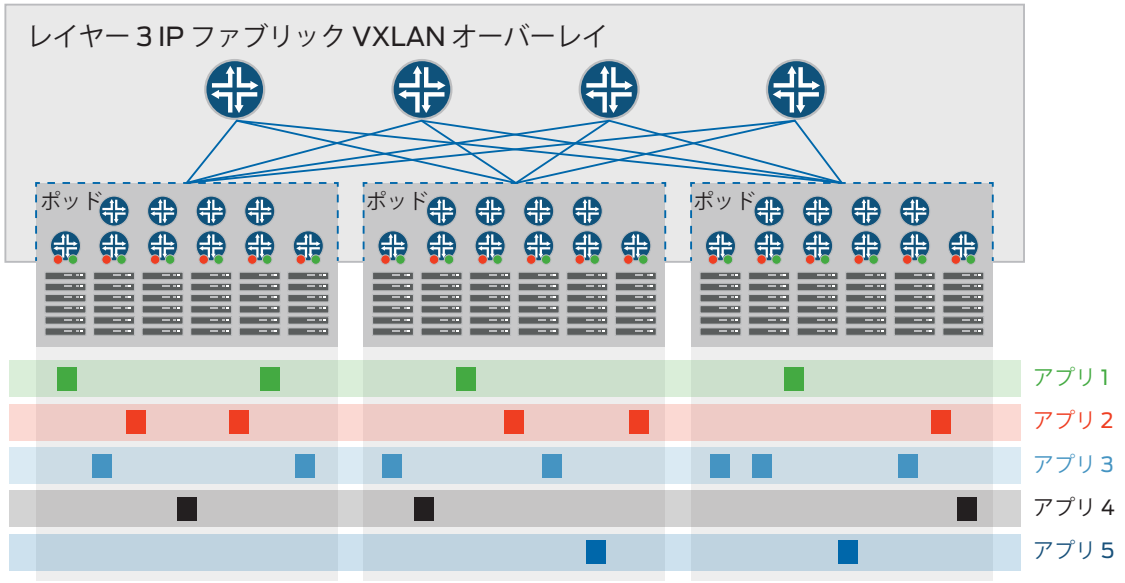


図 2：性能を最適化されたデータ センター（ポッド）間のアプリケーション移動

データ センター コントロール プレーン

VXLAN による L2 ネットワークの抽象化をもってしても、イーサネットプロトコルにおける「フラッディングとラーニング」[2] の挙動は変わることは無く、これは拡張性、効率性、利用性の点で制限事項として残存してしまいます。

VXLAN は、コントロール プレーン プロトコルを使用せずに、L3 Clos データ センター全体にトンネリング プロトコルとして導入することができます。主な方法は、マルチキャスト対応アンダーレイを使用した VXLAN、および静的ユニキャスト VXLAN トンネルという 2 つです。どちらもアンダーレイにおけるレガシーな L2 構成の 除去に実行可能なオプションですが、レイヤー 2 プロトコルを使用した場合、どちらもフラッドアンドラーン型の問題は解決せず、大規模なマルチテナント環境には拡張できません。

フラッディングを最小限に抑え、学習を容易にするためのソリューションは、コントロール プレーンの導入です。学習を容易にするために、コントロール プレーンはエンド ホストの情報を同じセグメント内の仮想トンネル エンド ポイント（VTEP）に配信します。

[2] スイッチが、宛先 MAC アドレスやポートをもたないブロードキャスト フレーム、マルチキャスト フレーム、ユニキャスト フレームのいずれかを受信すると、入力ポート以外の全ポートからのフレームを「フラッド」します。スイッチは、フレームの送信元ポートおよび MAC アドレスをスイッチの MAC アドレス テーブルに追加して、受信フレームを「学習」します。

マルチプロトコル BGP（MP-BGP）は、フラッディングとラーニングの問題に対処することが出来るプロトコルです。MP-BGP を使用すると、L2 メディア アクセス制御（MAC）と L3 IP 情報を同時にネットワークで送信できます。MAC と IP 情報を組み合わせて転送上の決定に使用できるため、最適なルーティングとスイッチングを提供することが可能です。BGP による L2 MAC と L3 IP 情報の転送を可能にするこのような拡張をイーサネット VPN（EVPN）と呼びます。

EVPN はイーサネットプロトコルにおけるフラッディングとラーニングの問題を解決します。IP ファブリック用のオーバーレイ プロトコルの選択肢として VXLAN が出現したことは、従来の MPLS 転送要件にとらわれずに、EVPN で VXLAN を転送に使用できることを意味します。そのうえ、EVPN は標準化技術 [3] をベースにしているためさまざまな Software Defined Networking（SDN）コントローラとの連動やソリューションの一翼を担う技術としても利用することが可能です。

EVPN の概要

EVPN、バーチャル プライベート LAN サービス（VPLS）、または L2VPN のような、コントロールベースのプロトコルはレガシーなフラッディングとラーニングの問題に対処を行います。ただし、これらの大部分は MPLS によって推進されていました。IP ファブリック用オーバーレイ プロトコルの選択肢として VXLAN が出現した場合、VXLAN を転送に使用することで、EVPN は従来の MPLS 転送要件の制限を受けなくなります。本資料の次のセクションでは、データ センター導入における EVPN のメリット、MPLS ベース EVPN との違い、導入時の考慮事項について詳細に説明します。

EVPN には次のようなメリットがあります。

ネットワーク効率の向上

- コントロール プレーン MAC 学習によって未知のユニキャスト フラッディングを削減
- コントロール プレーン内の MAC と IP のバインディングによるアドレス解決プロトコル（ARP）フラッディングを削減
- 複数のスパイン スイッチ上のマルチパス トラフィック（VXLAN エントロピー）
- アクティブ / アクティブ デュアルホームド サーバーへのマルチパス トラフィック
- 分散型 L3 ゲートウェイ：仮想マシン トラフィック最適化（VMTO）

迅速なコンバージェンス

- デュアルホームド サーバーとのリンクに障害が発生した場合の迅速な再コンバージェンス（エイリアシング）
- VM 移動時の迅速な再コンバージェンス

拡張性

- 拡張性に優れた BGP ベースのコントロール プレーン

柔軟性

- L3VPN と L2VPN を簡単に統合して、データ センター相互接続（DCI）を実現
- きめ細かなポリシーを適用可能な BGP ベース コントロール プレーン

EVPN は、データ センター コントロール プレーン プロトコルのメリットを提供する、完全に標準に準拠した唯一のソリューションです。

[3] 関連する EVPN 標準には、RFC 4364『BGP/MPLS IP Virtual Private Networks (VPNs)』、RFC 4761『Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling』、RFC 7432『BGP MPLS-Based Ethernet VPN』などがあります。

EVPN のコンセプト

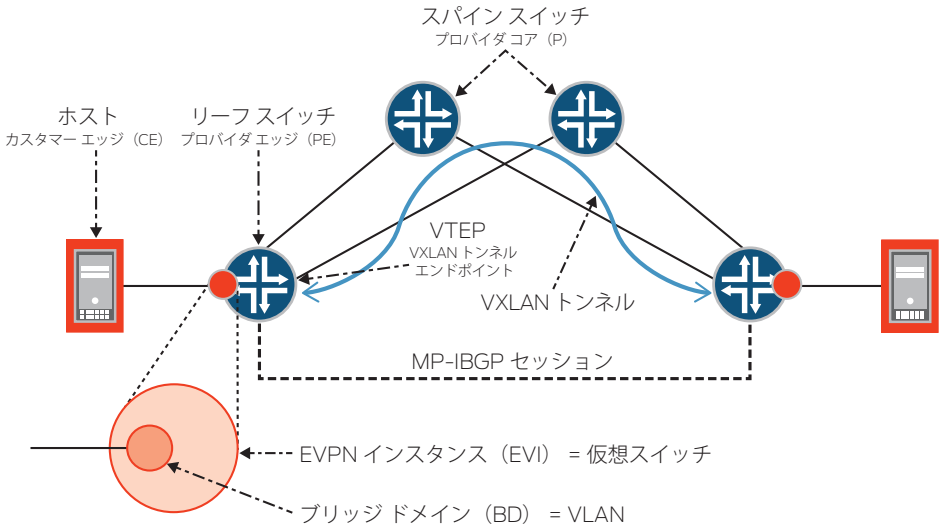


図 3：EVPN の用語

図 3 は、L3 Clos トポロジーのリーフ スイッチ 2 台（「トップオブラック」）を示しています。これら 2 台のデバイスの間には、N 台の IP 転送スイッチ / ルーター（「プロバイダ コア」デバイス）があります。

EVI = EVPN インスタンス。特定の EVPN に参加しているプロバイダ エッジ（PE）デバイス内の仮想スイッチインスタンス。

MAC-VRF：PE デバイス上の MAC アドレス用の仮想ルーティングおよび転送テーブル。MAC-VRF ごとに固有なルート識別子（RD）が定義されています。

ES = イーサネット セグメント。各イーサネット セグメントには、EVPN 内で一意の識別子が必要です。顧客サイトがイーサネット リンクのセットを介して 1 台以上の PE デバイスに接続されている場合、このイーサネット リンクのセットは 1 つの ES で構成されています。

ESI = イーサネット セグメント識別子。マルチホームド サイトの場合、各 ES は一意のゼロ以外な識別子で識別されます。この識別子をイーサネット セグメント識別子（ESI）と呼びます。一般的に、イーサネット セグメントには、ネットワーク内（つまり、すべての PE デバイス上のすべての EVPN インスタンス）で一意の予約済みでない ESI が必要です。

リーフ スイッチ（「プロバイダ エッジ」）は、ホスト（サーバー、ストレージ、ベアメタル デバイスなど）を収容しています。このホストをカスタマー エッジ（CE）デバイスと呼びます。

リーフ デバイス間に MP-BGP セッションを確立します。EVPN ではこれを利用してオーバーレイ制御プロトコルで使用されるルートを配信します。

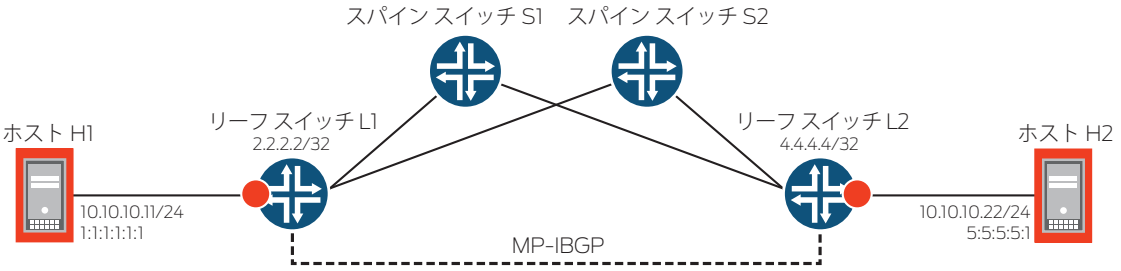
EVPN にはルート タイプというコンセプトが導入されています。本資料の公開時点では、ルート タイプは 5 つあります。

- ルート タイプ 1：イーサネット自動検知（AD）ルート
  - EVI 単位および ESI 単位でアドバタイズされます。イーサネット自動検知ルートは、CE デバイスがマルチホーミングの場合に必要です。CE デバイスが単一ホーミングの場合、ESI はゼロになります。
- ルート タイプ 2：MAC/IP アドバタイズメント ルート
  - EVPN では、EVPN ネットワーク レイヤー到達性情報（NLRI）内でエンド ホストの IP と MAC アドレスをアドバタイズできます。これにより、コントロール プレーンはエンド システムの MAC アドレスを学習できます。
- ルート タイプ 3：インクルーシブ マルチキャスト イーサネット タグ ルート
  - このルートでは、ブロードキャスト、未知のユニキャスト、マルチキャスト（BUM）トラフィックのパスを、PE デバイスからリモートの PD デバイスまで、VLAN 単位および ESI 単位で設定します。
- ルート タイプ 4：イーサネット セグメント ルート
  - ESI により、2 台以上の PE デバイスに対して、CE デバイスはシングル / アクティブ モードまたはアクティブ / アクティブ モードでマルチホーミングできます。同じイーサネット セグメントに接続している PE デバイスは、ES ルートを介して相互に検知します。
- ルート タイプ 5：IP プレフィックス ルート
  - （オプション）サブネット間の転送を行うための IP プレフィックスルートを伝播します。

EVPN リモート MAC 学習

EVPN MP-BGP セッションが 2 台のデバイス間で確立されると、EVPN コントロール プレーンはさまざまなタイプの到達性情報をアドバタイズします。

最初に EVPN タイプ 2 のルートについて説明します。



NLRI	ルート タイプ	MAC/IP アドバタイズメント ルート（タイプ 2）
	ルート識別子（RD）	リーフ スイッチ L2 上の赤い EVI の RD
	イーサネット セグメント識別子（ESI）	0（シングル ホームド ホスト）
	イーサネット タグ ID	ブリッジ ドメインのグローバル VXLAN VNID
	MAC アドレス	ホスト H2 の MAC アドレス（5:5:5:5:5:5:1）
	IP アドレス	ホスト H2 の IP アドレス（10.10.10.22）**
	MPLS ラベル 1 + MPLS ラベル 2	VNID
ネクストホップ		L2 のループバック IP アドレス（4.4.4.4）
拡張コミュニティ		ルートターゲット（赤）
他の属性（発信元、AS-Path、Local-Pref など）		…

図 4：リモート学習（MAC/IP アドバタイズメント、EVPN タイプ 2 ルート）

図 4 では、リーフ スイッチ L2 が従来の L2 学習によってホスト H2 の MAC アドレスをローカルに学習します。オプションとして、L2 は動的ホスト構成プロトコル（DHCP）または ARP スヌーピングを使用して、IP と MAC のバインディングも学習できます。

従来のフラッドアンドラーン型のネットワークでは、H2 がトラフィックを H1 に送信するか、H1 が H2 から BUM トラフィック（ARP 要求）を受信するまでは、リーフ スイッチ L1 は H2 の MAC アドレスを学習しませんでした。リーフ スイッチ L1 が H2 の MAC アドレスを学習するまでは、H1 から H2 宛てのすべてのトラフィックは、同じ ES 内のネットワーク全体の全リーフ スイッチ間で未知のユニキャストとしてフラッドされます。

一方、EVPN を使用した場合、リーフ スイッチ L2 がホスト H2 の MAC アドレスをローカルに学習すると即座に、タイプ 2 ルートを介して、同じ VXLAN VNID に属するその MP-BGP ピアすべてにこの情報がアドバタイズされます。これが EVPN コントロール プレーンの大きなメリットの 1 つです。

EVPN サーバー マルチホーミング

冗長構成のトップオブラック デバイスへのサーバー マルチホーミングは、データ センターにおける一般的な要件です。従来は、この要件にはマルチシャーシ リンク アグリゲーション グループ（MC-LAG）、スタッキングやバーチャル シャーシなど、ベンダー独自のソリューションが必要でした。それぞれのソリューションにメリットがありますが、デバイスのベンダーを同じにする必要があり、MC-LAG の場合には、マルチホーミングが 2 台の PE デバイスに限定されていました。

一方、EVPN は標準に準拠したマルチホーミング ソリューションであり、任意の数の PE デバイスに水平的に拡張でき、マルチベンダーの L3 Clos ファブリックにシームレスに統合できます。

EVPN サーバーマルチホーミングには、ESI を表す新しいタイプのルートが必要です。これが EVPN タイプ 1 ルートです。

イーサネット自動検知（タイプ 1）ルート
イーサネット セグメント（ES）単位：マルチパスと迅速なコンバージェンス

NLRI	ルート タイプ	イーサネット自動検知ルート（タイプ 1）
	ルート識別子（RD）	リーフ スイッチ LS2 上の EVI の RD（LS2 の IP を含む）
	イーサネット セグメント ID（ESI）	0:1:1:1:1:1:1:1:1:1
	イーサネット タグ ID	MAX-ET
	MPLS ラベル	0
拡張コミュニティ		ESI ラベル拡張コミュニティ： • シングルアクティブ フラグ = false (0) • ESI ラベル = null
ネクストホップ		LS2 のループバック IP アドレス
他の属性（発信元、AS-Path、Local-Pref など）		

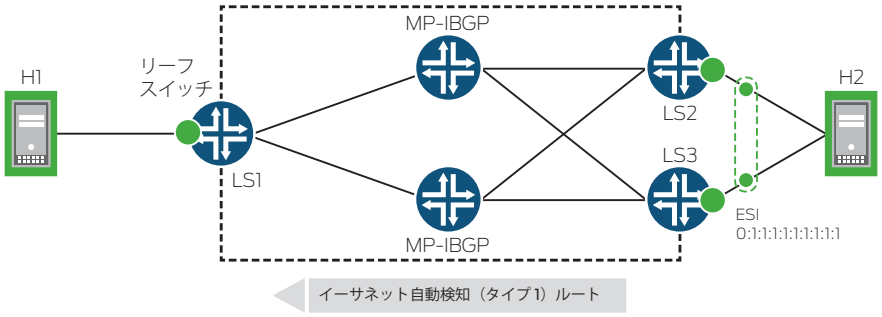


図 5：EVPN タイプ 1 アドバタイズメント、ESI

図 5 では、H2 は標準的なリンク アグリゲーション グループ（LAG）を介して同じ L2 ドメイン内の LS2 と LS3 にマルチホーミングされています。LS2 と LS3 はどちらも、LS1 へのタイプ 1 ルートを介して、この L2 セグメントへの直接的な到達性、または ESI をアドバタイズします。

タイプ 1 ルートでは、この ESI により学習された MAC アドレスはアドバタイズされません。MAC の到達性には、タイプ 2 ルートが必要です。

最もシンプルなケースとして、LS2 と LS3 がどちらも H2 の MAC アドレスを学習したと想定します。

MAC/IP アドバタイズメント（タイプ 2）ルート
再検討（マルチホームド ホストの場合）

NLRI	ルート タイプ	MAC/IP アドバタイズメント ルート（タイプ 2）
	ルート識別子（RD）	…
	イーサネット セグメント識別子（ESI）	0:1:1:1:1:1:1:1:1:1
	イーサネット タグ ID	VNID
	MAC アドレス	ホスト H2 の MAC アドレス（5:5:5:5:5:5:1）
	IP アドレス	ホスト H2 の IP アドレス（10.10.10.22）
	MPLS ラベル 1 + MPLS ラベル 2	VNID
ネクストホップ		LS3 のループバック
他の属性（発信元、AS-Path、Local-Pref など）		…

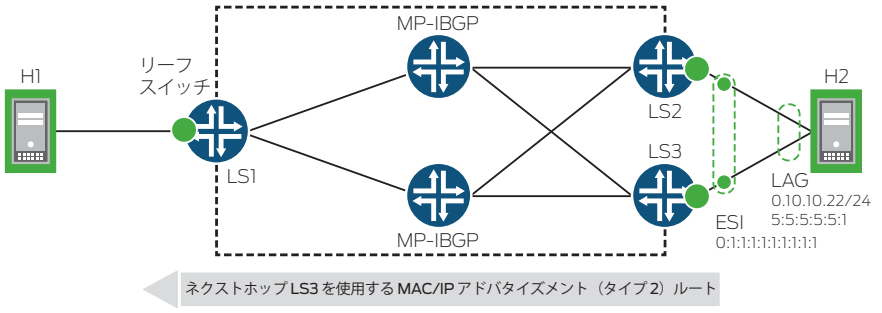


図 6：EVPN タイプ 2 アドバタイズメントと関連付けられた ESI

図 6 では、LS1 は H2 の MAC に関するタイプ 2 アドバタイズメントを LS3 から受信し、関連付けられた ESI 0:1:1:1:1:1:1:1:1:1 も受信します。同様に、LS1 は H2 のタイプ 2 アドバタイズメントを同じ ESI（非表示）の LS2 からも受信します。したがって、LS1 は H2 が両方のピアを経由して到達可能なことを認識します。

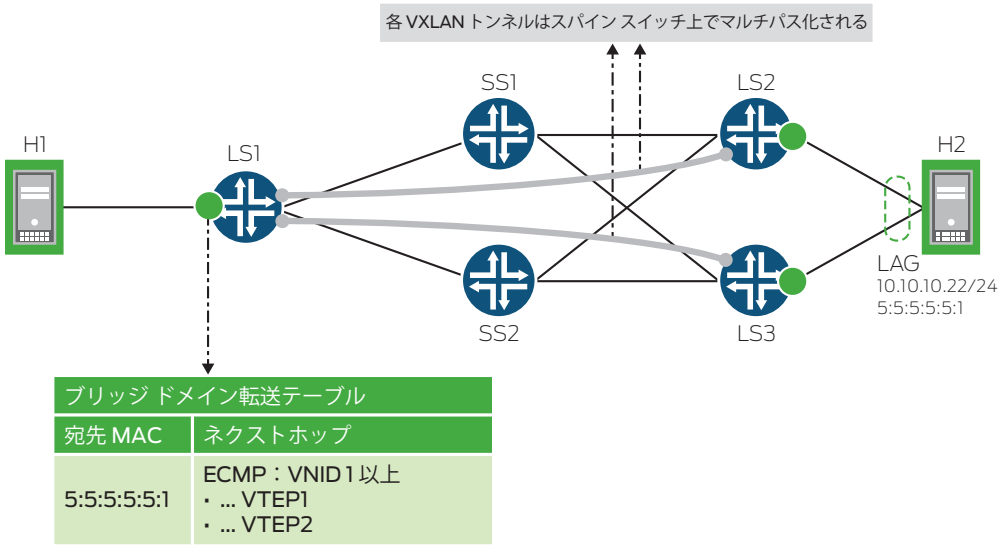


図 7：LS2 と LS3 を経由した LS1 から H2 へのマルチパス

図 7 は、VXLAN トンネルを経由した、H2 に到達するための LS1 から LS2 と LS3 両方へのマルチパスを示しています。ただし、LS2/LS3 ペアのうちの片方だけが H2 の MAC アドレスを学習した場合、問題が発生します（図 8 を参照）。このシナリオでは、マルチパスを成功させるために EVPN エイリアシングが必要です（図 9 を参照）。

