WHITE PAPER

QFX5100 スイッチを使用した CLOS IP ファブリック

レイヤー 3 プロトコルおよびオーバーレイ ネットワークを利用した、柔軟でプログラム可能なデータ センター ネットワークの構築

目次

図一覧

エグゼクティブ サマリー

本書では、ジュニパーネットワークスの QFX5100 シリーズ スイッチを用いて大規模 IP ファブリックを構築する方法について説明します。本書はネットワーク エンジニアがネットワーク エンジニア向けに記述したもので、Clos IP ファブリックを構築する理由、および QFX5100 スイッチを用いて Clos IP ネットワークを構築、設定、確認する方法を示します。本書で示す設定例に基づいて、Clos IP ネットワークを設計、設定することができます。

はじめに

ネットワークの世界では、いたるところで IP ファブリックや Clos ネットワークについて目にすることがあるでしょう。何かが始まろうとしています。それはいったい何でしょうか。また IP ファブリックの必要性を高めているのは何でしょうか。IP ファブリックが答えだとしたら、解決しようとしている問題は何でしょうか。

これまで長い間、多くの OTT（Over The Top）企業が大規模な IP ファブリックを構築してきましたが、それについてほとんど注目されることはありませんでした。一般的にこれらの企業は、コンピューティングの仮想化を必要とせず、高可用性を組み込む方法でアプリケーションを作成します。インテリジェントなアプリケーションを使用し、コンピューティングの仮想化が必要ないのであれば、レイヤー 3 プロトコルだけを使って IP ファブリックを構築するのは理にかなっています。従来、レイヤー 2 は拡張性や高可用性の観点から、データ センターの弱点とされてきました。多数のデバイス間でトラフィックを大量に送信し、ネイティブで生存時間フィールドをもたないイーサネット フレーム上でループを回避しなければならない場合、これは解決が難しい問題です。

これまで企業が長い間大規模 IP ファブリックを構築してきたのにもかかわらず、最近 IP ファブリックが注目されるようになってきたのはなぜでしょうか。その答えは、データ センターのオーバーレイ ネットワークに関連しています。これにより解決される問題は 2 つあります。1）ネットワークの俊敏性、および 2）ネットワークの簡素化です。データ センターで IP ファブリックと統合されたオーバーレイ ネットワーク、これが答えです。

オーバーレイ ネットワーク

次世代データ センターの設計でまず始めに考慮すべきことの 1 つは、「アプリケーションを迅速に導入できるようにするため、データ センターのすべてのリソースを中央でオーケストレーションさせる必要があるか ?」ということです。また、それを補足する質問として、「現在データ センターのコンピューティングやストレージをハイパーバイザまたはクラウド管理プラットフォームで仮想化しているかどうか ?」という質問があります。これらの質問に対する答えが「はい」の場合、データ センター ネットワークにオーバーレイ アーキテクチャを検討する必要があります。

コンピューティングとストレージがどのように仮想化されているかを確認したら、次のステップはデータ センター ネットワークの仮想化です。データ センターでオーバーレイ アーキテクチャを使用することで、物理ハードウェアをネットワークから切り離すことができます。これは仮想化の基本原則の 1 つです。ネットワークを物理ハードウェアから切り離すことで、データ センター ネットワークをプログラムによって瞬時にプロビジョニングできます。

オーバーレイ ネットワークの 2 つ目の利点は、VM とサーバー間のレイヤー 2 およびレイヤー 3 転送の両方に対応していることです。これは従来の IT データ センターにおいて非常に求められることです。3 つ目の利点は、従来の VLAN より大幅に拡張性があり、最高 1,670 万テナントまで対応していることです。オーバーレイ アーキテクチャに対応している製品の例として、ジュニパーネットワークス Contrail と VMware NSX が挙げられます。