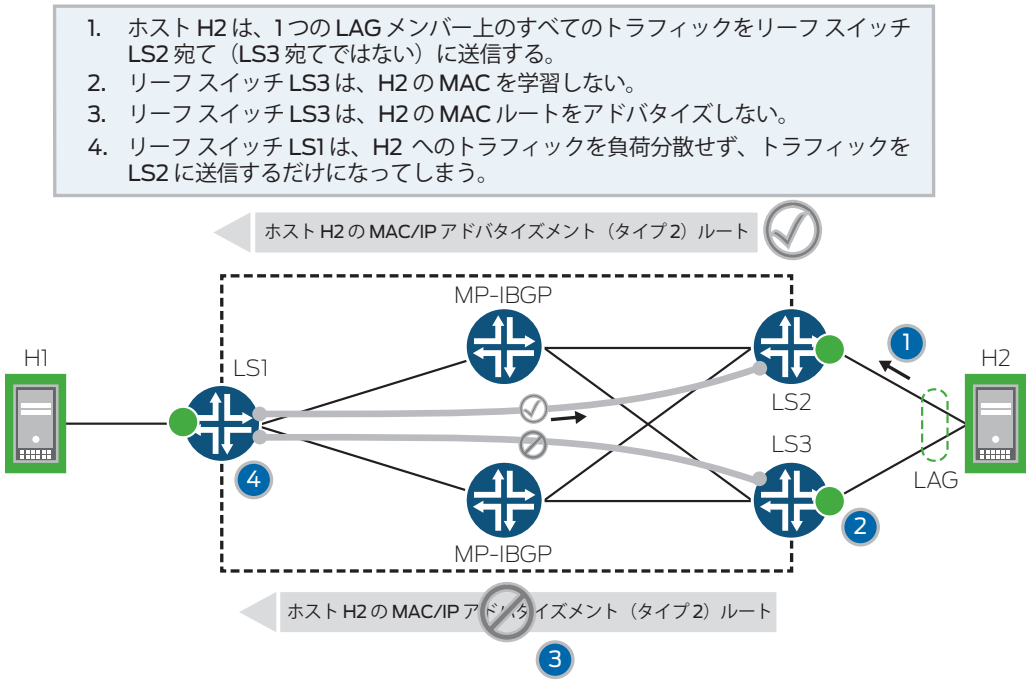


図 8：エイリアシングを使用せず、LS2 と LS3 を経由した LS1 から H2 へのマルチパスへの問題

このシナリオでエイリアシングを使用した場合を次の図 9 に示します。

1. LS1 は、LS2 と LS3 の両方からイーサネット自動検知（タイプ 1）ルートを受信する。
2. LS1 は、LS2 からのみ MAC/IP アドバタイズメント（タイプ 2）ルートを受信する。
   LS1 は、ESI ホスト H2 の場所を認識している。
   LS1 は、LS2 および LS3 を経由して H2 が存在する ESI に到達可能なことを認識している。
3. LS1 は、LS2 への VTP および LS3 への VTEP 経由の両方で ECMP トラフィックをホスト H2 に送信できる。

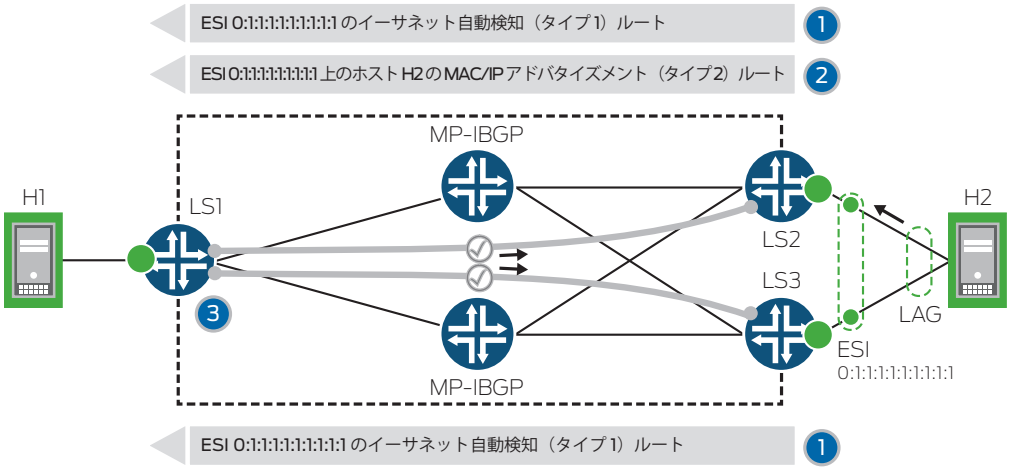


図 9：エイリアシングを使用し、LS2 と LS3 を経由した LS1 から H2 へのマルチパス

H1 は、LS2 からのタイプ 2 ルートと ESI 0:1:1:1:1:1:1:1:1:1 を使用して H2 の MAC アドレスを学習します。したがって、LS3 がこの同じ ESI のタイプ 1 をアドバタイズすることで、H2 の MAC アドレスが LS3 を介しても到達可能なことを判別できます。

EVPN の迅速なコンバージェンス

典型的なフラッドアンドラーン型プロトコルとレガシー L2 コントロール プレーンでは、リンク障害が多数の MAC アドレスへの到達性にかかわる場合、コンバージェンスまでの時間は長期化してしまう可能性があります。

障害後の再コンバージェンスには時間がかかる：

1. リーフ スイッチ LS3 とスイッチ S 間の接続がダウンする。
2. イーグレス リーフスイッチ LS3 が、スイッチ S の背後にあるすべてのホスト（H2～H100）のルートを取り消す。
3. イングレス リーフ スイッチ LS1 が、取り消されたすべてのルートを転送テーブルから削除する。

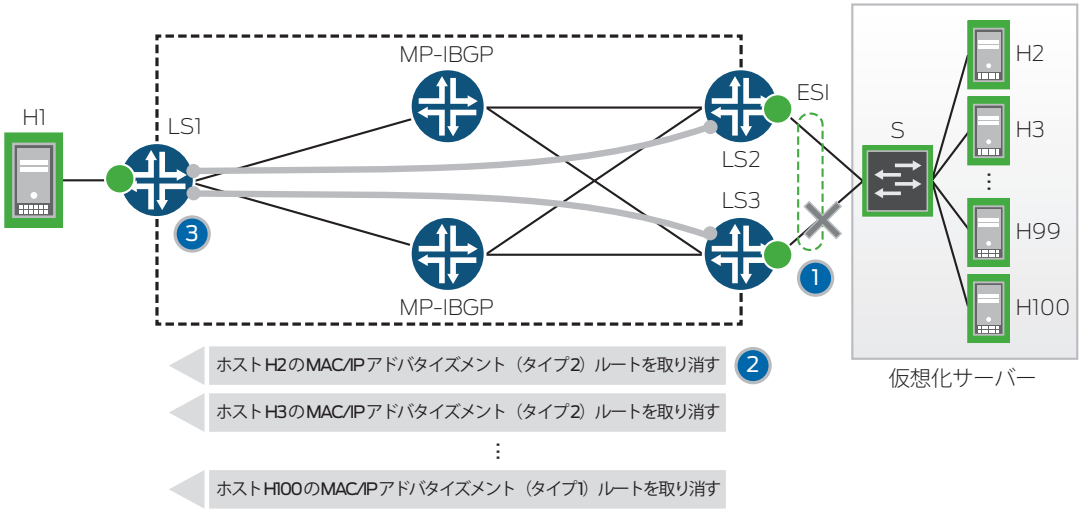


図 10：個々の MAC アドバタイズメントによる低速なコンバージェンス

図 10 では、LS2 と LS3 およびホスト H2 ～ H100 の間に中間のマルチホームド L2 スイッチ（S1）を配置しました。S1 は、物理スイッチ、または通常ハイパーバイザ上で稼働する仮想スイッチ / ルーターが使用されます。

LS3 が S1 とのリンクを失うと、LS1 宛ての 100 の MAC アドレス アドバタイズメントを取り消す必要があります。LS1 は、100 件の取り消しすべてを受信するまで、LS3 宛てにこれらのホストへのトラフィックを送信し続けます。

この問題を解決するために、EVPN では図 11 に示すエイリアシングのコンセプトを導入しています。

1. リーフ スイッチ LS3 とスイッチ S 間の接続がダウンする。
2. イーグレス リーフ スイッチ LS3 が、ホストのルートを取り消す前に ESI のルートを取り消す。
3. LS1 は、ESI が取り消されたことを認識すると、その ESI のすべてのルートを FIB から削除する。

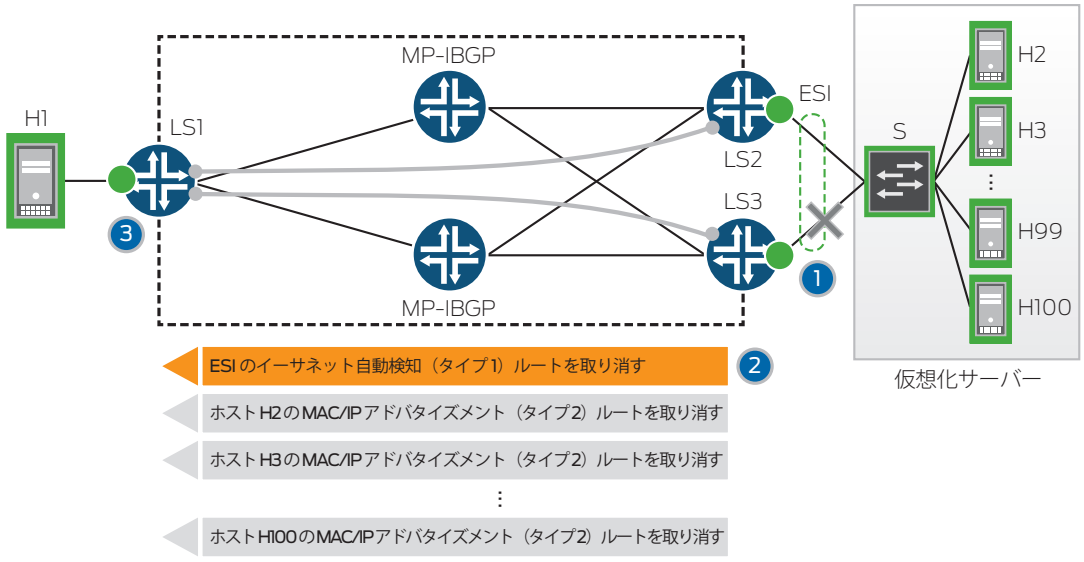


図 11：EVPN エイリアシングを使用した個々の MAC アドバタイズメント

EVPN エイリアシングを使用した場合、LS3 は最初にタイプ 1 ESI ルートを取り消してから、100 のタイプ 2 ルートをそれぞれ取り消します。

H1 は、タイプ 1 の取り消しを受信すると即座に、この ESI を通して LS3 から学習したすべての MAC アドレスをパージします。このため、万一リンク障害が発生した場合のコンバージェンスが大幅に向上します。

EVPN ブロードキャスト、未知のユニキャスト、マルチキャスト（BUM）トラフィックの概要

EVPN は、ブロードキャスト、未知のユニキャスト、マルチキャスト（BUM）トラフィック転送用の拡張性に優れたメカニズムを備えています。

EVPN の未知のユニキャスト フラッディングの大部分は、MAC アドレスの BGP コントロール プレーン プロパゲーションのおかげで回避されます。これは EVPN に備わっているメリットです。ただし、PE デバイス上のホストが別のホスト宛てにユニキャスト トラフィックを送信し、そのホストの MAC アドレス アドバタイズメントが PE デバイスにまだ届いていない場合には、競合状態が発生する可能性があります。

EVPN での BUM トラフィック転送方法は選択でき、一般的に、システム管理者には 2 つのオプションがあります。1 番目のオプションは、マルチキャスト アンダーレイ（アンダーレイ レプリケーションとも呼びます）を使用することです。2 番目のオプションは、オーバーレイでレプリケーションを実行することです（イングレス レプリケーションとも呼びます）。

2 つのオプションの相違点を図 12 に示します。

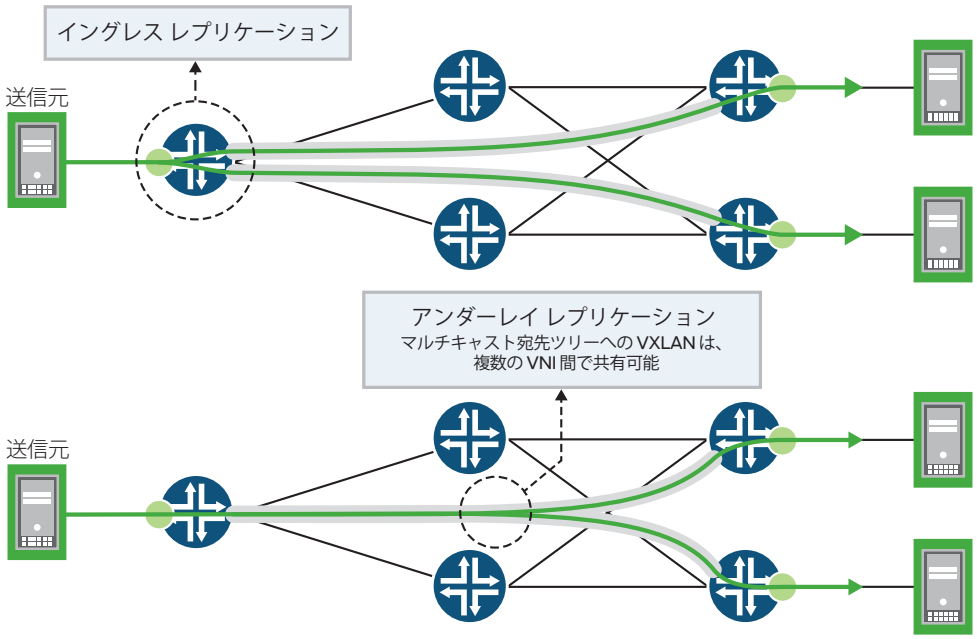


図 12：イングレス レプリケーションとアンダーレイ レプリケーションの比較

EVPN BUM トラフィック：アンダーレイ レプリケーション

アンダーレイで BUM トラフィックをレプリケートする利点は、その効率性にあります。アンダーレイ・マルチキャスト・ツリーを使用すると、パケットのレプリケーションは、それを必要とするトポロジー内の最も遠いポイントに向けてプッシュ送信されます。

アンダーレイを使用した場合、BUM トラフィック転送は、システム管理者の求めに応じて、細かくも粗くもできます。各 VNI に固有の独立したマルチキャスト グループをもたせるか、すべての仮想ネットワーク識別子（VNI）が同じマルチキャスト ツリーを共有することができます。あるいはこの 2 つをどのようにでも組み合わせることができます。

EVPN で BUM トラフィックを特定のマルチキャスト グループ宛てに転送する仕組みと、特定のマルチキャスト プロトコルを使用する仕組みは、タイプ 3 ルートによって可能になります。この詳細を図 13 に示します。

<table>
<tr><td rowspan="4">NLRI</td><td>ルート タイプ</td><td colspan="2">インクルーシブ マルチキャスト イーサネット タグ ルート（タイプ 3）</td></tr>
<tr><td>ルート識別子（RD）</td><td colspan="2">...</td></tr>
<tr><td>イーサネット タグ ID</td><td colspan="2">0</td></tr>
<tr><td>発信元 IP アドレス</td><td colspan="2">...</td></tr>
<tr><td colspan="2" rowspan="4">プロバイダ マルチキャスト サービス インターフェイス（PMSI）トンネル</td><td>フラグ</td><td>0（リーフ情報は不要）</td></tr>
<tr><td>トンネル タイプ</td><td>イングレス レプリケーション、PIM-SSM、PIM-SM、BIDIR-PIM、...</td></tr>
<tr><td>MPLS ラベル</td><td>0（未使用）</td></tr>
<tr><td>トンネル識別子</td><td>マルチキャスト グループ IP アドレス
送信者 IIP アドレス</td></tr>
<tr><td colspan="2">拡張コミュニティ</td><td>ルートターゲット</td><td></td></tr>
<tr><td colspan="2">他の属性（発信元、ネクストホップ、AS-Path、Local-Pref など）</td><td>...</td><td></td></tr>
</table>

図 13：EVPN タイプ 3 ルート

EVPN BUM トラフィック：イングレス レプリケーション

イングレス レプリケーションはアンダーレイ レプリケーションほど効率的ではありませんが、アンダーレイにおけるマルチキャスト プロトコル管理の複雑性を解消できるというメリットがあります。この理由から、データ センター導入における EVPN ではイングレス レプリケーションを使用するのが主流となっています。

この方法では、イングレス PE デバイスが、このデータの受信を必要とするすべてのイーグレス PE デバイス宛てにデータのユニキャスト コピーを作成します。コピーは、それぞれそのユニキャスト VXLAN トンネル上で転送されます。

EVPN イングレス レプリケーション：スプリット ホライズンと代表フォワーダ

CE デバイスが 1 台の PE デバイスとのみ接続している場合、ループの防止には EVPN のスプリット ホライズン ルールがあれば十分です。

PE デバイスがローカル CE デバイスから BUM パケットを受信すると、次のようになります。

- 同じ VLAN 内のローカル CE デバイス宛てにフラッドする
- 同じ VLAN 内のリモート PE デバイス宛てにフラッドする
- 送信元の CE デバイス宛てにはフラッドしない

PE デバイスがリモート PE デバイスから BUM パケットを受信すると、次のようになります。

- 同じ VLAN 内のローカル CE デバイス宛てにフラッドする
- リモート PE デバイス宛てにはフラッドしない

EVPN スプリット ホライズン ルールを図 14 に示します。

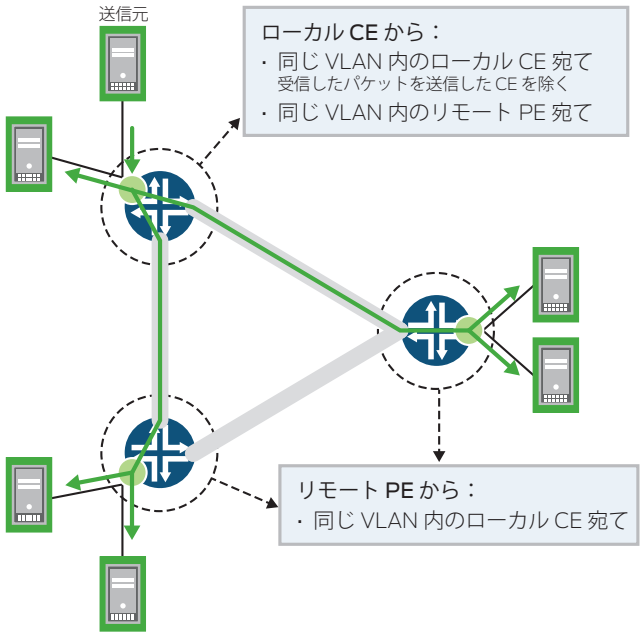


図 14：EVPN スプリット ホライズン

CE デバイスが複数の PE デバイス宛てにマルチホーミングされている場合、代表フォワーダのコンセプトが必要です。代表フォワーダを使用しないと、マルチホーミングされたホストは重複したパケットを受信します。

以下の図 15 について考えてみましょう。

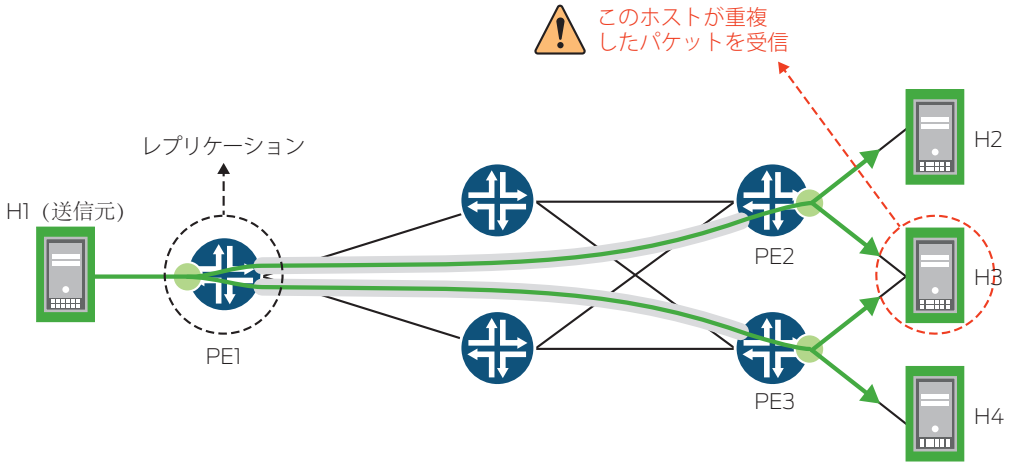


図 15：EVPN のイングレス レプリケーションと代表フォワーダの必要性

スプリット ホライズン ルールのみに従った場合、PE1 は H1 の BUM トラフィックを PE2 および PE3 方向へのトンネル上でレプリケートします。PE2 と PE3 は、どちらも同じ VLAN 上にローカル ホストをもっています。

スプリット ホライズン ルールに従うと、PE2 は、PE1 に戻るこの BUM トラフィックをレプリケートせず、PE1 宛てのトラフィックもレプリケートしません。PE2 は、同じルールに従って、同じ VLAN 上のローカルに接続されているホスト H2 と H3 にトラフィックを転送します。

同様に、PE3 はオーバーレイに戻る H1 のトラフィックをレプリケートせずに、同じ VLAN 上のローカル ホスト H3 と H4 に転送します。ここで問題が発生します。H3 は PE2 と PE3 両方から H1 の重複したトラフィックを受信するためです。

したがって、図 16 に示す代表フォワーダが必要になります。

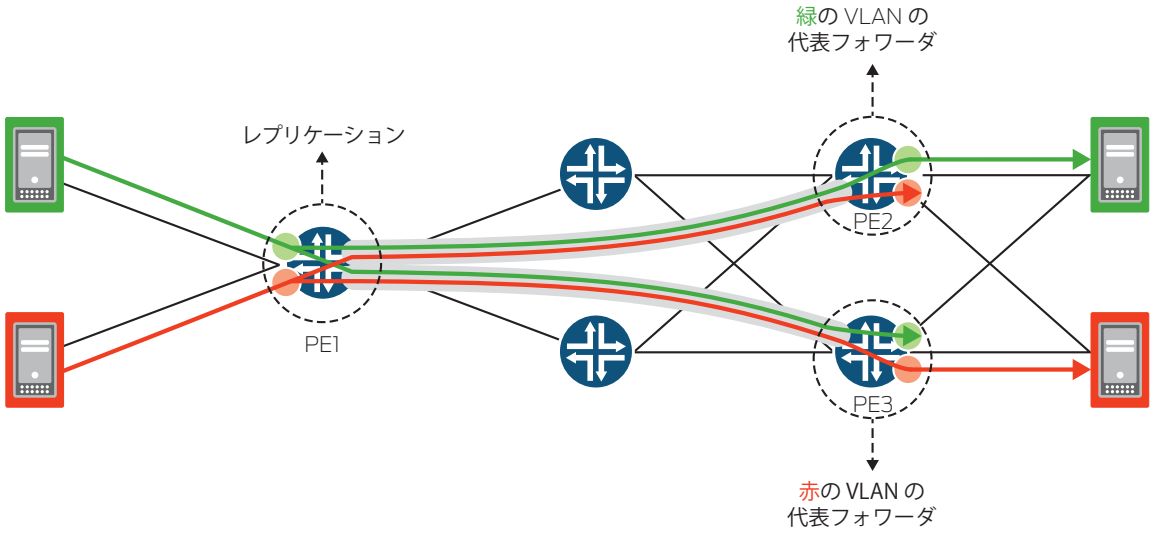


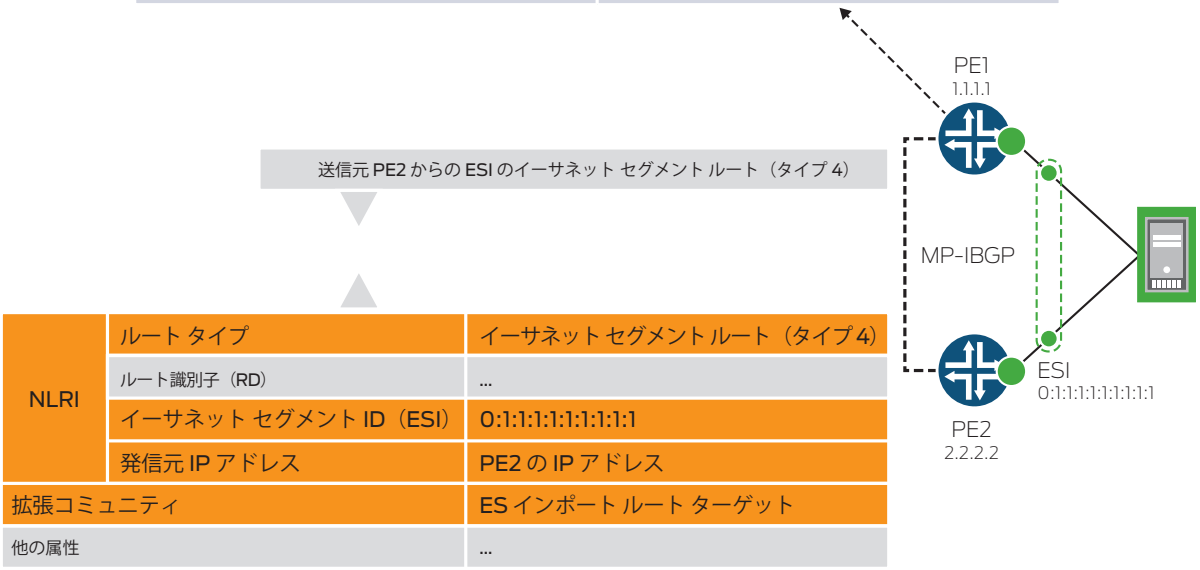
図 16：EVPN イングレス レプリケーションと代表フォワーダ

PE2 は緑の VLAN の代表フォワーダになり、PE3 は赤の VLAN の代表フォワーダになります。こうすると、代表フォワーダの負荷が分散され、マルチホームド ホストに重複したパケットが送信されません。

ESI の代表フォワーダは、図 17 に示すように、タイプ 4 ルート アドバタイズメントに基づき選択されます。

ESI 0:1:1:1:1:1:1:1:1:1 の指定フォワーダ テーブル

送信元（IP アドレスでソート）	VLAN の指定フォワーダ
PE1 (1.1.1.1)	0、2、4、6、...、4094
PE2 (2.2.2.2)	1、3、5、7、...、4095



NLRI	ルート タイプ	イーサネット セグメント ルート（タイプ 4）
	ルート識別子（RD）	...
	イーサネット セグメント ID（ESI）	0:1:1:1:1:1:1:1:1:1
	発信元 IP アドレス	PE2 の IP アドレス
拡張コミュニティ		ES インポート ルート ターゲット
他の属性		...

図 17：タイプ 4 アドバタイズメントに基づく ESI 代表フォワーダ

- PE デバイスは、接続されたイーサネット セグメントの ESI を検知すると、ES ルート、および関連する ES- インポート拡張コミュニティ属性をその MP-BGP ピアにアドバタイズします。

次に、PE デバイスがタイマーを起動し（デフォルト値は 3 秒）、他の PE デバイスまたは同じ ES に接続しているノードからの、イーサネット セグメント ルートの受信を許可します。このタイマー値は、同じ ES に接続されたすべての PE デバイス間で同じです。

- タイマーが切れると、各 PE デバイスは、ES に接続されたすべての PE ノードのアドレス（それ自体のものも含む）に関する順序付けられたリストを、番号値を増やしながら作成します。次に、すべての PE デバイス間で、この ES をアドバタイズしている PE ノードから番号順に、VLAN が指定フォワーダにラウンドロビンで割り当てられ、その結果、すべての PE ノードの指定フォワーダが一致します。

スプリット ホライズンと代表フォワーダの両方を使用しても、追加ルールに従っていない場合には、マルチホームド CE デバイスが重複したパケットを受信することがまれにあります。図 18 について考えてみましょう。

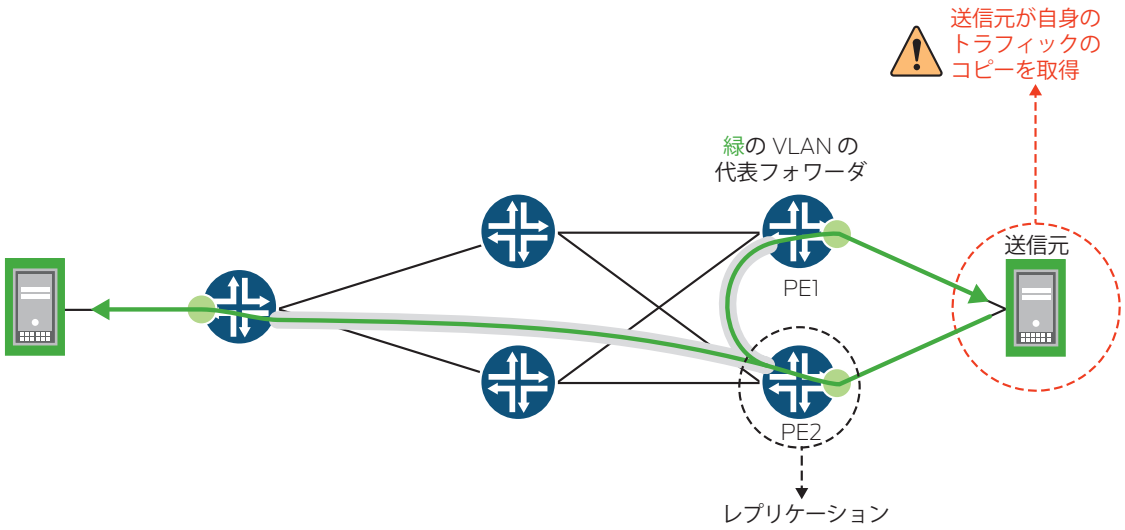


図 18：EVPN イングレス レプリケーション：代表フォワーダを使用して送信元にループ バックされるトラフィック

この例では、送信元はそのインターフェイスから PE2 宛ての BUM トラフィックを送信しています。PE2 は、この ES/VLAN の代表フォワーダではありません。

スプリット ホライズン ルールに従うと、PE2 は PE1 宛てのトラフィックをレプリケートします。PE1 は代表フォワーダです。追加ルールに従わないと、この代表フォワーダはこの BUM トラフィックを忠実に送信元に転送します。このため、送信元は自身のトラフィックを受信します。

最後に、ローカル バイアスのコンセプトを紹介します。このルールには以下が含まれます。

- ローカルに接続された CE デバイスから BUM トラフィックを受信する PE デバイスは、同じイーサネット タグ内のすべてのローカル サーバー宛て、および同じイーサネット タグに属するすべてのリモート PE デバイス宛てに転送します。
- PE デバイスがピア PE デバイスから BUM トラフィックを受信すると、以下を実行します。
  - この BUM トラフィックについて、VTEP の送信元 IP アドレスを確認する。
  - この送信元 IP アドレスからタイプ 4（イーサネット セグメント AD）ルートを調べる。
  - ESI がこの送信元 IP のタイプ 4 ルートに存在する場合、この ESI 宛ての BUM トラフィックをドロップする。
  - 存在しない場合、この同じイーサネット タグのすべてのローカル CE デバイス宛てに転送する。

EVPN の MAC の移動

L3 ファブリックで L2 ドメインを増やす場合に EVPN が提供する主なメリットの 1 つは、ワークロードのシームレスな移行が可能だという点です。

急な移動を何度も行った場合に発生する競合状態を防ぐため、EVPN にはタイプ 2 ルートのシーケンス番号付け機能があります。

図 19 は、ホスト H が当初は L1 に接続していたというシナリオを示しています。次に、ホスト H を L2 に移動し、すぐにもう一度 L4 に移動します。

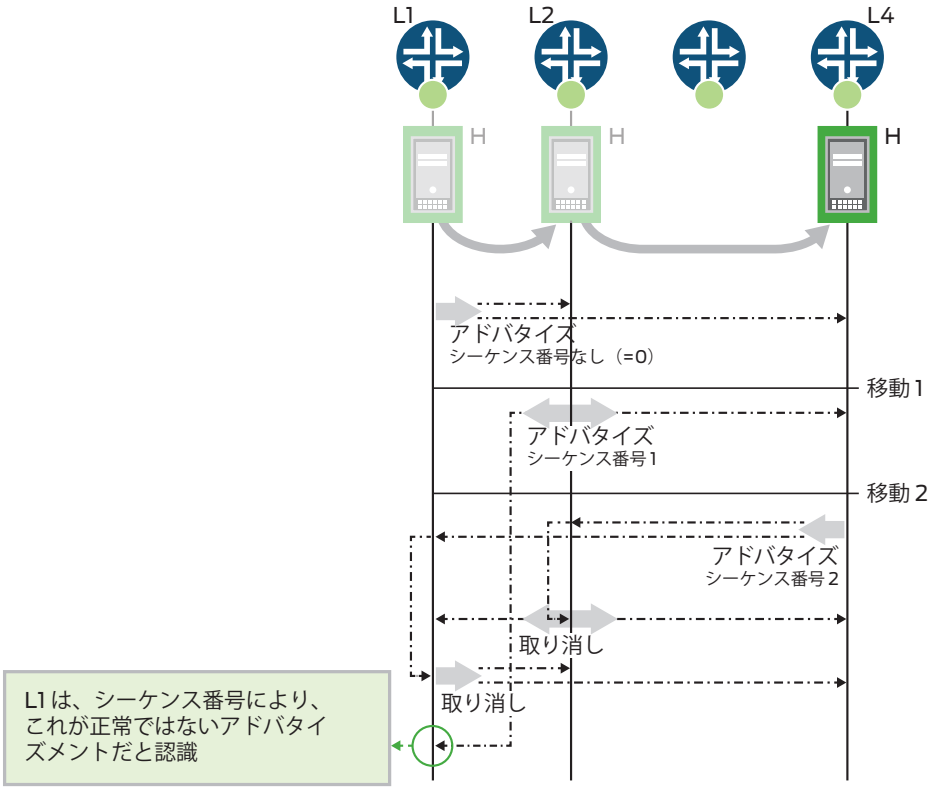


図 19：複数回のホスト移動による EVPN の MAC の移動

- 当初の状態では、L1 は最初のホスト H のタイプ 2 ルートをシーケンス 0 でアドバタイズしていました。ホスト H はそれまで EVPN コントロール プレーンに存在しなかったからです。
- ホスト H が L2 に移動した後に、L2 はホスト H の新しいタイプ 2 ルートをシーケンス 1 でアドバタイズします。このアドバタイズメントは L4 に到達しますが、L1 にはまだ到達していません。
- ホスト H が L4 に移動した時点で、L4 はホスト H の新しいタイプ 2 ルートをシーケンス 2 でアドバタイズします。L2 はこの更新されたルートを受信し、ホスト H が移動したことを認識し、シーケンス 1 ルートの「取り消し」メッセージを送信します。
- L1 は L4 のアドバタイズメント（シーケンス 2）および L2 の取り消しメッセージ（シーケンス 1）を受信します。L1 は自身のシーケンス 0 に関する自身の取り消しメッセージを送信します。
- 最後に、L1 は L2 のシーケンス 1 ルートを遅れて受信します。L1 は既に新しいシーケンス 2 ルートを学習しているため、この古いアドバタイズメントを安全に破棄できます。

EVPN の分散型デフォルト ゲートウェイ

EVPN は一意で拡張性の高いソリューションを提供できるため、従来の IP ゲートウェイを任意の数のネットワーク要素にアクティブに分散させることができます。これは、L2 テナントがファブリックのどこかに存在する可能性があるクラウド環境において、特に関連性があります。

機能という観点からは、分散型デフォルト ゲートウェイをすべてのホストのできる限り近くに配置することが求められます。したがって、リーフ上に分散型デフォルト ゲートウェイを配置するアーキテクチャになります。

ただし、QFX5100 などの Broadcom Trident2 を使用したプラットフォームでは、VXLAN ルーティングは単一のハードウェア パス [4] においてサポートされません。このため、スパインまたはコアで、この VXLAN ルーティング機能を L3 Clos 階層まで移動する必要があります。

図 20 の例を参照してください。

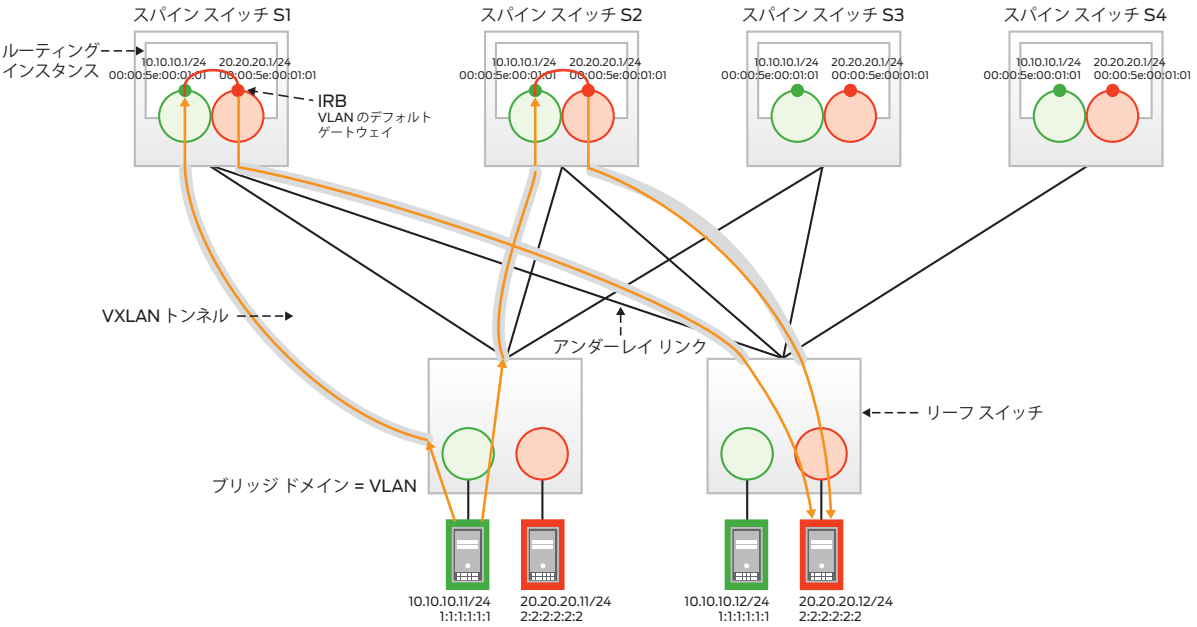


図 20：スパインにおける EVPN の分散型デフォルト ゲートウェイ

この例では、2 台のリーフ スイッチと 4 台のスパインが示されています。スパインごとに、緑と赤の VNI に対して統合ルーティング / ブリッジング（IRB）が設定されています。さらに、すべてのスパインは各 IRB で同じエニーキャスト IP と MAC も共有しています。

緑のホスト 10.10.10.11 は、赤のホスト 20.20.20.12 宛てに、一意なトラフィック フローを 2 つ送信します（SSH 接続 1 本と別の HTTPS 接続 1 本など）。これら 2 つの等価コスト マルチパス（ECMP）ストリームは、2 つの異なるエニーキャスト ゲートウェイ間で負荷分散されます。追加の ECMP パスは、すべてのエニーキャスト ゲートウェイへのすべてのパスを使用します。

任意の数のエニーキャスト ゲートウェイ間でのこのようなアクティブ / アクティブ負荷分散は、CE デバイスのマルチホーミングと同じメカニズムを使用することで可能になります。

コンセプト上は、すべてのスパイン上にマルチホームド「IRB」があると考えることができます。各スパインは、この IRB のタイプ 1 ESI および同じタイプ 2 MAC アドレスをアドバタイズします。リモート PE デバイスは、すべてのスパインにおけるこの同じ MAC と ESI への等価コスト到達性を確認します。図 21 は、2 つのスパイン、2 つの IRB、2 つのリーフを使用した単純な例を示しています。

[4] Broadcom Trident2 プラットフォームは、MPLS カプセル化されたルーティングをサポートします。したがって、MPLS の導入による EVPN+MPLS カプセル化は VXLAN カプセル化に伴うハードウェアの制限事項を解消します。

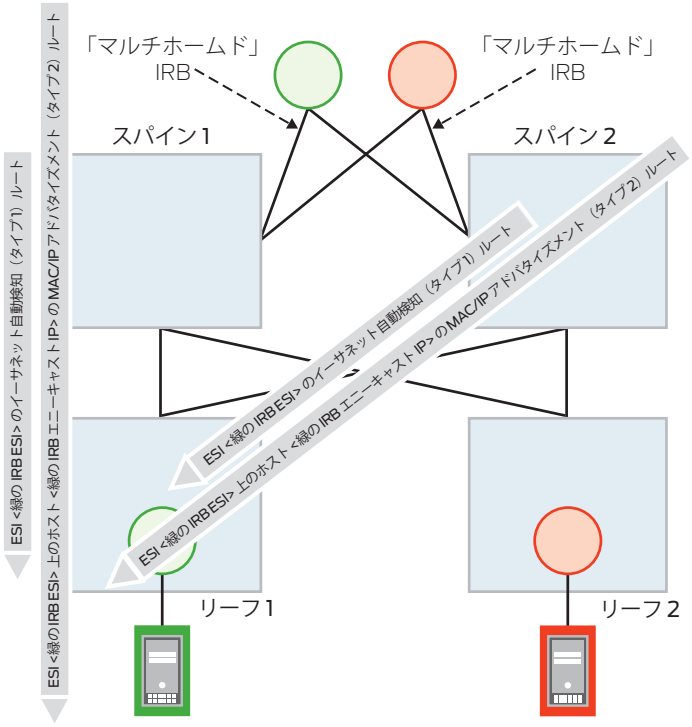


図 21：EVPN 分散型デフォルト ゲートウェイのルート アドバタイズメント

リーフ 1 がスパイン 1 とスパイン 2 の両方から、緑の IRB のタイプ 1 ESI アドバタイズメントを受信しているのがわかります。リーフ 1 は、スパイン 1 とスパイン 2 の両方から、緑の IRB MAC のタイプ 2 アドバタイズメントも受信しています。

同じように、リーフ 1 が赤の IRB のタイプ 1 とタイプ 2 のルートを受信し、リーフ 2 も スパイン 1 とスパイン 2 から両方の IRB の同じタイプ 1 とタイプ 2 のアドバタイズメントを受信します。

EVPN と VXLAN の構成

アンダーレイ

EVPN を構成する前に、まずネットワーク アンダーレイを利用するネットワークの設計を検討します。大規模なデータ センター導入では L3 Clos ファブリックが標準的ですが、設計には多くの選択肢があります。

図 22 は、異なる 3 階層で構成された大規模な 5 段階の L3 Clos ファブリックを示しています。ポッド内にリーフ スイッチまたは「トップオブラック」スイッチがあり、それらは 2 層目のスパイン層により集約されています。複数のポッドが、上位の「コア」層または「ファブリック」層を介して接続されています。トポロジー内には明確な階層が 3 つしかありませんが、これを一般的に 5 段階 Clos ファブリックと呼びます。ポッド A のサーバーはポッド B の別のサーバーと通信し、5 つのネットワーク要素を行き来するため、5 段階になります。