オーバーレイ アーキテクチャに移行すると、別の「ネットワークに関する負担」がデータ センターにかかります。従来、ネットワークに接続されるサーバーや仮想マシンは、それぞれネットワークの MAC アドレスとホスト ルート エントリを必要とします。しかしオーバーレイ アーキテクチャにおいては、仮想トンネルのエンドポイント（VTEP）のみがネットワークの MAC アドレスとホスト ルート エントリを必要とします。すべての仮想マシン（VM）とサーバー間のトラフィックは VTEP 間でカプセル化され、MAC アドレスや各 VM とサーバーのホスト ルートは、基盤となるネットワーク機器からは見えません。MAC アドレスやホスト ルートの拡張性は、物理ネットワーク ハードウェアからハイパーバイザに依存するようになりました。

ベア メタル サーバー

コンピュート リソースを 100% 仮想化しているデータ センターを見かけることはあまりありません。通常、パフォーマンスやコンプライアンス、その他数多くの理由により、サーバーの一部が仮想化できません。これは次のような興味深い質問を引き出します。データ センターのサーバーの 80% を仮想化してオーバーレイ アーキテクチャを活用する場合、残り 20% の物理サーバーとの接続性はどのように実現できるでしょうか。

オーバーレイ アーキテクチャは、物理サーバーとの接続を可能にするいくつかのメカニズムに対応しています。最も一般的なオプションは、図 1 に示すように、VTEP を物理アクセス スイッチに組み込む方法です。

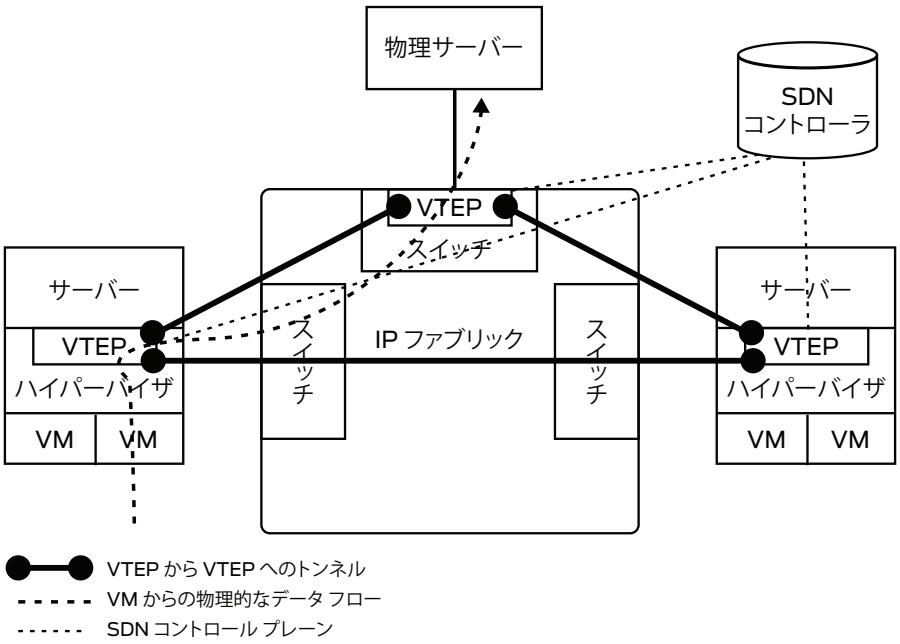

図 1：オーバーレイ アーキテクチャにおける仮想マシンからの物理的なデータ フロー

図 1 で IP ファブリックの左側と右側に示される各サーバーは、ハイパーバイザで仮想化されています。各ハイパーバイザは内部に VTEP があり、仮想マシン間のデータ プレーン トラフィックのカプセル化を処理します。各 VTEP は MAC アドレスの学習や新規仮想ネットワークのプロビジョニング、その他の設定変更も行います。IP ファブリック上にあるサーバーはシンプルな物理サーバーですが、それ自体には VTEP の機能はありません。物理サーバーがオーバーレイ アーキテクチャに加わるには、データ プレーン トラフィックをカプセル化し、MAC アドレスの学習を行う何らかの機能が必要です。アクセス スイッチ内部の VTEP の役割を処理できることで、オーバーレイ アーキテクチャが簡素化されます。アクセス スイッチに接続する物理サーバーをもつ各アクセス スイッチは、オーバーレイをカプセル化し、コントロール プレーンを物理サーバーに代わって実行できます。物理サーバーの観点から見ると、他には何もせずにネットワークへのトラフィック送信のみ簡素化します。