3 階層のそれぞれは、組織のニーズに応じて水平的に拡張でき、階層間のオーバーサブスクリプションも組織の要件に従って管理できます。

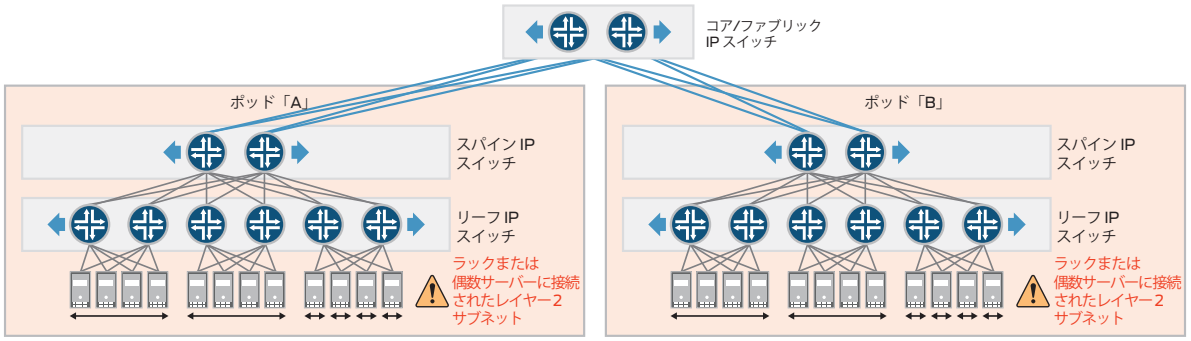


図 22：5 段階の L3 Clos ファブリック

ルーティング プロトコルは、この後に述べる設計や構成とともに、L3 Clos トポロジーを設計する際の重要な選択肢ですが、本資料のスコープ外です。規模の小さなトポロジーには、Dijkstra ベースのプロトコルを選択できます。ただし、大規模なトポロジーには一般的に BGP を使用します。

BGP では設計の選択肢が数多く提供されていますが、これもまた本資料のスコープ外です。各階層が単一の AS 番号（ASN）をもつか（図 23）、各ネットワーク デバイスがそれぞれ一意の ASN に属するか、という選択肢があります（図 24）。

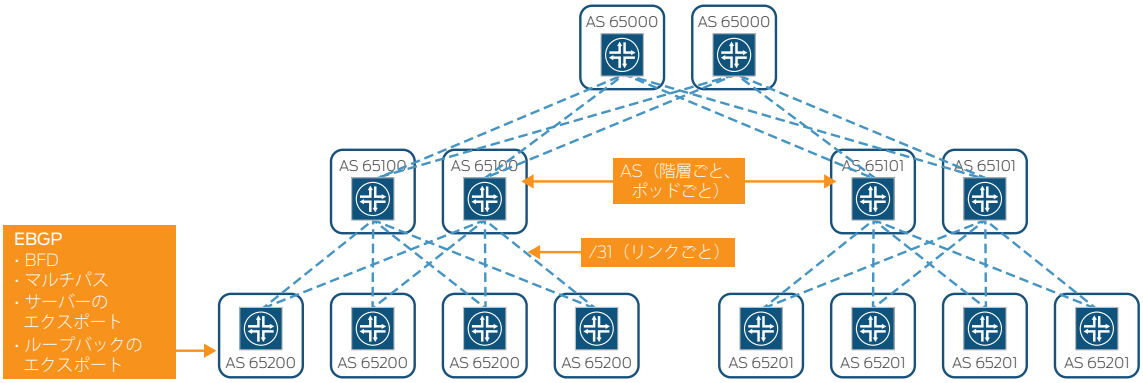


図 23：5 段階 L3 Clos ファブリック、階層ごと、ポッドごとに一意の ASN

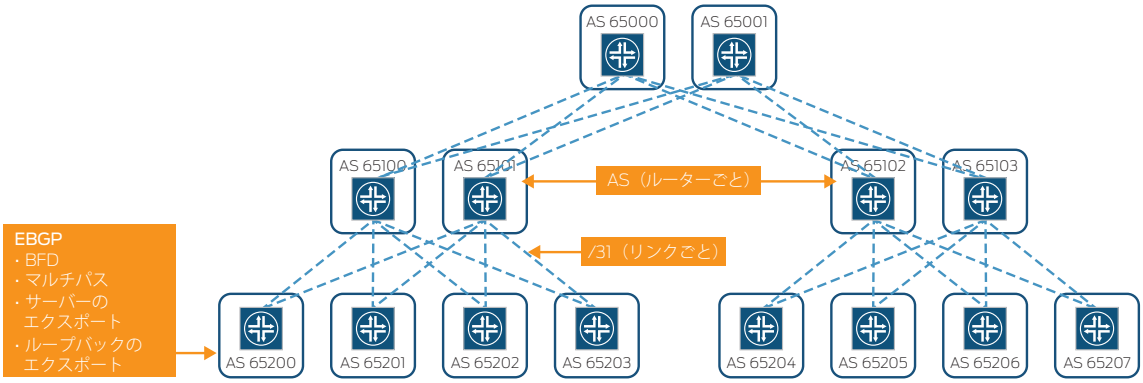


図 24：5 段階 L3 Clos ファブリック、デバイスごとに一意の ASN

ルーティング プロトコルおよび選択した設計に関係なく、アンダーレイではすべてのネットワーク要素のループバック アドレスの IP 到達性を提供する必要があります。このループバック アドレスは、オーバーレイ BGP コントロール プレーン接続の確立に使用され、デバイス間にオーバーレイ VXLAN トンネルを構築するための VTEP 送信元（および宛先）インターフェイスとなります。

以降では、図 25 のトポロジーを参考にしながら説明します。

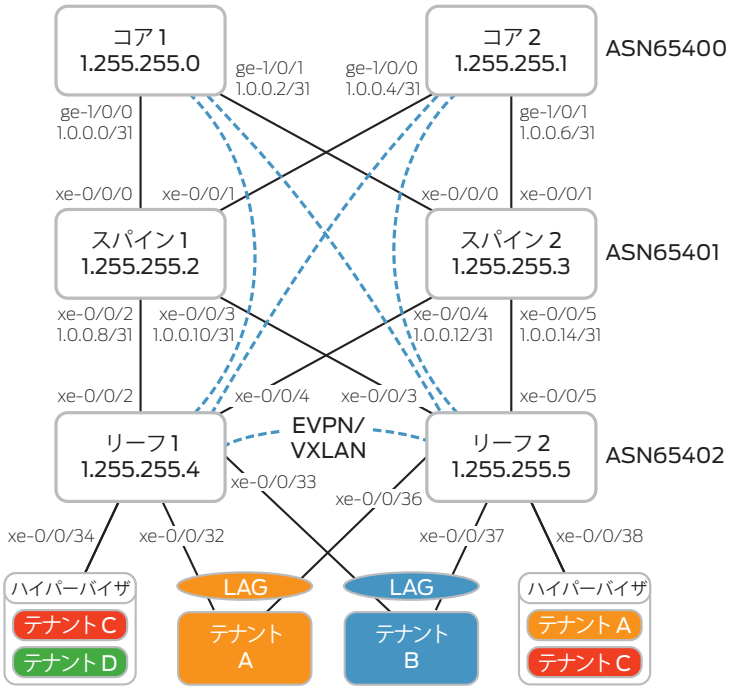


図 25：EVPN/VXLAN トポロジーの例

この例のトポロジーは、次のように構成されています。

- コア 1 とコア 2 は、ジュニパーネットワークス MX シリーズ 3D ユニバーサル エッジ ルーターで、EVPN オーバーレイの IP ゲートウェイとして機能します。
- スパイン 1 とスパイン 2 は、ジュニパーネットワークス QFX5100 スイッチで、オーバーレイの IP 転送専用として機能します。
- リーフ 1 とリーフ 2 は、ジュニパーネットワークス QFX5100 スイッチで、EVPN トポロジー内の PE デバイスとして機能します。
- 各階層は、単一の ASN で構成されています。
- EVPN コントロール プレーンには次の接続が含まれています。
  - リーフ 1 <-> リーフ 2
  - リーフ 1 <-> コア 1
  - リーフ 1 <-> コア 2
  - リーフ 2 <-> コア 1
  - リーフ 2 <-> コア 2

まずアンダーレイの外部 BGP（EBGP）を、リーフからコアまで上に向かって見ていきます。

リーフ 1

```
lab@leaf-1> show configuration routing-options
router-id 1.255.255.4;
autonomous-system 65402;
forwarding-table {
    export load-balance;
}

lab@leaf-1> show configuration policy-options policy-statement load-balance
term 1 {
```

```
    then {
        load-balance per-packet;
    }
}
```

スパインへのアンダーレイ BGP ピアリング セションには、2 つの重要な構成が導入されています。1 番目に、lo0 がエクスポートされているため、アンダーレイにアドバタイズされます。2 番目に、family inet unicast loops 2 を設定します。これが必要なのは、選択した設計において、階層内で同じ ASN を再利用するためです。

```
lab@leaf-1> show configuration protocols bgp group underlay
type external;
advertise-peer-as;
family inet {
    unicast {
        loops 2;
    }
}
export lo0;
peer-as 65401;
multipath;
neighbor 1.0.0.8 {
    description spine-1;
}
neighbor 1.0.0.12 {
    description spine-2;
}
lab@leaf-1> show configuration policy-options policy-statement lo0
from {
    family inet;
    protocol direct;
    route-filter 0.0.0.0/0 prefix-length-range /32-/32;
}
then accept;
```

advertise-peer-as ステートメントは、今は無視して、スパイン 1 のアンダーレイ設定の検証時に説明します。

デフォルトでは、ルーターは、EBGP ピアから受信したルートの更新において、自身の ASN が使用されている場合、ルートを拒否します。ただし、リーフ 1 は、アンダーレイ内にあるリーフ 2 のループバックを学習する必要があります。したがって、受信した AS_PATH は 65401 65402 になります。

これは、以下の出力に示されています。

```
lab@leaf-1> show route 1.255.255.5 detail table inet.0

inet.0: 13 destinations, 18 routes (13 active, 0 holddown, 0 hidden)
1.255.255.5/32 (2 entries, 1 announced)
        *BGP    Preference: 170/-101
                Next hop type: Router, Next hop index: 131070
                Address: 0x9760010
                Next-hop reference count: 84
                Source: 1.0.0.8
                Next hop: 1.0.0.8 via xe-0/0/2.0
                Session Id: 0x0
                Next hop: 1.0.0.12 via xe-0/0/4.0, selected
```

```
                Session Id: 0x0
                State: <Active Ext>
                Local AS: 65402 Peer AS: 65401
                Age: 2d 4:00:39
                Validation State: unverified
                Task: BGP_65401.1.0.0.8+61723
                Announcement bits (2): 0-KRT 2-Resolve tree 2
                AS path: 65401 65402 I (Looped: 65402)
                Accepted Multipath
                Localpref: 100
                Router ID: 1.255.255.2
        BGP     Preference: 170/-101
                Next hop type: Router, Next hop index: 1723
                Address: 0x95f8860
                Next-hop reference count: 7
                Source: 1.0.0.12
                Next hop: 1.0.0.12 via xe-0/0/4.0, selected
                Session Id: 0x0
                State: <NotBest Ext>
                Inactive reason: Not Best in its group - Active preferred
                Local AS: 65402 Peer AS: 65401
                Age: 2d 4:00:39
                Validation State: unverified
                Task: BGP_65401.1.0.0.12+60413
                AS path: 65401 65402 I (Looped: 65402)
                Accepted MultipathContrib
                Localpref: 100
                Router ID: 1.255.255.3

{master:0}
```

リーフ 1 が、スパイン 1 とスパイン 2 からの EBGP セッションでリーフ 2 のループバックを学習したことがわかります。ループの設定がなければ、このルートは破棄されていました。

スパイン1

```
lab@spine-1> show configuration routing-options
router-id 1.255.255.2;
autonomous-system 65401;
forwarding-table {
    export load-balance;
    ecmp-fast-reroute;
}

lab@spine-1> show configuration policy-options policy-statement load-balance
term 1 {
    then {
        load-balance per-packet;
    }
}
```

```
lab@spine-1> show configuration protocols bgp group underlay-leaf
type external;
advertise-peer-as;
family inet {
    unicast {
        loops 2;
    }
}
export lo0;
peer-as 65402;
multipath;
neighbor 1.0.0.9 {
    description leaf-1;
}
neighbor 1.0.0.11 {
    description leaf-2;
}

lab@spine-1> show configuration protocols bgp group underlay-core
type external;
advertise-peer-as;
family inet {
    unicast {
        loops 2;
    }
}
export lo0;
peer-as 65400;
multipath;
neighbor 1.0.0.0 {
    description core-1;
}
neighbor 1.0.0.4 {
    description core-2;
}

lab@spine-1> show configuration policy-options policy-statement lo0
from {
    family inet;
    protocol direct;
    route-filter 0.0.0.0/0 prefix-length-range /32-/32;
}
then accept;
```

スパイン 1 に関するアンダーレイ BGP の設定はリーフ 1 と同様ですが、ここでは、`bgp group underlay-leaf` 内での `advertise-peer-as` の使用について説明します。

リーフ 1 については、`family inet unicast loops 2` を設定して、リーフ 1 が、自身の ASN をもつ EBGP で学習したルートを拒否しないようにする必要がありました。

スパイン 1 は、リーフ 2 との直接的な EBGP セッションからリーフ 2 のループバックを学習します。ただし、EBGP のルールでは、スパイン 1 が特定の ASN に戻された ASN から学習したルートをもう一度アドバタイズしないことが定められています。したがって、正常な設定では、リーフ 1 はループの設定に関わらず、リーフ 1 のルートを受信しません。

したがって、鋭い読者の方はおわかりでしょうが、advertise-peer-as を使用することで、スパイン 1 はこのルールを迂回して、同じ ASN に戻すルートを再度アドバタイズできます。この点から、ループの設定は、求められているルート アドバタイズメントと学習の動作を完成させるために有用かつ必要です。

advertise-peer-as は、スパイン 1 およびスパイン 2 とのリーフ 1 の EBGP 設定においても設定されていたことを思い出してください。厳密には、これは必要ありません。スパインは EVPN には参加していないため、スパイン 1 がスパイン 2 のループバックを認識する必要はありません。逆も同様です。システム管理者はトラブルシューティングと一貫性の観点から、ループバックをどこにでも到達可能にしたがるため、リーフ 1 のオプション設定となっています。したがって、スパイン 1 の family inet unicast loops 2 設定もオプションです。

同じように、コア 1 とコア 2 も厳密には EVPN ピアではないため、他のループバックへの到達性は必要ありません。したがって、advertise-peer-as 設定は、bgp group underlay-core のスパイン 1 に関してはオプションです。

コア 1

```
lab@core-1> show configuration routing-options
router-id 1.255.255.0;
autonomous-system 65400;
forwarding-table {
    export load-balance;
    ecmp-fast-reroute;
}
lab@core-1> show configuration policy-options policy-statement load-balance
term 1 {
    then {
        load-balance per-packet;
    }
}
lab@core-1> show configuration protocols bgp
group underlay {
    type external;
    advertise-peer-as;
    family inet {
        unicast {
            loops 2;
        }
    }
    export lo0;
    peer-as 65401;
    multipath;
    neighbor 1.0.0.1 {
        description spine-1;
    }
    neighbor 1.0.0.3 {
        description spine-2;
    }
}

lab@core-1> show configuration policy-options policy-statement lo0
from {
```

```
    family inet;
    protocol direct;
    route-filter 0.0.0.0/0 prefix-length-range /32-/32;
}
then accept;
```

スパインと同様に、コア 1 も技術的にはコア 2 のループバックを学習する必要がありません。このため、コア 1 に関しては `family inet unicast loops 2` はオプションです。

コア 1 に関する唯一の追加設定は `forwarding-table ecmp-fast-reroute` です。MX シリーズ ルーター [5] については、特定の IP プレフィックスへの複数の ECMP パスが存在する場合、この追加設定によって迅速な L3 再コンバージェンスが可能になります。

オーバーレイ

アンダーレイでループバック アドレスへの到達性を確立できたので、EVPN/VXLAN オーバーレイを設定できます。

EVPN 設定で検討すべきコンポーネントは主に 2 つです。

1. Protocols bgp

   このセクションではマルチキャスト BGP（MBGP）セッションと EVPN ピアが、EVPN シグナリングで設定されています。

2. スイッチ レベルの EVPN 設定

   QFX5100 は単一の論理スイッチを提供しますが、MX シリーズは複数の仮想スイッチのオプションを提供します。これにより、この 2 者の間では EVPN 設定が異なる場合があります。

   物理スイッチ単位または仮想スイッチ単位で、次のような EVPN 固有の設定が必要です。

   - VTEP 送信元インターフェイス（lo0.0）
   - ルート識別子：MBGP で EVPN ルートのアドバタイズに使用される RD
   - vrf-import：スイッチの EVPN テーブルにインポートされるルート ターゲットを定義
   - vrf-export：EVPN ルートのアドバタイズに使用されるルート ターゲットを定義
   - protocols evpn
     - このスイッチ ドメインに属する VNI のリスト
     - BUM の転送方法（イングレス レプリケーションによる EVPN BUM トラフィック処理）
   - VLAN の設定
     - VNI から VLAN へのマッピング
     - BUM の転送方法

QFX5100 では、検討すべき 1 層目は `switch-options` の下にあります。

```
lab@leaf-1> show configuration switch-options
vtep-source-interface lo0.0;
route-distinguisher 1.255.255.4:1;
vrf-import vrf-imp;
vrf-target target:9999:9999;
```

`vtep-source-interface` は常に lo0.0 です。読者の方は、アンダーレイ ルーティング プロトコルを介してこれに到達可能な必要があることを思い出してください。

`route-distinguisher` は、ネットワーク全体のすべてのスイッチ（物理スイッチ、MX シリーズ ルーターの場合は仮想スイッチ）で一意である必要があります。マルチプロトコル MGP（MP-MGP）内では、これによってすべてのルート アドバタイズメントがグローバルで一意になります。

少なくとも QFX5100 の場合、`vrf-target` は、これを使用してスイッチがすべての ESI（タイプ 1）ルートを送信するコミュニティになります。

[5] このことは、ジュニパーネットワークス PTX シリーズ パケット トランスポート ルーターにも当てはまります。

最後に、vrf-import vrf-imp はターゲット コミュニティ リストを定義します。このリストは、bgp.evpn.0 から default-switch.evpn.0 インスタンスにインポートされます。

```
lab@leaf-1> show configuration policy-options policy-statement vrf-imp
term t1 {
    from community com100;
    then accept;
}
term t2 {
    from community com200;
    then accept;
}
term t3 {
    from community com300;
    then accept;
}
term t4 {
    from community com400;
    then accept;
}
term t5 {
    then reject;
}
lab@leaf-1> show configuration policy-options | grep members
community com100 members target:1:100;
community com200 members target:1:200;
community com300 members target:1:300;
community com400 members target:1:400;
```

次に、protocols evpn の設定について説明します。

```
lab@leaf-1> show configuration protocols evpn
encapsulation vxlan;
extended-vni-list [ 1100 1200 1300 1400 ];
multicast-mode ingress-replication;
vni-routing-options {
    vni 1100 {
        vrf-target export target:1:100;
    }
    vni 1200 {
        vrf-target export target:1:200;
    }
    vni 1300 {
        vrf-target export target:1:300;
    }
    vni 1400 {
        vrf-target export target:1:400;
    }
}
```

extended-vni-list を使用して、EVPN/VXLAN MP-BGP ドメインに属する VNI を設定します。読者の方は、前述の EVPN で使用可能な BUM レプリケーション オプションを思い出してください。EVPN 環境における VXLAN イングレス レプリケーションのメリットを考慮し、ここではマルチキャスト アンダーレイを使用しません。

vni-routing-options の下で、VNI インスタンスごとに異なるルート ターゲット（RT）を設定します。後で、タイプ 2 ルートがこれらの RT を使用してエクスポートされていることを確認します。

```
lab@leaf-1> show configuration vlans
v100 {
    vlan-id 100;
    vxlan {
        vni 1100;
        ingress-node-replication;
    }
}
v200 {
    vlan-id 200;
    vxlan {
        vni 1200;
        ingress-node-replication;
    }
}
v300 {
    vlan-id 300;
    vxlan {
        vni 1300;
        ingress-node-replication;
    }
}
v400 {
    vlan-id 400;
    vxlan {
        vni 1400;
        ingress-node-replication;
    }
}
```

vlans スタンザの下で、ローカルに有効な vlan-id をグローバルに有効な VNI にマップします。繰り返しますが、マルチキャスト アンダーレイに依存するのではなく、イングレス レプリケーションを設定します。

最後に、EVPN MP-BGP セッションを設定します。

```
lab@leaf-1> show configuration protocols bgp group EVPN_VXLAN_CORE
type external;
multihop {
    ttl 255;
    no-nexthop-change;
}
local-address 1.255.255.4;
family evpn {
    signaling;
}
```

```
peer-as 65400;
neighbor 1.255.255.0 {
    description core-1;
}
neighbor 1.255.255.1 {
    description core-2;
}

{master:0}
lab@leaf-1> show configuration protocols bgp group EVPN_VXLAN_LEAF
type internal;
local-address 1.255.255.4;
family evpn {
    signaling;
}
neighbor 1.255.255.5 {
    description leaf-2;
}
```

ここでは、`family evpn signaling` を使用して EVPN MP-BGP セッションを設定します。EBGP EVPN セッションの場合、これは常に `multihop` が指定されたループバック ピアリングになります。

リーフ 1 の `EVPN_VXLAN_CORE` 設定の下に、`no-nexthop-change` があることに気付いた方がいるかもしれません。これは、様々な要素を組み合わせて指定された今回の設計トポロジーに必要な設定です。

1番目に、このトポロジー例では階層ごとに単一の ASN を使用します。したがって、リーフ 1 からリーフ 2 への EVPN MP-BGP セッションは内部 BGP（IBGP）を介して実行されます。2 番目に、リーフ 1 とリーフ 2 は両方とも eBGP を介して各コアとピアになっています。

図 25 でリーフ 2 に接続されたシングルホームド Tenant_A CE デバイスを参考にして、コア 1 が受信する Tenant_A のタイプ 2 ルートを検討しましょう。コア 1 は、BGP 送信元と、リーフ 2 のループバック アドレスのプロトコルネクストホップを使用して、リーフ 2 の直接的な EBGP アドバタイズメントを受信します。

リーフ 2 もまたこの同じタイプ 2 ルートをリーフ 1 にアドバタイズします。リーフ 1 は、それ以降、BGP 送信元と、リーフ 1 自体のループバック アドレスのプロトコルネクストホップを使用して、このタイプ 2 ルートをコア 2 にアドバタイズします。このルートが IBGP を介してリーフ 2 から学習されたためです。

コア 1 は同じタイプ 2 ルートのそれ以外は等しいコピーを 2 つもつようになりました。ただし、コア 1 がリーフ 1 のプロトコルネクストホップによるタイプ 2 ルートを使って転送しようとした場合、このトラフィックはブラック ホール化されます。コア 1 がその VTEP からリーフ 1 までこのトラフィックを送信した場合、リーフ 1 はその VTEP からコア 1 までのこのトラフィックを正常に受信します。ただし、リーフ 1 のこの Tenant_C MAC への到達可能性は、その VTEP からリーフ 2 までです。トラフィックがドロップされるのは、このポイントです。EVPN スプリット ホライズン ルールでは、PE に向かうインターフェイスで受信したトラフィックは、PE に向かう別のインターフェイスから再度転送してはならないと定められています（『EVPN イングレス レプリケーション：スプリット ホライズンと代表フォワーダ』を参照）。

したがって、元のプロトコルネクストホップを保持して、コア 1（またはコア 2）が、特定のシングルホームド CE デバイスが接続された PE デバイスに直接送信するには、`no-nexthop-change` を設定する必要があります。

しかし、注目すべきことに、現実の導入においてこの設定が必要になることはないようです。これには、複数の要因があります。PE デバイスごとに単一の AS を使用した場合、このような IBGP/EBGP のやり取りは存在せず、この設定は不要です。そのうえ、さらに重要なことは、中規模から大規模な EVPN 導入の場合、PE デバイスの数が増えるにつれ、EVPN BGP メッシュの設定が大変になるため、これを設定することがあまりないということです。

代わりに考えられる現実的な方法として、スタンドアロン型 MBGP EBGP ルート リフレクタまたはルート サーバー（ジュニパーネットワークス Junos® OS 仮想ルート リフレクタ機能など）の導入があります。この場合、元のプロトコルネクストホップは保持されます。各 PE デバイスは、代わりにこのルート リフレクタ（または冗長構成のルート リフレクタ）への単一の MP-EBGP セッションを実行します。

またリーフ 1 にも、インターフェイスレベルの追加設定を必要とするマルチホームド テナントがあります。

```
lab@leaf-1> show configuration interfaces xe-0/0/32
ether-options {
    802.3ad ae0;
}
```

```
{master:0}
lab@leaf-1> show configuration interfaces xe-0/0/33
ether-options {
    802.3ad ae1;
}

{master:0}
lab@leaf-1> show configuration interfaces ae0
esi {
    00:01:01:01:01:01:01:01:01:01;
    all-active;
}
aggregated-ether-options {
    lacp {
        active;
        system-id 00:00:00:01:01:01;
    }
}
unit 0 {
    family ethernet-switching {
        interface-mode access;
        vlan {
            members v100;
        }
    }
}

{master:0}
lab@leaf-1> show configuration interfaces ae1
esi {
    00:02:02:02:02:02:02:02:02:02;
    all-active;
}
aggregated-ether-options {
    lacp {
        active;
        system-id 00:00:00:01:01:01;
    }
}
unit 0 {
    family ethernet-switching {
        interface-mode access;
        vlan {
            members v200;
        }
    }
}
```

既にこれまで、Link Aggregation Control Protocol（LACP）対応アグリゲート イーサネット（LAG）インターフェイスを 2 つ設定しました。EVPN 固有の設定は esi の下にあります。最初に、EVPN ドメイン全体でグローバルに一意のエンド システム識別子（ESI）の値を設定する必要があります。all-active を設定することで、このマルチホームド テナントが接続されているすべての PE ルーターは CE デバイスとの間でトラフィックを転送でき、すべての CE ルーター リンクが有効に使用されます。

コア 1

コア 1（MX シリーズ）の設定の多くは、ルーティング インスタンス スタンザで行われます。ここには、仮想スイッチと仮想ルーターの両方があります。

```
lab@core-1> show configuration routing-instances
VRF_Tenant_A {
    instance-type vrf;
    interface irb.1100;
    route-distinguisher 1.255.255.0:1100;
    vrf-target target:10:100;
}
VRF_Tenant_B {
    instance-type vrf;
    interface irb.1200;
    route-distinguisher 1.255.255.0:1200;
    vrf-target target:10:200;
}
VRF_Tenant_C {
    instance-type vrf;
    interface irb.1300;
    route-distinguisher 1.255.255.0:1300;
    vrf-target target:10:300;
}
VRF_Tenant_D {
    instance-type vrf;
    interface irb.1400;
    route-distinguisher 1.255.255.0:1400;
    vrf-target target:10:400;
}
VS_VLAN100 {
    vtep-source-interface lo0.0;
    instance-type virtual-switch;
    route-distinguisher 1.255.255.0:100;
    vrf-import VS_VLAN100_IMP;
    vrf-target target:1:100;
    protocols {
        evpn {
            encapsulation vxlan;
            extended-vni-list 1100;
            multicast-mode ingress-replication;
        }
    }
    bridge-domains {
        bd1100 {
            vlan-id 100;
            routing-interface irb.1100;
```

```
                vxlan {
                    vni 1100;
                    ingress-node-replication;
                }
            }
        }
    }
    VS_VLAN200 {
        vtep-source-interface lo0.0;
        instance-type virtual-switch;
        route-distinguisher 1.255.255.0:200;
        vrf-import VS_VLAN200_IMP;
        vrf-target target:1:200;
        protocols {
            evpn {
                encapsulation vxlan;
                extended-vni-list 1200;
                multicast-mode ingress-replication;
            }
        }
        bridge-domains {
            bd1200 {
                vlan-id 200;
                routing-interface irb.1200;
                vxlan {
                    vni 1200;
                    ingress-node-replication;
                }
            }
        }
    }
    VS_VLAN300 {
        vtep-source-interface lo0.0;
        instance-type virtual-switch;
        route-distinguisher 1.255.255.0:300;
        vrf-import VS_VLAN300_IMP;
        vrf-target target:1:300;
        protocols {
            evpn {
                encapsulation vxlan;
                extended-vni-list 1300;
                multicast-mode ingress-replication;
            }
        }
        bridge-domains {
            bd1300 {
                vlan-id 300;
                routing-interface irb.1300;
                vxlan {
```

```
                vni 1300;
                ingress-node-replication;
            }
        }
    }
}
VS_VLAN400 {
    vtep-source-interface lo0.0;
    instance-type virtual-switch;
    route-distinguisher 1.255.255.0:400;
    vrf-import VS_VLAN400_IMP;
    vrf-target target:1:400;
    protocols {
        evpn {
            encapsulation vxlan;
            extended-vni-list 1400;
            multicast-mode ingress-replication;
        }
    }
    bridge-domains {
        bd1400 {
            vlan-id 400;
            routing-interface irb.1400;
            vxlan {
                vni 1400;
                ingress-node-replication;
            }
        }
    }
```

最初に、各 VNI に 1 つずつ、全部で 4 つある、個々の仮想スイッチ インスタンスについて見てみます。各仮想スイッチのコア 1 で設定する項目は、単一 `switch-options` 階層のリーフ 1 で設定したものと同様です。

たとえば、VS_VLAN400 の場合、グローバルに有効な RD、`vrf-target`、この仮想スイッチの `vrf-import` リストを定義します。VNI 1400 がこの EVPN インスタンスに属するように指定し、（マルチキャスト アンダーレイではなく）イングレス レプリケーションを指定し、ローカルに有効な VLAN400 をグローバルに有効な VNI 1400 に関連付けます。

コア 1 は VNI 1400 のデフォルト ゲートウェイとして機能するため、ブリッジ ドメインの下の IRB インターフェイスとも関連付けます。

さらに、各テナントに対して、IP VPN 仮想ルーティングおよび転送（VRF）テーブルを設定します。これによって、テナント間で L3 が分離され、これらの IP VPN VRF を MPLS に対応する WAN 全体に拡張できます。

この例では、ブリッジ ドメインと IP VPN 間に 1 対 1 のマッピングがあります。これは制限事項ではありません。ただし、テナントには複数のブリッジ ドメイン（および複数の IRB）が関連付けられていることがあります。この場合、複数の IRB を同じ VRF インスタンス内に設定して、L3 接続を可能にすることができます。

各仮想スイッチに固有の `vrf-import` ポリシーが設定されていることに注意してください。このような個々のポリシーを次に示します。

```
lab@core-1> show configuration policy-options
policy-statement VS_VLAN100_IMP {
    term ESI {
        from community comm-leaf_esi;
```

```
                then accept;
            }
            term VS_VLAN100 {
                from community comm-VS_VLAN100;
                then accept;
            }
        }
        policy-statement VS_VLAN200_IMP {
            term ESI {
                from community comm-leaf_esi;
                then accept;
            }
            term VS_VLAN200 {
                from community comm-VS_VLAN200;
                then accept;
            }
        }
        policy-statement VS_VLAN300_IMP {
            term ESI {
                from community comm-leaf_esi;
                then accept;
            }
            term VS_VLAN300 {
                from community comm-VS_VLAN300;
                then accept;
            }
        }
        policy-statement VS_VLAN400_IMP {
            term ESI {
                from community comm-leaf_esi;
                then accept;
            }
            term VS_VLAN400 {
                from community comm-VS_VLAN400;
                then accept;
            }
        }
        policy-statement lo0 {
            from {
                family inet;
                protocol direct;
```

```
            route-filter 0.0.0.0/0 prefix-length-range /32-/32;
        }
        then accept;
    }
    policy-statement load-balance {
        term 1 {
            then {
                load-balance per-packet;
            }
        }
    }
    community comm-VS_VLAN100 members target:1:100;
    community comm-VS_VLAN200 members target:1:200;
    community comm-VS_VLAN300 members target:1:300;
    community comm-VS_VLAN400 members target:1:400;
    community comm-leaf_esi members target:9999:9999;
```

仮想スイッチ固有の各ポリシーには、2 つの項目があります。1 つ目は target:9999:9999 の受け入れです。これは、すべての仮想スイッチがすべてのリーフからタイプ 1 ESI ルートを確実にインポートするために必要です。2 つ目は、単一の VNI 固有の RT からタイプ 2 ESI ルートをインポートすることです。

これを見ると興味深いことがわかります。図 25 では、Tenant_A および Tenant_B という マルチホームド テナントが 2 つ存在します。厳密には、VS_VLAN100 が Tenant_B の ESI を認識する必要はなく、VS_VLAN200 が Tenant_A の ESI を認識する必要もありません。

vrf-target が（仮想）スイッチ レベルの設定であり、リーフ 1 が単一のスイッチ インスタンスをサポートすることを考慮すると、すべての ESI はすべての VRF 内にあります。ただし、Tenant_A が Tenant_B ESI へのタイプ 1 ルートをもつ可能性があるときは、この ESI に関連付けられたタイプ 2（MAC）ルートが存在するため、テナントの分離はそのまま残ります。

IRB レベルの設定を以下に示します。

```
lab@core-1> show configuration interfaces irb
unit 1100 {
    family inet {
        address 100.0.0.2/24 {
            virtual-gateway-address 100.0.0.1;
        }
    }
}
unit 1200 {
    family inet {
        address 200.0.0.2/24 {
            virtual-gateway-address 200.0.0.1;
        }
    }
}
unit 1300 {
```

```
    family inet {
        address 10.10.10.2/24 {
            virtual-gateway-address 10.10.10.1;
        }
    }
}
unit 1400 {
    family inet {
        address 10.10.10.2/24 {
            virtual-gateway-address 10.10.10.1;
        }
    }
}
```

ここで、2 つのことに気付きます。1 つ目は、これはマルチテナント環境であるため、テナント間の IP 重複は有効であるということです。2 つ目は、IRB ごとに `virtual-gateway-address` が設定されているということです。これは、コア 1 とコア 2 で共有される MAC アドレスと IP アドレスです。

この例ではゲートウェイ デバイスを 2 台しか使用していませんが、このソリューションは任意の数のデバイスに水平的に拡張可能だということは重要です。したがって、IRB ごとに冗長構成のアクティブなゲートウェイを多数配置し、そのすべてが各 PE デバイスからのトラフィックを有効に負荷分散することが可能です。このことは、2 台までしか拡張できなかった従来の「マルチシャーシ LAG」配置よりも明らかに優れています。EVPN によって提供される拡張性は、中規模から大規模のデータ センターの要件です。

注：本資料の公開時点では、VXLAN 機能を搭載した EVPN エニーキャスト ゲートウェイは、MX シリーズ および QFX10000 プラットフォームでサポートされています。QFX5100 のような Broadcom Trident2 チップセットをベースにしたプラットフォームは、VXLAN 間のネイティブなルーティングに対応しません。Broadcom Trident2+ で構築される将来のプラットフォームでは、この制限事項はありません。

最後に、リーフ 1 およびリーフ 2 宛ての MP-BGP セッションを設定する必要があります。

```
lab@core-1> show configuration protocols bgp group EVPN_VXLAN
type external;
local-address 1.255.255.0;
family evpn {
    signaling;
}
peer-as 65402;
multipath;
neighbor 1.255.255.4 {
    description leaf-1;
    multihop {
        ttl 255;
    }
}
neighbor 1.255.255.5 {
    description leaf-2;
    multihop {
        ttl 255;
    }
}
```

EVPN と VXLAN のトラブルシューティング

アンダーレイと EVPN オーバーレイの両方を設定すると、多数の便利なコマンドをトラブルシューティングに使用できます。

有効な EVPN ルートが一度受信されると、ハードウェアにプログラムされるまでは、ソフトウェアの以下のシーケンスが使用されます。

- bgp.evpn.0 = Junos OS ルーティング プロトコル プロセス（RPD）内のグローバル EVPN ルーティング テーブル
- default.switch.evpn.0（QFX5100 の場合）= Junos OS RPD 内のスイッチ レベル EVPN 転送テーブル
- <virtual-switch-name>.evpn.0（MX の場合）= Junos OS RPD 内の仮想スイッチ レベル EVPN 転送テーブル
- L2ALD（レイヤー 2 アドレス学習デーモン）
  “show ethernet-switching-table” || “show bridge mac-table”
- カーネル
  “show route forwarding-table”
- パケット転送エンジン（PFE）
  from FPC vty: “show l2 manager mac-table”

次に、シングルホームド CE デバイス、マルチホームド CE デバイス、エニーキャスト ゲートウェイという 3 つの例について説明します。

シングルホームド CE デバイス

1 番目の例では、Tenant_C シングルホームドからリーフ 1 への MAC の到達性について説明します。

まず MAC アドレスがリーフ 1 上でローカルに学習されていることを検証します。タイプ 2 ルートは、MAC が最初に学習された後に初めて、リーフ 1 によって生成されます。

```
lab@leaf-1> show ethernet-switching table vlan-id 300
MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent
static
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC, O -
ovsdb MAC)
Ethernet switching table : 5 entries, 5 learned
Routing instance : default-switch
   Vlan                MAC                 MAC      Logical           Active
   name                address             flags    interface         source
   v300                00:00:5e:00:01:01   DR       esi.1726
05:00:00:ff:78:00:00:05:14:00
   v300                00:21:59:c8:24:65   D        xe-0/0/34.0
   v300                00:21:59:c8:24:69   D        vtep.32771        1.255.255.5
   v300                40:a6:77:9a:43:f0   D        vtep.32769        1.255.255.0
   v300                40:a6:77:9a:47:f0   D        vtep.32770        1.255.255.1
```

上記の出力から、MAC 00:21:59:c8:24:65 が xe-0/0/34.0 の Tenant_C CE デバイスに対して正常に学習されていることがわかります。以下の出力から、コア 1 宛てのタイプ 2 ルートが生成されていることがわかります。

```
lab@leaf-1> show route advertising-protocol bgp 1.255.255.0 evpn-mac-address
00:21:59:c8:24:65
bgp.evpn.0: 77 destinations, 77 routes (77 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop        MED     Lclpref    AS path
  2:1.255.255.4:1::1300::00:21:59:c8:24:65/304
*                          Self                                  I
  2:1.255.255.4:1::1400::00:21:59:c8:24:65/304
*                          Self                                  I

default-switch.evpn.0: 67 destinations, 67 routes (67 active, 0 holddown, 0
hidden)
  Prefix                  Nexthop        MED     Lclpref    AS path
```

```
  2:1.255.255.4:1::1300::00:21:59:c8:24:65/304
*                          Self                                  I
  2:1.255.255.4:1::1400::00:21:59:c8:24:65/304
*                          Self                                  I

__default_evpn__.evpn.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
```

読者の方がお気付きのように、異なる 2 つの RT を使用して同じ MAC が 2 回アドバタイズされています。xe-0/0/34 では、Tenant_C と Tenant_D が同じ物理サーバー上にあり、異なる仮想マシンを実行しています。異なる VNI と RT でこれらの MAC アドレスをアドバタイズすることで、テナントの分離が維持されます。

コア 1 で、このタイプ 2 ルートが受信されて bgp.evpn.0 になっていることを確認します。

```
lab@core-1> show route receive-protocol bgp 1.255.255.4 evpn-mac-address
00:21:59:c8:24:65 extensive table bgp.evpn.0

<output omitted>

* 2:1.255.255.4:1::1300::00:21:59:c8:24:65/304 (2 entries, 0 announced)
     Import Accepted
     Route Distinguisher: 1.255.255.4:1
     Nexthop: 1.255.255.4
     AS path: 65402 I
     Communities: target:1:300 encapsulation0:0:0:0:vxlan

* 2:1.255.255.4:1::1400::00:21:59:c8:24:65/304 (2 entries, 0 announced)
     Import Accepted
     Route Distinguisher: 1.255.255.4:1
     Nexthop: 1.255.255.4
     AS path: 65402 I
     Communities: target:1:400 encapsulation0:0:0:0:vxlan
```

もう一度 00:21:59:c8:24:65 の 2 つのタイプ 2 ルートを受信しますが、RT が異なっています。リーフ 1 からの RD は 1.255.255.4:1 に設定されています。

次にコア 1 で、このタイプ 2 ルートが、bgp.evpn.0 テーブルから EVPN スイッチ インスタンスに正常にインポートされているかどうかを確認します。

まず、Tenant_C の仮想スイッチには、target:1:1300 を使用してアドバタイズされたと推定されるタイプ 2 ルートしかありません。

```
lab@core-1> show route table VS_VLAN300.evpn.0 evpn-mac-address
00:21:59:c8:24:65 | grep 00:21:59:c8:24:65

2:1.255.255.4:1::1300::00:21:59:c8:24:65/304

lab@core-1>
```

このタイプ 2 ルートの詳細を見てみましょう。

```
lab@core-1> show route table VS_VLAN300.evpn.0 evpn-mac-address 00:21:59:c8:24:65
extensive

<output omitted>
2:1.255.255.4:1::1300::00:21:59:c8:24:65/304 (2 entries, 1 announced)
   *BGP   Preference:170/-101
```

```
                Route Distinguisher: 1.255.255.4:1
                Next hop type: Indirect
                Address: 0x2cf479c
                Next-hop reference count: 76
                Source: 1.255.255.4
                Protocol next hop: 1.255.255.4
                Indirect next hop: 0x2 no-forward INH Session ID: 0x0
                State: <Secondary Active Ext>
                Local AS: 65400 Peer AS: 65402
                Age: 4:47  Metric2: 0
                Validation State: unverified
                Task: BGP_65402.1.255.255.4+179
                Announcement bits (1): 0-VS_VLAN300-evpn
                AS path: 65402 I
                Communities: target:1:300 encapsulation0:0:0:0:vxlan
                Import Accepted
                Localpref: 100
                Router ID: 1.255.255.4
                Primary Routing Table bgp.evpn.0
                Indirect next hops: 1
                        Protocol next hop: 1.255.255.4
                        Indirect next hop: 0x2 no-forward INH Session ID: 0x0
                        Indirect path forwarding next hops: 2
                                Next hop type: Router
                                Next hop: 1.0.0.1 via ge-1/0/0.0
                                Session Id: 0x140
                                Next hop: 1.0.0.3 via ge-1/0/1.0
                                Session Id: 0x141
                           1.255.255.4/32 Originating RIB: inet.0
                             Node path count: 1
                             Forwarding nexthops: 2
                              Nexthop: 1.0.0.1 via ge-1/0/0.0
        BGP     Preference: 170/-101
                Route Distinguisher: 1.255.255.4:1
                Next hop type: Indirect
                Address: 0x2cf479c
                Next-hop reference count: 76
                Source: 1.255.255.5
                Protocol next hop: 1.255.255.4
                Indirect next hop: 0x2 no-forward INH Session ID: 0x0
                State: <Secondary NotBest Ext>
                Inactive reason: Not Best in its group - Router ID
                Local AS: 65400 Peer AS: 65402
                Age: 4:47  Metric2: 0
                Validation State: unverified
                Task: BGP_65402.1.255.255.5+61407
                AS path: 65402 I
                Communities: target:1:300 encapsulation0:0:0:0:vxlan
                Import Accepted
```

```
                Localpref: 100
                Router ID: 1.255.255.5
                Primary Routing Table bgp.evpn.0
                Indirect next hops: 1
                        Protocol next hop: 1.255.255.4
                        Indirect next hop: 0x2 no-forward INH Session ID: 0x0
                        Indirect path forwarding next hops: 2
                                Next hop type: Router
                                Next hop: 1.0.0.1 via ge-1/0/0.0
                                Session Id: 0x140
                                Next hop: 1.0.0.3 via ge-1/0/1.0
                                Session Id: 0x141
                           1.255.255.4/32 Originating RIB: inet.0
                             Node path count: 1
                             Forwarding nexthops: 2
                              Nexthop: 1.0.0.1 via ge-1/0/0.0
```

コア 1 はコピーを 2 つ受信しています。最初のコピーはリーフ 1 からの直接のアドバタイズメントです（送信元：1.255.255.4）。2 番目のコピーはリーフ 1 から、リーフ 2、コア 1 に送信された間接的なアドバタイズメントです（送信元：1.255.255.5）。リーフ 2 で no-nexthop-change を設定しているため、1.255.255.4 という正しいプロトコル ネクストホップが維持されます。