IP ファブリック

簡単に言うと、IP ファブリックには主に 2 つの推進要因があります。それは、レイヤー 3 を要件とする OTT 企業、そして基盤レイヤーとして IP ファブリックを使用するオーバーレイ ネットワークの導入です。ではまず始めに、データ センターにおけるオーバーレイ ネットワークの要件と、IP ファブリックがどのようにその要件を満たし、それを超えるかについて見ていきます。

オーバーレイ アーキテクチャの VTEP 間では、すべての VM およびサーバーの MAC アドレスやトラフィック、フラッディングをカプセル化できます。VTEP のネットワーク要件は、レイヤー 3 接続に対応しているということだけです。このネットワーク要件を満たすネットワークを構築するのは非常に簡単です。課題は、サイズの拡大に合わせて直線的に拡張できる転送アーキテクチャをどのように設計するかということです。通信業界でこれに非常に似た問題が 1953 年に解決されました。チャールズ・クロ氏が、図 2 に示すような、ネットワークの最大スイッチを超えて拡張できる多階層ネットワークの構築手法を発明しました。

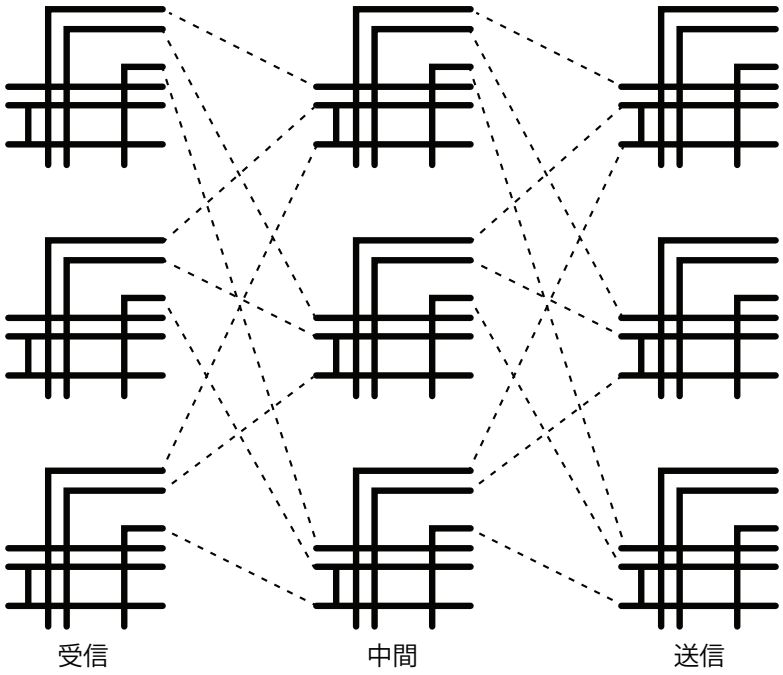

図 2：チャールズ・クロの多階層トポロジー

Clos トポロジーの利点は、ブロックされず、パフォーマンスを予測でき、拡張性が高いという点です。図 2 は、受信、中間、送信の 3 階層 Clos ネットワークを表しています。

IP ファブリックの構築時には、Clos ネットワークと同じ原則を引き継いで適用できます。すでに多くのネットワークがこのように設計され、これは図 3 に示すように、リーフ / スパイン型ネットワークと呼ばれます。

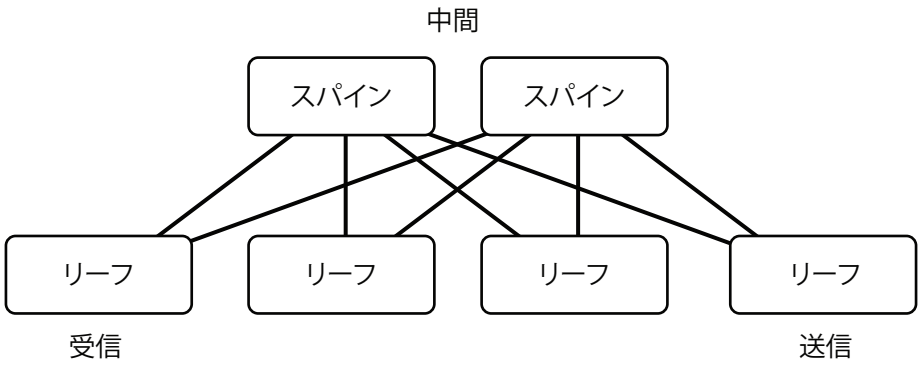

図 3：リーフ / スパイン型トポロジー

実はリーフ / スパイン型ネットワークは 3 階層 Clos ネットワークと同等で、図 3 に示すように、受信ポイントと送信ポイントが互いの上に折りたたまれているため、折りたたみ 3 階層 Clos ネットワークとも呼ばれます。この例では、スパイン スイッチは単純なレイヤー 3 スイッチ、リーフはトップオブラック スイッチで、サーバーと VTEP の接続を提供しています。

Clos ネットワークのポート数を拡張するための秘訣は、スパイン幅とオーバーサブスクリプション比の 2 つの値を調整することです。スパイン幅を広げると、IP ファブリックがより多くのリーフに対応できるようになります。また、リーフ上のオーバーサブスクリプション値を大きくすると、IP ファブリックがより大きなリーフに対応できるようになります。どのように各要素がまとめられて最終的な結果が得られるかを理解するため、トポロジーの例をいくつか見てみましょう。

768 x 10 ギガビット イーサネット バーチャル シャーシ ファブリック

1 例目は、バーチャル シャーシ ファブリックと呼ばれる、新しいジュニパー テクノロジです。これによって、単一デバイスとして管理できる QFX5100 スイッチ一式を使った 3 階層の IP ファブリック構築が可能になります。図 4 に示すように、ジュニパーネットワークス Junos® オペレーティング システム リリース 13.2 では、バーチャル シャーシ ファブリックのスイッチの最大数は 20 です。