L2ALD コピーは、次のように検証できます。

```
lab@core-1> show bridge mac-table instance VS_VLAN300

MAC flags       (S -static MAC, D -dynamic MAC, L -locally learned, C -Control MAC
    O -OVSDB MAC, SE -Statistics enabled, NM -Non configured MAC, R -Remote PE MAC)

Routing instance : VS_VLAN300
 Bridging domain : bd1300, VLAN : 300
   MAC                 MAC       Logical              Active
   address             flags     interface            source
   00:21:59:c8:24:65   D         vtep.32774           1.255.255.4
   00:21:59:c8:24:69   D         vtep.32783           1.255.255.5
```

上記の出力では、00:21:59:c8:24:65 は vtep.32774 を経由してリーフ 1 まで到達可能です。

次のコマンドを使用すると、カーネル レベルの転送テーブルのクエリーを実行できます。

```
lab@core-1> show route forwarding-table family bridge vpn VS_VLAN300
Routing table: VS_VLAN300.evpn-vxlan
VPLS:
Destination        Type RtRef Next hop           Type Index    NhRef Netif
default            perm     0                    dscd      540     1
vtep.32774         intf     0                    comp      699     7
vtep.32783         intf     0                    comp      714     5

Routing table: VS_VLAN300.evpn-vxlan
Bridging domain: bd1300.evpn-vxlan
VPLS:
Destination        Type RtRef Next hop           Type Index    NhRef Netif
00:21:59:c8:24:65/48 user     0                    comp      699     7
```

```
00:21:59:c8:24:69/48 user     0                    comp       714     5
0x30003/51           user     0                    comp       706     2
```

Tenant_C の MAC 00:21:59:c8:24:65 は、インデックス 699 を介して到達可能です。

インデックス 699（NH-Id）は、次のように正しい VNI 1300 およびリモート VTEP-ID である 1.255.255.4 に関連付けることもできます。

```
lab@core-1> show l2-learning vxlan-tunnel-end-point remote
Logical System Name       Id  SVTEP-IP         IFL   L3-Idx
<default>                 0   1.255.255.0      lo0.0    0
 RVTEP-IP         IFL-Idx   NH-Id
 1.255.255.4      351       697
    VNID          MC-Group-IP
    1100          0.0.0.0
 RVTEP-IP         IFL-Idx   NH-Id
 1.255.255.5      356       711
    VNID          MC-Group-IP
    1100          0.0.0.0
 RVTEP-IP         IFL-Idx   NH-Id
 1.255.255.4      352       698
    VNID          MC-Group-IP
    1200          0.0.0.0
 RVTEP-IP         IFL-Idx   NH-Id
 1.255.255.5      355       710
    VNID          MC-Group-IP
    1200          0.0.0.0
 RVTEP-IP         IFL-Idx   NH-Id
 1.255.255.4      353       699
    VNID          MC-Group-IP
    1300          0.0.0.0
 RVTEP-IP         IFL-Idx   NH-Id
 1.255.255.5      357       714
    VNID          MC-Group-IP
    1300          0.0.0.0
 RVTEP-IP         IFL-Idx   NH-Id
 1.255.255.4      354       700
    VNID          MC-Group-IP
    1400          0.0.0.0
 RVTEP-IP         IFL-Idx   NH-Id
 1.255.255.5      358       709
    VNID          MC-Group-IP
    1400          0.0.0.0
```

最後に、00:21:59:c8:24:65 が PFE ハードウェア内にプログラムされていることも確認できます。

```
# show l2 manager mac-table
<output omitted>
route table name    : VS_VLAN300.7
```

```
  mac counters
    maximum   count
    0           2
  mac table information
  mac address       BD     learn  Entry  entry  hal    hardware info
                    Index  vlan   Flags  ifl    ifl    pfe  mask  ifl
  ------------------------------------------------------------------------
  00:21:59:c8:24:65  4     0     0x0014 vtep.32774 vtep.32774  0   -D    src
unknown dest vtep.32774
  00:21:59:c8:24:69  4     0     0x0014 vtep.32783 vtep.32783  0   -D    src
unknown dest vtep.32783
 Displayed 2 entries for routing instance VS_VLAN300.7
```

マルチホームド CE デバイス

この例では、リーフ 1 およびリーフ 2 上のマルチホームド Tenant_B CE デバイスについて見てみます。マルチホーミングされているため、リーフ 1 とリーフ 2 はこの CE デバイス宛てのタイプ 1 とタイプ 2 の到達性をアドバタイズします。

```
lab@leaf-1> show ethernet-switching table vlan-id 200

MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent
static
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC, O -
ovsdb MAC)


Ethernet switching table : 4 entries, 4 learned
Routing instance : default-switch
   Vlan                MAC                 MAC     Logical             Active
   name                address             flags   interface           source
   v200                00:00:5e:00:01:01   DR       esi.1727
05:00:00:ff:78:00:00:04:b0:00
   v200                00:21:59:c8:24:64   DL       ae1.0
   v200                40:a6:77:9a:43:f0   D        vtep.32769         1.255.255.0
   v200                40:a6:77:9a:47:f0   D        vtep.32770         1.255.255.1

lab@leaf-2> show ethernet-switching table vlan-id 200

MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent
static
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC, O -
ovsdb MAC)

Ethernet switching table : 5 entries, 5 learned
Routing instance : default-switch
   Vlan                MAC                 MAC     Logical                Active
   name                address             flags   interface              source
   v200                00:00:5e:00:01:01   DR       esi.1727
05:00:00:ff:78:00:00:04:b0:00
   v200                00:21:59:c8:24:41   DL       ae1.0
```

```
   v200                00:21:59:c8:24:68   DL       ae1.0
   v200                40:a6:77:9a:43:f0   D        vtep.32770         1.255.255.0
   v200                40:a6:77:9a:47:f0   D        vtep.32769         1.255.255.1
```

このトポロジーの例で、00:21:59:c8:24:64 は、リーフ 1 に物理的に接続された Tenant _B ネットワーク インターフェイス カード（NIC）の物理 MAC を表します。00:21:59:c8:24:68 は、リーフ 2 に物理的に接続された Tenant _B NIC の物理 MAC を表します。00:21:59:c8:24:41 は仮想 MAC であるため、キャプチャ時点ではリーフ 2 でのみ学習されました。

前述のように ae1 は ESI 00:02:02:02:02:02:02:02:02:02 に属しています。

```
lab@leaf-1> show configuration interfaces ae1 esi
00:02:02:02:02:02:02:02:02:02;
all-active;
```

コア 1 では、Tenant_B の EVPN テーブルについてさまざまなことがわかります。

lab@core-1> show route table VS_VLAN200.evpn.0

```
VS_VLAN200.evpn.0: 18 destinations, 31 routes (18 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1:1.255.255.4:0::010101010101010101::FFFF:FFFF/304
                   *[BGP/170] 23:27:33, localpref 100, from 1.255.255.4
                      AS path: 65402 I, validation-state: unverified
                      to 1.0.0.1 via ge-1/0/0.0
                    > to 1.0.0.3 via ge-1/0/1.0
                    [BGP/170] 23:27:33, localpref 100, from 1.255.255.5
                      AS path: 65402 I, validation-state: unverified
                      to 1.0.0.1 via ge-1/0/0.0
                    > to 1.0.0.3 via ge-1/0/1.0
1:1.255.255.4:0::020202020202020202::FFFF:FFFF/304
                   *[BGP/170] 23:27:33, localpref 100, from 1.255.255.4
                      AS path: 65402 I, validation-state: unverified
                      to 1.0.0.1 via ge-1/0/0.0
                    > to 1.0.0.3 via ge-1/0/1.0
                    [BGP/170] 23:27:33, localpref 100, from 1.255.255.5
                      AS path: 65402 I, validation-state: unverified
                      to 1.0.0.1 via ge-1/0/0.0
                    > to 1.0.0.3 via ge-1/0/1.0
1:1.255.255.4:1::010101010101010101::0/304
                   *[BGP/170] 23:27:33, localpref 100, from 1.255.255.4
                      AS path: 65402 I, validation-state: unverified
                      to 1.0.0.1 via ge-1/0/0.0
                    > to 1.0.0.3 via ge-1/0/1.0
                    [BGP/170] 23:27:33, localpref 100, from 1.255.255.5
                      AS path: 65402 I, validation-state: unverified
                      to 1.0.0.1 via ge-1/0/0.0
                    > to 1.0.0.3 via ge-1/0/1.0
1:1.255.255.4:1::020202020202020202::0/304
                   *[BGP/170] 23:27:33, localpref 100, from 1.255.255.4
                      AS path: 65402 I, validation-state: unverified
```

```
                        to 1.0.0.1 via ge-1/0/0.0
                      > to 1.0.0.3 via ge-1/0/1.0
                      [BGP/170] 23:27:33, localpref 100, from 1.255.255.5
                        AS path: 65402 I, validation-state: unverified
                        to 1.0.0.1 via ge-1/0/0.0
                      > to 1.0.0.3 via ge-1/0/1.0
1:1.255.255.5:0::010101010101010101::FFFF:FFFF/304
                    *[BGP/170] 23:27:33, localpref 100, from 1.255.255.4
                        AS path: 65402 I, validation-state: unverified
                        to 1.0.0.1 via ge-1/0/0.0
                      > to 1.0.0.3 via ge-1/0/1.0
                      [BGP/170] 23:27:33, localpref 100, from 1.255.255.5
                        AS path: 65402 I, validation-state: unverified
                        to 1.0.0.1 via ge-1/0/0.0
                      > to 1.0.0.3 via ge-1/0/1.0
1:1.255.255.5:0::020202020202020202::FFFF:FFFF/304
                    *[BGP/170] 23:27:33, localpref 100, from 1.255.255.4
                        AS path: 65402 I, validation-state: unverified
                        to 1.0.0.1 via ge-1/0/0.0
                      > to 1.0.0.3 via ge-1/0/1.0
                      [BGP/170] 23:27:33, localpref 100, from 1.255.255.5
                        AS path: 65402 I, validation-state: unverified
                        to 1.0.0.1 via ge-1/0/0.0
                      > to 1.0.0.3 via ge-1/0/1.0
1:1.255.255.5:1::010101010101010101::0/304
                    *[BGP/170] 23:27:33, localpref 100, from 1.255.255.4
                        AS path: 65402 I, validation-state: unverified
                        to 1.0.0.1 via ge-1/0/0.0
                      > to 1.0.0.3 via ge-1/0/1.0
                      [BGP/170] 23:27:33, localpref 100, from 1.255.255.5
                        AS path: 65402 I, validation-state: unverified
                        to 1.0.0.1 via ge-1/0/0.0
                      > to 1.0.0.3 via ge-1/0/1.0
1:1.255.255.5:1::020202020202020202::0/304
                    *[BGP/170] 23:27:33, localpref 100, from 1.255.255.4
                        AS path: 65402 I, validation-state: unverified
                        to 1.0.0.1 via ge-1/0/0.0
                      > to 1.0.0.3 via ge-1/0/1.0
                      [BGP/170] 23:27:33, localpref 100, from 1.255.255.5
                        AS path: 65402 I, validation-state: unverified
                        to 1.0.0.1 via ge-1/0/0.0
                      > to 1.0.0.3 via ge-1/0/1.0
<output omitted>
2:1.255.255.4:1::1200::00:21:59:c8:24:64/304
                    *[BGP/170] 4d 10:47:09, localpref 100, from 1.255.255.4
                        AS path: 65402 I, validation-state: unverified
                      > to 1.0.0.1 via ge-1/0/0.0
                        to 1.0.0.3 via ge-1/0/1.0
                      [BGP/170] 1d 00:44:04, localpref 100, from 1.255.255.5
```

```
                        AS path: 65402 I, validation-state: unverified
                      > to 1.0.0.1 via ge-1/0/0.0
                        to 1.0.0.3 via ge-1/0/1.0
2:1.255.255.5:1::1200::00:21:59:c8:24:41/304
                     *[BGP/170] 00:14:09, localpref 100, from 1.255.255.4
                        AS path: 65402 I, validation-state: unverified
                      > to 1.0.0.1 via ge-1/0/0.0
                        to 1.0.0.3 via ge-1/0/1.0
                      [BGP/170] 00:14:09, localpref 100, from 1.255.255.5
                        AS path: 65402 I, validation-state: unverified
                      > to 1.0.0.1 via ge-1/0/0.0
                        to 1.0.0.3 via ge-1/0/1.0
2:1.255.255.5:1::1200::00:21:59:c8:24:68/304
                     *[BGP/170] 1d 00:44:04, localpref 100, from 1.255.255.4
                        AS path: 65402 I, validation-state: unverified
                        to 1.0.0.1 via ge-1/0/0.0
                      > to 1.0.0.3 via ge-1/0/1.0
                      [BGP/170] 4d 10:47:09, localpref 100, from 1.255.255.5
                        AS path: 65402 I, validation-state: unverified
                        to 1.0.0.1 via ge-1/0/0.0
                      > to 1.0.0.3 via ge-1/0/1.0

<output omitted>
```

1. 既に説明したように、この仮想スイッチ インスタンスにすべての ESI ルート（target:9999:9999）があり、タイプ 2 ルートは target:1200 のものしかありません。
2. Tenant_B CE デバイスについては、ESI 00:02:02:02:02:02:02:02:02:02 の異なる 4 つのルートを確認できます。

   a. `1:1.255.255.4:0::020202020202020202::FFFF:FFFF/304`

      i. これは、リーフ 1 から送信された各イーサネット セグメント AD のタイプ 1 EVPN ルートです。RD は、グローバル レベルのルーティングオプションから取得されています。

      ii. コア 1 は、元々はリーフ 1 から送信されたタイプ 1 ルートを、リーフ 1 とリーフ 2 の両方から受信します。

   b. `1:1.255.255.4:1::020202020202020202::0/304`

      i. これは、各 EVI の AD タイプ 1 EVPN ルートです。RD はルーティング インスタンスから、または QFX5100 の場合、スイッチ オプションから取得されます。

      ii. コア 1 は、元々はリーフ 1 から送信されたタイプ 1 ルートを、リーフ 1 とリーフ 2 の両方から受信します。

   c. `1:1.255.255.5:0::020202020202020202::FFFF:FFFF/304`

      i. これは、リーフ 2 から送信された各イーサネット セグメントの AD タイプ 1 EVPN ルートです。RD は、グローバル レベルのルーティングオプションから取得されています。

      ii. コア 1 は、元々はリーフ 2 から送信されたタイプ 1 ルートを、リーフ 2 とリーフ 1 の両方から受信します。

   d. `1:1.255.255.5:1::020202020202020202::0/304`

      i. これは、各 EVI の AD タイプ 1 EVPN ルートです。RD はルーティング インスタンスから、または QFX5100 の場合、スイッチ オプションから取得されます。

      ii. コア 1 は、元々はリーフ 2 から送信されたタイプ 1 ルートを、リーフ 2 とリーフ 1 の両方から受信します。

3. Tenant_B CE デバイスに属する 2 つの物理 MAC アドレスと 1 つの仮想 MAC アドレスに関して、期待どおりに送信された、タイプ 2 ルートを受信します。

コア 1 の L2ALD テーブルを以下に示します。

```
lab@core-1> show bridge mac-table instance VS_VLAN200
```

```
MAC flags       (S -static MAC, D -dynamic MAC, L -locally learned, C -Control MAC
    O -OVSDB MAC, SE -Statistics enabled, NM -Non configured MAC, R -Remote PE MAC)

Routing instance : VS_VLAN200
 Bridging domain : bd1200, VLAN : 200
   MAC                 MAC      Logical                Active
   address             flags    interface              source
   00:21:59:c8:24:41   DR       esi.703
00:02:02:02:02:02:02:02:02:02
   00:21:59:c8:24:64   DR       esi.703
00:02:02:02:02:02:02:02:02:02
   00:21:59:c8:24:68   DR       esi.703
00:02:02:02:02:02:02:02:02:02
```

コア1のカーネル転送テーブルは次のようになります。

```
lab@core-1> show route forwarding-table vpn VS_VLAN200
Routing table: VS_VLAN200.evpn-vxlan
VPLS:
Destination        Type RtRef Next hop           Type Index    NhRef Netif
default            perm     0                    dscd      536     1
vtep.32774         intf     0                    comp      702     6
vtep.32778         intf     0                    comp      720     6

Routing table: VS_VLAN200.evpn-vxlan
Bridging domain: bd1200.evpn-vxlan
VPLS:
Destination        Type RtRef Next hop           Type Index    NhRef Netif
00:21:59:c8:24:41/48 user     0                  indr  1048579     4
                                                 comp      703     2
00:21:59:c8:24:64/48 user     0                  indr  1048579     4
                                                 comp      703     2
00:21:59:c8:24:68/48 user     0                  indr  1048579     4
                                                 comp      703     2
0x30002/51         user     0                    comp      714     2
```

どの VTEP が ESI 00:02:02:02:02:02:02:02:02:02 への転送に使用されているかは、まだ結論付けることができません。このため、次のコマンドを実行します。

```
lab@core-1> show l2-learning vxlan-tunnel-end-point esi
<output omitted>
ESI                           RTT                      VLNBH INH     ESI-IFL
LOC-IFL   #RVTEPs
00:01:01:01:01:01:01:01:01:01 VS_VLAN200               704   1048580 esi.704
2
    RVTEP-IP             RVTEP-IFL      VENH     MASK-ID   FLAGS
    1.255.255.5          vtep.32778     720      1         2
    1.255.255.4          vtep.32774     702      0         2
<output omitted>
```

ここで、この ESI 内の MAC アドレスについて、VTEP インターフェイスの負荷がリーフ 1 とリーフ 2 両方に有効に分散されているのがわかります。したがって、リーフ 1 とリーフ 2 の設定がすべて有効なことを確認できます。

「esi.703」への参照は、複数の有効な個々の ECMP ネクストホップで構成された、内部「unilist」ネクストホップです。興味があれば、次のシェル レベル コマンドを実行して、これら個々のネクストホップを「unilist」に関連付けてみてください。

```
lab@core-1> start shell command "nhinfo -di 703"
<output omitted>
NHs in list: 720, 702,
<output omitted>
```

上記の出力から、インデックス 720 が リーフ 2 宛ての vtep.32778 に関連付けられ、インデックス 702 がリーフ 1 宛ての vtep.32774 に関連付けられていることを参照できます。

EVPN エニーキャスト ゲートウェイ

EVPN エニーキャスト ゲートウェイは、マルチホームド CE デバイスのシナリオと同様にトラブルシューティングします。このセクションでは、リーフ（QFX5100）からトラブルシューティングを行い、VNI 1100（リーフ 1 とリーフ 2 両方の VLAN100）上の、コア 1 とコア 2 両方のエニーキャスト ゲートウェイについて見ていきます。

注：本資料の公開時点での、初期リリースの QFX5100 搭載 EVPN+VXLAN は、**VNI 単位**でエニーキャスト ゲートウェイの負荷分散を実行します。つまり、指定された VNI について、QFX5100 リーフが単一の VTEP にトラフィックを転送するということです。次のリリースの QFX5100 搭載 EVPN+VXLAN では、マルチホームド CE デバイスの前述の例に示した MX シリーズ ルーターの動作と同じことが実行されます。実装後の QFX5100 は、同じ VNI の異なる VTEP（「コア」）に **5 タプル フロー**単位で負荷を分散させます。

```
lab@leaf-1> show route receive-protocol bgp 1.255.255.0

inet.0: 13 destinations, 18 routes (13 active, 0 holddown, 0 hidden)

:vxlan.inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)

bgp.evpn.0: 75 destinations, 123 routes (75 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop         MED     Lclpref    AS path
  1:1.255.255.0:0::050000ff780000044c00::FFFF:FFFF/304
*                         1.255.255.0                        65400 I
  1:1.255.255.0:0::050000ff78000004b000::FFFF:FFFF/304
*                         1.255.255.0                        65400 I
  1:1.255.255.0:0::050000ff780000051400::FFFF:FFFF/304
*                         1.255.255.0                        65400 I
  1:1.255.255.0:0::050000ff780000057800::FFFF:FFFF/304
*                         1.255.255.0                        65400 I
  2:1.255.255.0:300::1300::00:00:5e:00:01:01/304
*                         1.255.255.0                        65400 I
  2:1.255.255.0:300::1300::40:a6:77:9a:43:f0/304
*                         1.255.255.0                        65400 I
  2:1.255.255.0:100::1100::00:00:5e:00:01:01/304
*                         1.255.255.0                        65400 I
  2:1.255.255.0:100::1100::40:a6:77:9a:43:f0/304
*                         1.255.255.0                        65400 I
  2:1.255.255.0:400::1400::00:00:5e:00:01:01/304
*                         1.255.255.0                        65400 I
  2:1.255.255.0:400::1400::40:a6:77:9a:43:f0/304
*                         1.255.255.0                        65400 I
  2:1.255.255.0:200::1200::00:00:5e:00:01:01/304
*                         1.255.255.0                        65400 I
  2:1.255.255.0:200::1200::40:a6:77:9a:43:f0/304
```

```
*                          1.255.255.0                              65400 I
  2:1.255.255.0:300::1300::00:00:5e:00:01:01::10.10.10.1/304
*                          1.255.255.0                              65400 I
  2:1.255.255.0:300::1300::40:a6:77:9a:43:f0::10.10.10.2/304
*                          1.255.255.0                              65400 I
  2:1.255.255.0:100::1100::00:00:5e:00:01:01::100.0.0.1/304
*                          1.255.255.0                              65400 I
  2:1.255.255.0:100::1100::40:a6:77:9a:43:f0::100.0.0.2/304
*                          1.255.255.0                              65400 I
  2:1.255.255.0:400::1400::00:00:5e:00:01:01::10.10.10.1/304
*                          1.255.255.0                              65400 I
  2:1.255.255.0:400::1400::40:a6:77:9a:43:f0::10.10.10.2/304
*                          1.255.255.0                              65400 I
  2:1.255.255.0:200::1200::00:00:5e:00:01:01::200.0.0.1/304
*                          1.255.255.0                              65400 I
  2:1.255.255.0:200::1200::40:a6:77:9a:43:f0::200.0.0.2/304
*                          1.255.255.0                              65400 I
<output omitted>
```