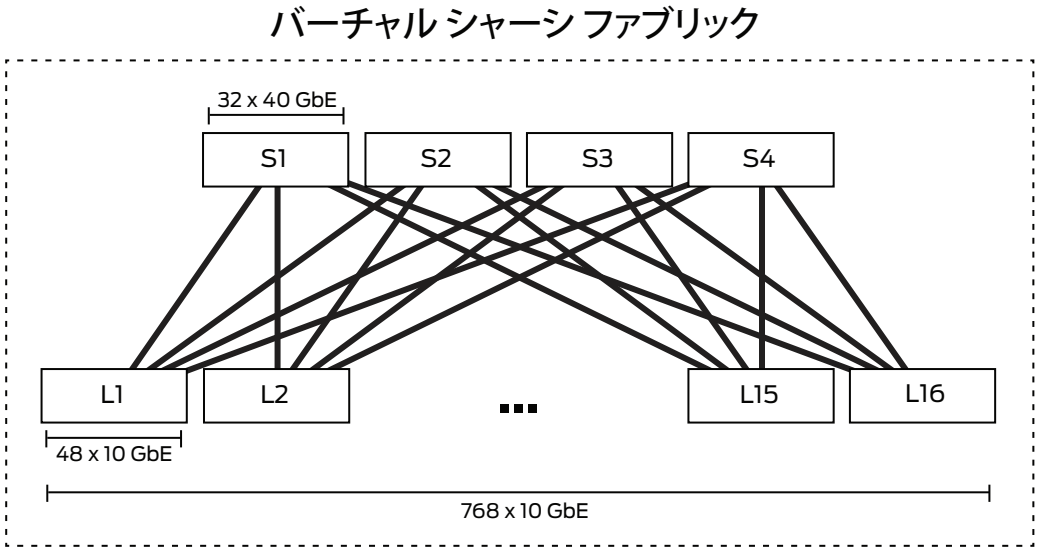

図 4：768 x 10 GbE ポートのバーチャル シャーシ ファブリック

この例では、スパインは 4 つの QFX5100-24Q スイッチから構成されています。各スイッチは、最高 32 x 40 ギガビット イーサネット（GbE）のインターフェイスに対応しています。リーフは、48 x 10 GbE および 6 x 40 GbE インターフェイスに対応した QFX5100-48S を使って構築されています。図 4 に示すように、各リーフはアップリンクとして 4 x 40 GbE インターフェイスを使用し、それぞれ各スパインにリンクしています。これは 1 リーフあたり 480 対 160 または 3 対 1 のオーバーサブスクリプション比になります。バーチャル シャーシ ファブリックは 20 スイッチまでしか対応していないため、合計 20 スイッチの場合、合計 4 個のスパイン スイッチと 16 個のリーフ スイッチが使用可能です。各リーフは 48 x 10 GbE インターフェイスに対応しています。合計 16 個のリーフがあるため、オーバーサブスクリプション比 3 対 1 の場合、合計ポート数は最高 768 x 10 GbE になります。

拡張要件がバーチャル シャーシ ファブリックのキャパシティーを超えている場合でも、問題にはなりません。次のオプションとして、何千ものポートに拡張できる、単純な 3 階層 IP ファブリックを構築するという方法があります。

3072 x 10 GbE IP ファブリック

次のオプションとして、QFX5100-24Q および QFX5100-96S を使い、単純な 3 階層 IP ファブリックを構築するという方法がありますが、今度はバーチャル シャーシ ファブリックは使いません。QFX5100-24S は 32 x 40 GbE スイッチで、QFX5100-96S は 96 x 10 GbE および 8 x 40 GbE スイッチです。図 5 に示すように、QFX5100-24Q と QFX5100-96S を統合し、有効な 3072 x 10 GbE ポートの IP ファブリックを構築します。

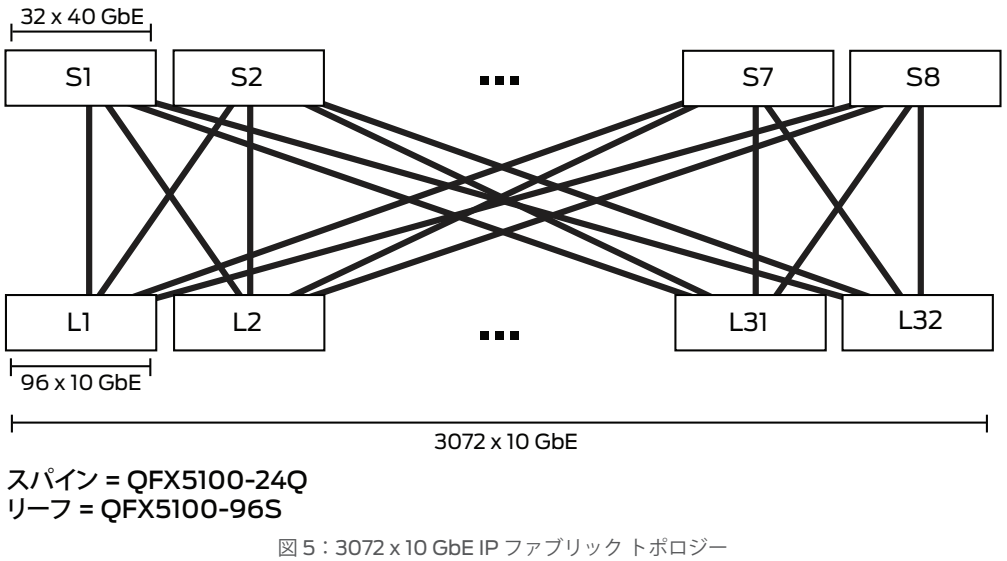

図 5：3072 x 10 GbE IP ファブリック トポロジー

各リーフは QFX5100-96S を使って構築し、スパインへのアップリンクとして 8 x 40 GbE インターフェイスを使用します。各リーフにはスパインへのアップリンクが 8 個あるため、スパインの最大幅は 8 です。1 リーフあたり 40 GbE のインターフェイスが、独立したスパインに接続しているため、各リーフは 1 スパインあたり 40 GbE インターフェイスを必要とします。IP ファブリックの最大サイズを計算するには、リーフのサーバー インターフェイス数に、スパインが対応するリーフ数を掛け合わせます。この例では、スパインは 32 のリーフに対応し、各リーフは 96 ポートの 10 GbE に対応しています。したがって、オーバーサブスクリプション比 3 対 1 の場合、有効な 10 GbE ポート数は合計 3072 になります。