リーフ 1 では、上記の出力から、エニーキャスト ゲートウェイについて、自動生成 ESI のタイプ 1 アドバタイズメントを受信することがわかります。エニーキャストおよび各ブリッジ ドメインの物理 MAC アドレス（a.b.c.2）も確認できます。

コア 2 からは、次のように同じタイプ 1、同じエニーキャストのタイプ 2、異なる物理タイプ 2（a.b.c.3）を確認できると予想されます。

```
lab@leaf-1> show route receive-protocol bgp 1.255.255.1

inet.0: 13 destinations, 18 routes (13 active, 0 holddown, 0 hidden)

:vxlan.inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)

bgp.evpn.0: 75 destinations, 123 routes (75 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop         MED     Lclpref    AS path
  1:1.255.255.1:0::050000ff780000044c00::FFFF:FFFF/304
*                          1.255.255.1                              65400 I
  1:1.255.255.1:0::050000ff78000004b000::FFFF:FFFF/304
*                          1.255.255.1                              65400 I
  1:1.255.255.1:0::050000ff780000051400::FFFF:FFFF/304
*                          1.255.255.1                              65400 I
  1:1.255.255.1:0::050000ff780000057800::FFFF:FFFF/304
*                          1.255.255.1                              65400 I
  2:1.255.255.1:300::1300::00:00:5e:00:01:01/304
*                          1.255.255.1                              65400 I
  2:1.255.255.1:300::1300::40:a6:77:9a:47:f0/304
*                          1.255.255.1                              65400 I
  2:1.255.255.1:100::1100::00:00:5e:00:01:01/304
*                          1.255.255.1                              65400 I
  2:1.255.255.1:100::1100::40:a6:77:9a:47:f0/304
*                          1.255.255.1                              65400 I
  2:1.255.255.1:400::1400::00:00:5e:00:01:01/304
*                          1.255.255.1                              65400 I
```

```
  2:1.255.255.1:400::1400::40:a6:77:9a:47:f0/304
*                         1.255.255.1                             65400 I
  2:1.255.255.1:200::1200::00:00:5e:00:01:01/304
*                         1.255.255.1                             65400 I
  2:1.255.255.1:200::1200::40:a6:77:9a:47:f0/304
*                         1.255.255.1                             65400 I
  2:1.255.255.1:300::1300::00:00:5e:00:01:01::10.10.10.1/304
*                         1.255.255.1                             65400 I
  2:1.255.255.1:300::1300::40:a6:77:9a:47:f0::10.10.10.3/304
*                         1.255.255.1                             65400 I
  2:1.255.255.1:100::1100::00:00:5e:00:01:01::100.0.0.1/304
*                         1.255.255.1                             65400 I
  2:1.255.255.1:100::1100::40:a6:77:9a:47:f0::100.0.0.3/304
*                         1.255.255.1                             65400 I
  2:1.255.255.1:400::1400::00:00:5e:00:01:01::10.10.10.1/304
*                         1.255.255.1                             65400 I
  2:1.255.255.1:400::1400::40:a6:77:9a:47:f0::10.10.10.3/304
*                         1.255.255.1                             65400 I
  2:1.255.255.1:200::1200::00:00:5e:00:01:01::200.0.0.1/304
*                         1.255.255.1                             65400 I
  2:1.255.255.1:200::1200::40:a6:77:9a:47:f0::200.0.0.3/304
<output omitted>
```

同様に、リーフ1の the default-switch.evpn.0 テーブルを表示すると、VNI 1110 の ESI を確認できます。

lab@leaf-1> show route table default-switch.evpn.0 evpn-esi-value 05:00:00:ff:78:00:00:04:4c:00

```
default-switch.evpn.0: 66 destinations, 114 routes (66 active, 0 holddown, 0
hidden)
+ = Active Route, - = Last Active, * = Both

1:1.255.255.0:0::050000ff780000044c00::FFFF:FFFF/304
                   *[BGP/170] 00:23:37, localpref 100, from 1.255.255.0
                      AS path: 65400 I, validation-state: unverified
                    > to 1.0.0.8 via xe-0/0/2.0
                      to 1.0.0.12 via xe-0/0/4.0
                    [BGP/170] 00:09:29, localpref 100, from 1.255.255.5
                      AS path: 65400 I, validation-state: unverified
                    > to 1.0.0.8 via xe-0/0/2.0
                      to 1.0.0.12 via xe-0/0/4.0
1:1.255.255.1:0::050000ff780000044c00::FFFF:FFFF/304
                   *[BGP/170] 00:23:33, localpref 100, from 1.255.255.1
                      AS path: 65400 I, validation-state: unverified
                    > to 1.0.0.8 via xe-0/0/2.0
                      to 1.0.0.12 via xe-0/0/4.0
                    [BGP/170] 00:09:29, localpref 100, from 1.255.255.5
                      AS path: 65400 I, validation-state: unverified
                    > to 1.0.0.8 via xe-0/0/2.0
                      to 1.0.0.12 via xe-0/0/4.0
```

このESIに固有な、対応するRDをもつアドバタイズメントを2つ（コア1から1つとコア2から1つ）受信します。リーフ1とリーフ2はIBGPのピアになっているため、元のプロトコルネクストホップが保持されたまま、リーフ2からの各ルートの追加コピーも確認できます。

以下はリーフ1のL2ALDテーブルです。

```
lab@leaf-1> show ethernet-switching table vlan-id 100

MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent
static
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC, O -
ovsdb MAC)

Ethernet switching table : 5 entries, 5 learned
Routing instance : default-switch
   Vlan                MAC                 MAC      Logical            Active
   name                address             flags    interface          source
   v100                00:00:5e:00:01:01   DR       esi.1727
05:00:00:ff:78:00:00:04:4c:00
   v100                00:21:59:c8:24:63   DL       ae0.0
   v100                00:21:59:c8:24:69   D        vtep.32771         1.255.255.5
   v100                40:a6:77:9a:43:f0   D        vtep.32769         1.255.255.0
   v100                40:a6:77:9a:47:f0   D        vtep.32770         1.255.255.1
```

以下はリーフ1のカーネル テーブルです。

```
lab@leaf-1> show route forwarding-table family ethernet-switching
<output omitted>

Routing table: default-switch.bridge
VPLS:
Destination        Type RtRef Next hop           Type Index    NhRef Netif
default            perm     0                    dscd     1688     1
vtep.32769         intf     0                    comp     1724    19
vtep.32770         intf     0                    comp     1737    15
vtep.32771         intf     0                    comp     1738     9
<output omitted>

Routing table: default-switch.bridge
Bridging domain: v100.bridge
VPLS:
Destination        Type RtRef Next hop           Type Index    NhRef Netif
00:00:5e:00:01:01/48 user     0                  comp     1724    19
<output omitted>
```

ここでは、リーフ1は、VNI 1100のエニーキャスト ゲートウェイのすべてのトラフィックを転送するために、vtep.32769に関連付けられたインデックス1724を選択しているのがわかります。

vtep.32769とは何でしょうか。

```
lab@leaf-1> show ethernet-switching vxlan-tunnel-end-point esi
<output omitted>
05:00:00:ff:78:00:00:04:4c:00 default-switch           1727  131074  esi.1727
2
```

```
RVTEP-IP             RVTEP-IFL      VENH     MASK-ID   FLAGS
1.255.255.1          vtep.32770     1737     1         2
1.255.255.0          vtep.32769     1724     0         2
```

上記の出力から、コア 1 への VTEP であることがわかります。これもまた、QFX5100 での EVPN+VXLAN Junos オペレーティング システムの初期リリースにおける動作です。将来のリリースでは、前の例で説明した MX シリーズ ルーターと同様に unilist で負荷分散します。

全設定

リーフ1

```
set system host-name leaf-1
set chassis aggregated-devices ethernet device-count 2
set interfaces xe-0/0/2 unit 0 family inet address 1.0.0.9/31
set interfaces xe-0/0/4 unit 0 family inet address 1.0.0.13/31
set interfaces xe-0/0/32 ether-options 802.3ad ae0
set interfaces xe-0/0/33 ether-options 802.3ad ae1
set interfaces xe-0/0/34 unit 0 family ethernet-switching interface-mode trunk
set interfaces xe-0/0/34 unit 0 family ethernet-switching vlan members v300
set interfaces xe-0/0/34 unit 0 family ethernet-switching vlan members v400
set interfaces ae0 esi 00:01:01:01:01:01:01:01:01:01
set interfaces ae0 esi all-active
set interfaces ae0 aggregated-ether-options lacp active
set interfaces ae0 aggregated-ether-options lacp system-id 00:00:00:01:01:01
set interfaces ae0 unit 0 family ethernet-switching interface-mode access
set interfaces ae0 unit 0 family ethernet-switching vlan members v100
set interfaces ae1 esi 00:02:02:02:02:02:02:02:02:02
set interfaces ae1 esi all-active
set interfaces ae1 aggregated-ether-options lacp active
set interfaces ae1 aggregated-ether-options lacp system-id 00:00:00:01:01:01
set interfaces ae1 unit 0 family ethernet-switching interface-mode access
set interfaces ae1 unit 0 family ethernet-switching vlan members v200
set interfaces lo0 unit 0 family inet address 1.255.255.4/32
set routing-options router-id 1.255.255.4
set routing-options autonomous-system 65402
set routing-options forwarding-table export load-balance
set routing-options forwarding-table ecmp-fast-reroute
set protocols bgp group underlay type external
set protocols bgp group underlay advertise-peer-as
set protocols bgp group underlay family inet unicast loops 2
set protocols bgp group underlay export lo0
set protocols bgp group underlay peer-as 65401
set protocols bgp group underlay multipath
set protocols bgp group underlay neighbor 1.0.0.8 description spine-1
set protocols bgp group underlay neighbor 1.0.0.12 description spine-2
set protocols bgp group EVPN_VXLAN_CORE type external
set protocols bgp group EVPN_VXLAN_CORE multihop ttl 255
set protocols bgp group EVPN_VXLAN_CORE multihop no-nexthop-change
set protocols bgp group EVPN_VXLAN_CORE local-address 1.255.255.4
set protocols bgp group EVPN_VXLAN_CORE family evpn signaling
set protocols bgp group EVPN_VXLAN_CORE peer-as 65400
```

```
set protocols bgp group EVPN_VXLAN_CORE local-as 65403
set protocols bgp group EVPN_VXLAN_CORE neighbor 1.255.255.0 description core-1
set protocols bgp group EVPN_VXLAN_CORE neighbor 1.255.255.1 description core-2
set protocols bgp group EVPN_VXLAN_LEAF type internal
set protocols bgp group EVPN_VXLAN_LEAF local-address 1.255.255.4
set protocols bgp group EVPN_VXLAN_LEAF family evpn signaling
set protocols bgp group EVPN_VXLAN_LEAF export LEAF-PREPEND
set protocols bgp group EVPN_VXLAN_LEAF neighbor 1.255.255.5 description leaf-2
set protocols evpn encapsulation vxlan
set protocols evpn extended-vni-list 1100
set protocols evpn extended-vni-list 1200
set protocols evpn extended-vni-list 1300
set protocols evpn extended-vni-list 1400
set protocols evpn multicast-mode ingress-replication
set protocols evpn vni-routing-options vni 1100 vrf-target export target:1:100
set protocols evpn vni-routing-options vni 1200 vrf-target export target:1:200
set protocols evpn vni-routing-options vni 1300 vrf-target export target:1:300
set protocols evpn vni-routing-options vni 1400 vrf-target export target:1:400
set protocols l2-learning traceoptions file l2ald.log
set protocols l2-learning traceoptions file size 10m
set protocols l2-learning traceoptions level all
set protocols l2-learning traceoptions flag all
set protocols lldp port-id-subtype interface-name
set protocols lldp interface all
set policy-options policy-statement LEAF-PREPEND then as-path-prepend 65402
set policy-options policy-statement lo0 from family inet
set policy-options policy-statement lo0 from protocol direct
set policy-options policy-statement lo0 from route-filter 0.0.0.0/0 prefix-length-
range /32-/32
set policy-options policy-statement lo0 then accept
set policy-options policy-statement load-balance term 1 then load-balance per-
packet
set policy-options policy-statement vrf-imp term t1 from community com100
set policy-options policy-statement vrf-imp term t1 then accept
set policy-options policy-statement vrf-imp term t2 from community com200
set policy-options policy-statement vrf-imp term t2 then accept
set policy-options policy-statement vrf-imp term t3 from community com300
set policy-options policy-statement vrf-imp term t3 then accept
set policy-options policy-statement vrf-imp term t4 from community com400
set policy-options policy-statement vrf-imp term t4 then accept
set policy-options policy-statement vrf-imp term t5 then reject
set policy-options community com100 members target:1:100
set policy-options community com200 members target:1:200
set policy-options community com300 members target:1:300
set policy-options community com400 members target:1:400
set switch-options vtep-source-interface lo0.0
set switch-options route-distinguisher 1.255.255.4:1
set switch-options vrf-import vrf-imp
set switch-options vrf-target target:9999:9999
```

```
set vlans v100 vlan-id 100
set vlans v100 vxlan vni 1100
set vlans v100 vxlan ingress-node-replication
set vlans v200 vlan-id 200
set vlans v200 vxlan vni 1200
set vlans v200 vxlan ingress-node-replication
set vlans v300 vlan-id 300
set vlans v300 vxlan vni 1300
set vlans v300 vxlan ingress-node-replication
set vlans v400 vlan-id 400
set vlans v400 vxlan vni 1400
set vlans v400 vxlan ingress-node-replication
```

リーフ 2

```
set system host-name leaf-2
set chassis aggregated-devices ethernet device-count 2
set interfaces xe-0/0/3 unit 0 family inet address 1.0.0.11/31
set interfaces xe-0/0/5 unit 0 family inet address 1.0.0.15/31
set interfaces xe-0/0/36 ether-options 802.3ad ae0
set interfaces xe-0/0/37 ether-options 802.3ad ae1
set interfaces xe-0/0/38 unit 0 family ethernet-switching interface-mode trunk
set interfaces xe-0/0/38 unit 0 family ethernet-switching vlan members v300
set interfaces xe-0/0/38 unit 0 family ethernet-switching vlan members v100
set interfaces ae0 esi 00:01:01:01:01:01:01:01:01:01
set interfaces ae0 esi all-active
set interfaces ae0 aggregated-ether-options lacp passive
set interfaces ae0 aggregated-ether-options lacp system-id 00:00:00:01:01:01
set interfaces ae0 unit 0 family ethernet-switching interface-mode access
set interfaces ae0 unit 0 family ethernet-switching vlan members v100
set interfaces ae1 esi 00:02:02:02:02:02:02:02:02:02
set interfaces ae1 esi all-active
set interfaces ae1 aggregated-ether-options lacp passive
set interfaces ae1 aggregated-ether-options lacp system-id 00:00:00:01:01:01
set interfaces ae1 unit 0 family ethernet-switching interface-mode access
set interfaces ae1 unit 0 family ethernet-switching vlan members v200
set interfaces lo0 unit 0 family inet address 1.255.255.5/32
set routing-options router-id 1.255.255.5
set routing-options autonomous-system 65402
set routing-options forwarding-table export load-balance
set routing-options forwarding-table ecmp-fast-reroute
set protocols bgp group underlay type external
set protocols bgp group underlay advertise-peer-as
set protocols bgp group underlay family inet unicast loops 2
set protocols bgp group underlay export lo0
set protocols bgp group underlay peer-as 65401
set protocols bgp group underlay multipath
set protocols bgp group underlay neighbor 1.0.0.10 description spine-1
set protocols bgp group underlay neighbor 1.0.0.14 description spine-2
set protocols bgp group EVPN_VXLAN_CORE type external
```

```
set protocols bgp group EVPN_VXLAN_CORE multihop ttl 255
set protocols bgp group EVPN_VXLAN_CORE multihop no-nexthop-change
set protocols bgp group EVPN_VXLAN_CORE local-address 1.255.255.5
set protocols bgp group EVPN_VXLAN_CORE family evpn signaling
set protocols bgp group EVPN_VXLAN_CORE peer-as 65400
set protocols bgp group EVPN_VXLAN_CORE neighbor 1.255.255.0 description core-1
set protocols bgp group EVPN_VXLAN_CORE neighbor 1.255.255.1 description core-2
set protocols bgp group EVPN_VXLAN_LEAF type internal
set protocols bgp group EVPN_VXLAN_LEAF local-address 1.255.255.5
set protocols bgp group EVPN_VXLAN_LEAF family evpn signaling
set protocols bgp group EVPN_VXLAN_LEAF export LEAF-PREPEND
set protocols bgp group EVPN_VXLAN_LEAF neighbor 1.255.255.4 description leaf-1
set protocols evpn encapsulation vxlan
set protocols evpn extended-vni-list 1100
set protocols evpn extended-vni-list 1200
set protocols evpn extended-vni-list 1300
set protocols evpn extended-vni-list 1400
set protocols evpn multicast-mode ingress-replication
set protocols evpn vni-routing-options vni 1100 vrf-target export target:1:100
set protocols evpn vni-routing-options vni 1200 vrf-target export target:1:200
set protocols evpn vni-routing-options vni 1300 vrf-target export target:1:300
set protocols evpn vni-routing-options vni 1400 vrf-target export target:1:400
set protocols lldp port-id-subtype interface-name
set protocols lldp interface all
set policy-options policy-statement LEAF-PREPEND then as-path-prepend 65402
set policy-options policy-statement lo0 from family inet
set policy-options policy-statement lo0 from protocol direct
set policy-options policy-statement lo0 from route-filter 0.0.0.0/0 prefix-length-
range /32-/32
set policy-options policy-statement lo0 then accept
set policy-options policy-statement load-balance term 1 then load-balance per-
packet
set policy-options policy-statement vrf-imp term t1 from community com100
set policy-options policy-statement vrf-imp term t1 then accept
set policy-options policy-statement vrf-imp term t2 from community com200
set policy-options policy-statement vrf-imp term t2 then accept
set policy-options policy-statement vrf-imp term t3 from community com300
set policy-options policy-statement vrf-imp term t3 then accept
set policy-options policy-statement vrf-imp term t4 from community com400
set policy-options policy-statement vrf-imp term t4 then accept
set policy-options policy-statement vrf-imp term t5 then reject
set policy-options community com100 members target:1:100
set policy-options community com200 members target:1:200
set policy-options community com300 members target:1:300
set policy-options community com400 members target:1:400
set switch-options vtep-source-interface lo0.0
set switch-options route-distinguisher 1.255.255.5:1
set switch-options vrf-import vrf-imp
set switch-options vrf-target target:9999:9999
```

```
set vlans v100 vlan-id 100
set vlans v100 vxlan vni 1100
set vlans v100 vxlan ingress-node-replication
set vlans v200 vlan-id 200
set vlans v200 vxlan vni 1200
set vlans v200 vxlan ingress-node-replication
set vlans v300 vlan-id 300
set vlans v300 vxlan vni 1300
set vlans v300 vxlan ingress-node-replication
set vlans v400 vlan-id 400
set vlans v400 vxlan vni 1400
set vlans v400 vxlan ingress-node-replication
```

スパイン1

```
set system host-name spine-1
set interfaces xe-0/0/0 unit 0 family inet address 1.0.0.1/31
set interfaces xe-0/0/1 unit 0 family inet address 1.0.0.5/31
set interfaces xe-0/0/2 unit 0 family inet address 1.0.0.8/31
set interfaces xe-0/0/3 unit 0 family inet address 1.0.0.10/31
set interfaces lo0 unit 0 family inet address 1.255.255.2/32
set routing-options router-id 1.255.255.2
set routing-options autonomous-system 65401
set routing-options forwarding-table export load-balance
set routing-options forwarding-table ecmp-fast-reroute
set protocols bgp group underlay-leaf type external
set protocols bgp group underlay-leaf advertise-peer-as
set protocols bgp group underlay-leaf family inet unicast loops 2
set protocols bgp group underlay-leaf export lo0
set protocols bgp group underlay-leaf peer-as 65402
set protocols bgp group underlay-leaf multipath
set protocols bgp group underlay-leaf neighbor 1.0.0.9 description leaf-1
set protocols bgp group underlay-leaf neighbor 1.0.0.11 description leaf-2
set protocols bgp group underlay-core type external
set protocols bgp group underlay-core advertise-peer-as
set protocols bgp group underlay-core family inet unicast loops 2
set protocols bgp group underlay-core export lo0
set protocols bgp group underlay-core peer-as 65400
set protocols bgp group underlay-core multipath
set protocols bgp group underlay-core neighbor 1.0.0.0 description core-1
set protocols bgp group underlay-core neighbor 1.0.0.4 description core-2
set protocols lldp port-id-subtype interface-name
set protocols lldp interface all
set policy-options policy-statement lo0 from family inet
set policy-options policy-statement lo0 from protocol direct
set policy-options policy-statement lo0 from route-filter 0.0.0.0/0 prefix-length-
range /32-/32
set policy-options policy-statement lo0 then accept
set policy-options policy-statement load-balance term 1 then load-balance per-
packet
```

スパイン 2

```
set system host-name spine-2
set interfaces xe-0/0/0 unit 0 family inet address 1.0.0.3/31
set interfaces xe-0/0/1 unit 0 family inet address 1.0.0.7/31
set interfaces xe-0/0/4 unit 0 family inet address 1.0.0.12/31
set interfaces xe-0/0/5 unit 0 family inet address 1.0.0.14/31
set interfaces lo0 unit 0 family inet address 1.255.255.3/32
set routing-options router-id 1.255.255.3
set routing-options autonomous-system 65401
set routing-options forwarding-table export load-balance
set routing-options forwarding-table ecmp-fast-reroute
set protocols bgp group underlay-leaf type external
set protocols bgp group underlay-leaf advertise-peer-as
set protocols bgp group underlay-leaf family inet unicast loops 2
set protocols bgp group underlay-leaf export lo0
set protocols bgp group underlay-leaf peer-as 65402
set protocols bgp group underlay-leaf multipath
set protocols bgp group underlay-leaf neighbor 1.0.0.13 description leaf-1
set protocols bgp group underlay-leaf neighbor 1.0.0.15 description leaf-2
set protocols bgp group underlay-core type external
set protocols bgp group underlay-core advertise-peer-as
set protocols bgp group underlay-core family inet unicast loops 2
set protocols bgp group underlay-core export lo0
set protocols bgp group underlay-core peer-as 65400
set protocols bgp group underlay-core multipath
set protocols bgp group underlay-core neighbor 1.0.0.2 description core-1
set protocols bgp group underlay-core neighbor 1.0.0.6 description core-2
set protocols lldp port-id-subtype interface-name
set protocols lldp interface all
set policy-options policy-statement lo0 from family inet
set policy-options policy-statement lo0 from protocol direct
set policy-options policy-statement lo0 from route-filter 0.0.0.0/0 prefix-length-
range /32-/32
set policy-options policy-statement lo0 then accept
set policy-options policy-statement load-balance term 1 then load-balance per-
packet
```

コア 1

```
set system host-name core-1
set interfaces ge-1/0/0 unit 0 family inet address 1.0.0.0/31
set interfaces ge-1/0/1 unit 0 family inet address 1.0.0.2/31
set interfaces irb unit 1100 family inet address 100.0.0.2/24 virtual-gateway-
address 100.0.0.1
set interfaces irb unit 1200 family inet address 200.0.0.2/24 virtual-gateway-
address 200.0.0.1
set interfaces irb unit 1300 family inet address 10.10.10.2/24 virtual-gateway-
address 10.10.10.1
set interfaces irb unit 1400 family inet address 10.10.10.2/24 virtual-gateway-
address 10.10.10.1
```

```
set interfaces lo0 unit 0 family inet address 1.255.255.0/32
set routing-options router-id 1.255.255.0
set routing-options autonomous-system 65400
set routing-options forwarding-table export load-balance
set routing-options forwarding-table ecmp-fast-reroute
set protocols bgp group underlay type external
set protocols bgp group underlay advertise-peer-as
set protocols bgp group underlay family inet unicast loops 2
set protocols bgp group underlay export lo0
set protocols bgp group underlay peer-as 65401
set protocols bgp group underlay multipath
set protocols bgp group underlay neighbor 1.0.0.1 description spine-1
set protocols bgp group underlay neighbor 1.0.0.3 description spine-2
set protocols bgp group EVPN_VXLAN type external
set protocols bgp group EVPN_VXLAN local-address 1.255.255.0
set protocols bgp group EVPN_VXLAN family evpn signaling
set protocols bgp group EVPN_VXLAN peer-as 65402
set protocols bgp group EVPN_VXLAN multipath
set protocols bgp group EVPN_VXLAN neighbor 1.255.255.4 description leaf-1
set protocols bgp group EVPN_VXLAN neighbor 1.255.255.4 multihop ttl 255
set protocols bgp group EVPN_VXLAN neighbor 1.255.255.5 description leaf-2
set protocols bgp group EVPN_VXLAN neighbor 1.255.255.5 multihop ttl 255
set protocols bgp group EVPN_VXLAN_TEMP type external
set protocols bgp group EVPN_VXLAN_TEMP local-address 1.255.255.0
set protocols bgp group EVPN_VXLAN_TEMP family evpn signaling
set protocols bgp group EVPN_VXLAN_TEMP peer-as 65403
set protocols bgp group EVPN_VXLAN_TEMP multipath
set protocols bgp group EVPN_VXLAN_TEMP neighbor 1.255.255.4 description leaf-1
set protocols bgp group EVPN_VXLAN_TEMP neighbor 1.255.255.4 multihop ttl 255
set protocols bgp group EVPN_VXLAN_TEMP neighbor 1.255.255.5 description leaf-2
set protocols bgp group EVPN_VXLAN_TEMP neighbor 1.255.255.5 multihop ttl 255
set protocols lldp port-id-subtype interface-name
set protocols lldp interface all
set policy-options policy-statement VS_VLAN100_IMP term ESI from community comm-
leaf_esi
set policy-options policy-statement VS_VLAN100_IMP term ESI then accept
set policy-options policy-statement VS_VLAN100_IMP term VS_VLAN100 from community
comm-VS_VLAN100
set policy-options policy-statement VS_VLAN100_IMP term VS_VLAN100 then accept
set policy-options policy-statement VS_VLAN200_IMP term ESI from community comm-
leaf_esi
set policy-options policy-statement VS_VLAN200_IMP term ESI then accept
set policy-options policy-statement VS_VLAN200_IMP term VS_VLAN200 from community
comm-VS_VLAN200
set policy-options policy-statement VS_VLAN200_IMP term VS_VLAN200 then accept
set policy-options policy-statement VS_VLAN300_IMP term ESI from community comm-
leaf_esi
set policy-options policy-statement VS_VLAN300_IMP term ESI then accept
set policy-options policy-statement VS_VLAN300_IMP term VS_VLAN300 from community
```

```
comm-VS_VLAN300
set policy-options policy-statement VS_VLAN300_IMP term VS_VLAN300 then accept
set policy-options policy-statement VS_VLAN400_IMP term ESI from community comm-
leaf_esi
set policy-options policy-statement VS_VLAN400_IMP term ESI then accept
set policy-options policy-statement VS_VLAN400_IMP term VS_VLAN400 from community
comm-VS_VLAN400
set policy-options policy-statement VS_VLAN400_IMP term VS_VLAN400 then accept
set policy-options policy-statement lo0 from family inet
set policy-options policy-statement lo0 from protocol direct
set policy-options policy-statement lo0 from route-filter 0.0.0.0/0 prefix-length-
range /32-/32
set policy-options policy-statement lo0 then accept
set policy-options policy-statement load-balance term 1 then load-balance per-
packet
set policy-options community comm-VS_VLAN100 members target:1:100
set policy-options community comm-VS_VLAN200 members target:1:200
set policy-options community comm-VS_VLAN300 members target:1:300
set policy-options community comm-VS_VLAN400 members target:1:400
set policy-options community comm-leaf_esi members target:9999:9999
set routing-instances VRF_Tenant_A instance-type vrf
set routing-instances VRF_Tenant_A interface irb.1100
set routing-instances VRF_Tenant_A route-distinguisher 1.255.255.0:1100
set routing-instances VRF_Tenant_A vrf-target target:10:100
set routing-instances VRF_Tenant_A routing-options auto-export
set routing-instances VRF_Tenant_B instance-type vrf
set routing-instances VRF_Tenant_B interface irb.1200
set routing-instances VRF_Tenant_B route-distinguisher 1.255.255.0:1200
set routing-instances VRF_Tenant_B vrf-target target:10:200
set routing-instances VRF_Tenant_C instance-type vrf
set routing-instances VRF_Tenant_C interface irb.1300
set routing-instances VRF_Tenant_C route-distinguisher 1.255.255.0:1300
set routing-instances VRF_Tenant_C vrf-target target:10:300
set routing-instances VRF_Tenant_D instance-type vrf
set routing-instances VRF_Tenant_D interface irb.1400
set routing-instances VRF_Tenant_D route-distinguisher 1.255.255.0:1400
set routing-instances VRF_Tenant_D vrf-target target:10:400
set routing-instances VS_VLAN100 vtep-source-interface lo0.0
set routing-instances VS_VLAN100 instance-type virtual-switch
set routing-instances VS_VLAN100 route-distinguisher 1.255.255.0:100
set routing-instances VS_VLAN100 vrf-import VS_VLAN100_IMP
set routing-instances VS_VLAN100 vrf-target target:1:100
set routing-instances VS_VLAN100 protocols evpn encapsulation vxlan
set routing-instances VS_VLAN100 protocols evpn extended-vni-list 1100
set routing-instances VS_VLAN100 protocols evpn multicast-mode ingress-replication
set routing-instances VS_VLAN100 bridge-domains bd1100 vlan-id 100
set routing-instances VS_VLAN100 bridge-domains bd1100 routing-interface irb.1100
set routing-instances VS_VLAN100 bridge-domains bd1100 vxlan vni 1100
set routing-instances VS_VLAN100 bridge-domains bd1100 vxlan ingress-node-
```

```
replication
set routing-instances VS_VLAN200 vtep-source-interface lo0.0
set routing-instances VS_VLAN200 instance-type virtual-switch
set routing-instances VS_VLAN200 route-distinguisher 1.255.255.0:200
set routing-instances VS_VLAN200 vrf-import VS_VLAN200_IMP
set routing-instances VS_VLAN200 vrf-target target:1:200
set routing-instances VS_VLAN200 protocols evpn encapsulation vxlan
set routing-instances VS_VLAN200 protocols evpn extended-vni-list 1200
set routing-instances VS_VLAN200 protocols evpn multicast-mode ingress-replication
set routing-instances VS_VLAN200 bridge-domains bd1200 vlan-id 200
set routing-instances VS_VLAN200 bridge-domains bd1200 routing-interface irb.1200
set routing-instances VS_VLAN200 bridge-domains bd1200 vxlan vni 1200
set routing-instances VS_VLAN200 bridge-domains bd1200 vxlan ingress-node-
replication
set routing-instances VS_VLAN300 vtep-source-interface lo0.0
set routing-instances VS_VLAN300 instance-type virtual-switch
set routing-instances VS_VLAN300 route-distinguisher 1.255.255.0:300
set routing-instances VS_VLAN300 vrf-import VS_VLAN300_IMP
set routing-instances VS_VLAN300 vrf-target target:1:300
set routing-instances VS_VLAN300 protocols evpn encapsulation vxlan
set routing-instances VS_VLAN300 protocols evpn extended-vni-list 1300
set routing-instances VS_VLAN300 protocols evpn multicast-mode ingress-replication
set routing-instances VS_VLAN300 bridge-domains bd1300 vlan-id 300
set routing-instances VS_VLAN300 bridge-domains bd1300 routing-interface irb.1300
set routing-instances VS_VLAN300 bridge-domains bd1300 vxlan vni 1300
set routing-instances VS_VLAN300 bridge-domains bd1300 vxlan ingress-node-
replication
set routing-instances VS_VLAN400 vtep-source-interface lo0.0
set routing-instances VS_VLAN400 instance-type virtual-switch
set routing-instances VS_VLAN400 route-distinguisher 1.255.255.0:400
set routing-instances VS_VLAN400 vrf-import VS_VLAN400_IMP
set routing-instances VS_VLAN400 vrf-target target:1:400
set routing-instances VS_VLAN400 protocols evpn encapsulation vxlan
set routing-instances VS_VLAN400 protocols evpn extended-vni-list 1400
set routing-instances VS_VLAN400 protocols evpn multicast-mode ingress-replication
set routing-instances VS_VLAN400 bridge-domains bd1400 vlan-id 400
set routing-instances VS_VLAN400 bridge-domains bd1400 routing-interface irb.1400
set routing-instances VS_VLAN400 bridge-domains bd1400 vxlan vni 1400
set routing-instances VS_VLAN400 bridge-domains bd1400 vxlan ingress-node-
replication
```

コア 2

```
set system host-name core-2
set chassis network-services enhanced-ip
set interfaces ge-1/0/0 unit 0 family inet address 1.0.0.4/31
set interfaces ge-1/0/1 unit 0 family inet address 1.0.0.6/31
set interfaces irb unit 1100 family inet address 100.0.0.3/24 virtual-gateway-
address 100.0.0.1
set interfaces irb unit 1200 family inet address 200.0.0.3/24 virtual-gateway-
```

```
address 200.0.0.1
set interfaces irb unit 1300 family inet address 10.10.10.3/24 virtual-gateway-
address 10.10.10.1
set interfaces irb unit 1400 family inet address 10.10.10.3/24 virtual-gateway-
address 10.10.10.1
set interfaces lo0 unit 0 family inet address 1.255.255.1/32
set routing-options router-id 1.255.255.1
set routing-options autonomous-system 65400
set routing-options forwarding-table export load-balance
set routing-options forwarding-table ecmp-fast-reroute
set protocols bgp group underlay type external
set protocols bgp group underlay advertise-peer-as
set protocols bgp group underlay family inet unicast loops 2
set protocols bgp group underlay export lo0
set protocols bgp group underlay peer-as 65401
set protocols bgp group underlay multipath
set protocols bgp group underlay neighbor 1.0.0.5 description spine-1
set protocols bgp group underlay neighbor 1.0.0.7 description spine-2
set protocols bgp group EVPN_VXLAN type external
set protocols bgp group EVPN_VXLAN local-address 1.255.255.1
set protocols bgp group EVPN_VXLAN family evpn signaling
set protocols bgp group EVPN_VXLAN peer-as 65402
set protocols bgp group EVPN_VXLAN multipath
set protocols bgp group EVPN_VXLAN neighbor 1.255.255.4 description leaf-1
set protocols bgp group EVPN_VXLAN neighbor 1.255.255.4 multihop ttl 255
set protocols bgp group EVPN_VXLAN neighbor 1.255.255.5 description leaf-2
set protocols bgp group EVPN_VXLAN neighbor 1.255.255.5 multihop ttl 255
set protocols lldp port-id-subtype interface-name
set protocols lldp interface all
set policy-options policy-statement VS_VLAN100_IMP term ESI from community comm-
leaf_esi
set policy-options policy-statement VS_VLAN100_IMP term ESI then accept
set policy-options policy-statement VS_VLAN100_IMP term VS_VLAN100 from community
comm-VS_VLAN100
set policy-options policy-statement VS_VLAN100_IMP term VS_VLAN100 then accept
set policy-options policy-statement VS_VLAN200_IMP term ESI from community comm-
leaf_esi
set policy-options policy-statement VS_VLAN200_IMP term ESI then accept
set policy-options policy-statement VS_VLAN200_IMP term VS_VLAN200 from community
comm-VS_VLAN200
set policy-options policy-statement VS_VLAN200_IMP term VS_VLAN200 then accept
set policy-options policy-statement VS_VLAN300_IMP term ESI from community comm-
leaf_esi
set policy-options policy-statement VS_VLAN300_IMP term ESI then accept
set policy-options policy-statement VS_VLAN300_IMP term VS_VLAN300 from community
comm-VS_VLAN300
set policy-options policy-statement VS_VLAN300_IMP term VS_VLAN300 then accept
set policy-options policy-statement VS_VLAN400_IMP term ESI from community comm-
leaf_esi
```

```
set policy-options policy-statement VS_VLAN400_IMP term ESI then accept
set policy-options policy-statement VS_VLAN400_IMP term VS_VLAN400 from community
comm-VS_VLAN400
set policy-options policy-statement VS_VLAN400_IMP term VS_VLAN400 then accept
set policy-options policy-statement lo0 from family inet
set policy-options policy-statement lo0 from protocol direct
set policy-options policy-statement lo0 from route-filter 0.0.0.0/0 prefix-length-
range /32-/32
set policy-options policy-statement lo0 then accept
set policy-options policy-statement load-balance term 1 then load-balance per-
packet
set policy-options community comm-VS_VLAN100 members target:1:100
set policy-options community comm-VS_VLAN200 members target:1:200
set policy-options community comm-VS_VLAN300 members target:1:300
set policy-options community comm-VS_VLAN400 members target:1:400
set policy-options community comm-leaf_esi members target:9999:9999
set routing-instances VRF_Tenant_A instance-type vrf
set routing-instances VRF_Tenant_A interface irb.1100
set routing-instances VRF_Tenant_A route-distinguisher 1.255.255.1:1100
set routing-instances VRF_Tenant_A vrf-target target:10:100
set routing-instances VRF_Tenant_A routing-options auto-export
set routing-instances VRF_Tenant_B instance-type vrf
set routing-instances VRF_Tenant_B interface irb.1200
set routing-instances VRF_Tenant_B route-distinguisher 1.255.255.1:1200
set routing-instances VRF_Tenant_B vrf-target target:10:200
set routing-instances VRF_Tenant_C instance-type vrf
set routing-instances VRF_Tenant_C interface irb.1300
set routing-instances VRF_Tenant_C route-distinguisher 1.255.255.1:1300
set routing-instances VRF_Tenant_C vrf-target target:10:300
set routing-instances VRF_Tenant_D instance-type vrf
set routing-instances VRF_Tenant_D interface irb.1400
set routing-instances VRF_Tenant_D route-distinguisher 1.255.255.1:1400
set routing-instances VRF_Tenant_D vrf-target target:10:400
set routing-instances VS_VLAN100 vtep-source-interface lo0.0
set routing-instances VS_VLAN100 instance-type virtual-switch
set routing-instances VS_VLAN100 route-distinguisher 1.255.255.1:100
set routing-instances VS_VLAN100 vrf-import VS_VLAN100_IMP
set routing-instances VS_VLAN100 vrf-target target:1:100
set routing-instances VS_VLAN100 protocols evpn encapsulation vxlan
set routing-instances VS_VLAN100 protocols evpn extended-vni-list 1100
set routing-instances VS_VLAN100 protocols evpn multicast-mode ingress-replication
set routing-instances VS_VLAN100 protocols evpn default-gateway no-gateway-
community
set routing-instances VS_VLAN100 bridge-domains bd1100 vlan-id 100
set routing-instances VS_VLAN100 bridge-domains bd1100 routing-interface irb.1100
set routing-instances VS_VLAN100 bridge-domains bd1100 vxlan vni 1100
set routing-instances VS_VLAN100 bridge-domains bd1100 vxlan ingress-node-
replication
set routing-instances VS_VLAN200 vtep-source-interface lo0.0
```

```
set routing-instances VS_VLAN200 instance-type virtual-switch
set routing-instances VS_VLAN200 route-distinguisher 1.255.255.1:200
set routing-instances VS_VLAN200 vrf-import VS_VLAN200_IMP
set routing-instances VS_VLAN200 vrf-target target:1:200
set routing-instances VS_VLAN200 protocols evpn encapsulation vxlan
set routing-instances VS_VLAN200 protocols evpn extended-vni-list 1200
set routing-instances VS_VLAN200 protocols evpn multicast-mode ingress-replication
set routing-instances VS_VLAN200 bridge-domains bd1200 vlan-id 200
set routing-instances VS_VLAN200 bridge-domains bd1200 routing-interface irb.1200
set routing-instances VS_VLAN200 bridge-domains bd1200 vxlan vni 1200
set routing-instances VS_VLAN200 bridge-domains bd1200 vxlan ingress-node-
replication
set routing-instances VS_VLAN300 vtep-source-interface lo0.0
set routing-instances VS_VLAN300 instance-type virtual-switch
set routing-instances VS_VLAN300 route-distinguisher 1.255.255.1:300
set routing-instances VS_VLAN300 vrf-import VS_VLAN300_IMP
set routing-instances VS_VLAN300 vrf-target target:1:300
set routing-instances VS_VLAN300 protocols evpn encapsulation vxlan
set routing-instances VS_VLAN300 protocols evpn extended-vni-list 1300
set routing-instances VS_VLAN300 protocols evpn multicast-mode ingress-replication
set routing-instances VS_VLAN300 bridge-domains bd1300 vlan-id 300
set routing-instances VS_VLAN300 bridge-domains bd1300 routing-interface irb.1300
set routing-instances VS_VLAN300 bridge-domains bd1300 vxlan vni 1300
set routing-instances VS_VLAN300 bridge-domains bd1300 vxlan ingress-node-
replication
set routing-instances VS_VLAN400 vtep-source-interface lo0.0
set routing-instances VS_VLAN400 instance-type virtual-switch
set routing-instances VS_VLAN400 route-distinguisher 1.255.255.1:400
set routing-instances VS_VLAN400 vrf-import VS_VLAN400_IMP
set routing-instances VS_VLAN400 vrf-target target:1:400
set routing-instances VS_VLAN400 protocols evpn encapsulation vxlan
set routing-instances VS_VLAN400 protocols evpn extended-vni-list 1400
set routing-instances VS_VLAN400 protocols evpn multicast-mode ingress-replication
set routing-instances VS_VLAN400 bridge-domains bd1400 vlan-id 400
set routing-instances VS_VLAN400 bridge-domains bd1400 routing-interface irb.1400
set routing-instances VS_VLAN400 bridge-domains bd1400 vxlan vni 1400
set routing-instances VS_VLAN400 bridge-domains bd1400 vxlan ingress-node-
replication
```

おわりに

企業の IT 戦略においてクラウド ベースのリソースが大きな部分を占めるようになり、セキュリティーや性能で妥協せずに、クラウドベースのサービスに追随するためのネットワーク アーキテクチャが求められています。また、データ センターのユーザーは時間と場所を問わずにアクセスできることや高レベルな応答性を求めており、今日のネットワーク アーキテクチャでは、その達成がますます困難になっています。

イーサネット VPN（EVPN）は、データ センター ネットワークを構築、相互接続する上で、効率性と拡張性に優れた方法です。EVPN が複数のデータ プレーン（VXLAN および MPLS）で機能するため、最初に複数のコントロール プレーンを統合する必要がなく、統合されたコントロール プレーンを使用して、拡張性の高いエンドツーエンドのネットワークを構築できます。EVPN は、ワークロードの移動、最適なルーティング、マルチパスによるネットワーク効率性に関した課題も、包括的に解決します。すべてのプラットフォーム（QFX シリーズ スイッチ、EX シリーズ スイッチ、MX シリーズ ルーター）に堅固な BGP/EVPN を実装するジュニパーネットワークスは、最適かつシームレスで、標準に準拠した L2 または L3 接続を提供しており、EVPN 技術の潜在力をすべて引き出す唯一の企業です。

ジュニパーネットワークスについて

ジュニパーネットワークスは、ネットワークイノベーション事業に従事しています。デバイスからデータセンターまで、消費者からクラウドプロバイダまで、ジュニパーネットワークスはネットワークの使い勝手や経済性を向上させるソフトウェア、シリコン技術やシステムを提供しています。弊社は、世界中にお客様やパートナーを抱えています。詳しい情報は、www.juniper.net/jp/ をご覧ください。

米国本社
Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089 USA
電話：888.JUNIPER（888.586.4737）
または +1.408.745.2000
FAX：+1.408.745.2100
www.juniper.net

アジアパシフィック、ヨーロッパ、中東、アフリカ
Juniper Networks International B.V.
Boeing Avenue 240
1119 PZ Schiphol-Rijk
Amsterdam, The Netherlands
電話：+31.0.207.125.700
FAX：+31.0.207.125.701

ジュニパーネットワークスのソリューションの購入については、03-5333-7400 にお電話いただくか、認定リセラーにお問い合わせください。

2000606-001-JP 2015 年 7 月