コントロール プレーン オプション

バーチャル シャーシ ファブリックを使用する最大の利点の 1 つは、IP ファブリックの基盤となるコントロール プレーン プロトコルについて気にする必要がないということです。常に問題なく動作します。ただし、バーチャル シャーシ ファブリックの拡張性を超えたネットワークを構築する必要がある場合、コントロール プレーン オプションとは何かについて理解しておく必要があります。

IP ファブリックを構築する際の基本的な要件の 1 つは、プレフィックスの配布です。各リーフは、IP ファブリック内のすべてのリーフと IP ルーティング情報を送受信しなければなりません。ここで問題となるのは、IP ファブリックコントロール プレーンのオプションには何があるか、またどれが最適かということです。まず始めに、IP ファブリックの基本的な要件を確認し、表 1 に示すように、その要件をコントロール プレーンのオプションに対応付けてみます。

表 1：IP ファブリック要件とコントロール プレーンのオプション

要件	OSPF	IS-IS	BGP
プレフィックスのアドバタイズ	対応	対応	対応
拡張性	制限あり	制限あり	幅広く対応
トラフィック エンジニアリング	制限あり	制限あり	幅広く対応
トラフィック タギング	制限あり	制限あり	幅広く対応
マルチベンダーの安定性	対応	対応	幅広く対応

IP ファブリックのコントロール プレーンの最も一般的なオプションは、OSPF、IS-IS、および BGP です。各プロトコルは基本的にプレフィックスをアドバタイズできますが、プロトコルによって拡張性や機能は様々です。OSPF および IS-IS は更新情報や他のルーティング情報を送信するためにフラッディング テクニックを使います。エリアを構築することでフラッディング量を制限することができますが、そうすると SPF ルーティング プロトコルの利点を打ち消してしまいます。一方、BGP は一から構築され、多数のプレフィックスやピアリング ポイントに対応しています。これを裏付ける最高の使用例はインターネットです。

IP ファブリック周辺のトラフィックをシフトする機能は有用です。たとえば、特定のスパイン スイッチがメンテナンス中でも、その周辺のトラフィックを制御することができます。OSPF と IS-IS のトラフィック エンジニアリングやトラフィック タギング機能は制限されています。繰り返しになりますが、BGP は一から構築され、多数のトラフィック エンジニアリング、そしてローカル設定や MED、拡張コミュニティなどの機能とのタギングに対応するよう設計されています。

大規模な IP ファブリックを構築する副次的影響の 1 つに、通常は繰り返し長い時間をかけて構築されることが挙げられます。複数のベンダーが 1 つの IP ファブリックを構築するというのはよくあることです。OSPF も IS-IS も複数ベンダー間で正しく動作しますが、最適なのは BGP です。前に説明したように、世界で最も優れた使用例はインターネットです。インターネットは多数のベンダーや機器、様々な要素から構成されていますが、プレフィックスのアドバタイズやトラフィック エンジニアリング、トラフィック タギングを行うコントロール プレーン プロトコルとして、すべて BGP を使用しています。

拡張性、トラフィック タギング、マルチベンダーの安定性の観点から、IP ファブリックのコントロール プレーン プロトコルを選択する上で、BGP が最適です。次の問題は、IP ファブリックの BGP をどのように設計するかです。

BGP 設計

最初に判断しなければならないことの 1 つは、IBGP と EBGP のどちらを使うかということです。IP ファブリックの本質は、ECMP（等価コスト マルチパス）に基づいています。設計時に考慮しなければならないことの 1 つに、各オプションが ECMP をどのように処理するかということがあります。デフォルトで、EBGP は問題なく ECMP に対応しています。ただし、IBGP が ECMP に完全に対応するためには、BGP ルート リフレクタと AddPath 機能が必要です。

では IP ファブリックにおける EBGP 設計について詳しく見ていきましょう。図 6 に示すように、各スイッチは異なる AS 番号をもち、各リーフは IP ファブリックのすべてのスパインとピアでなければなりません。

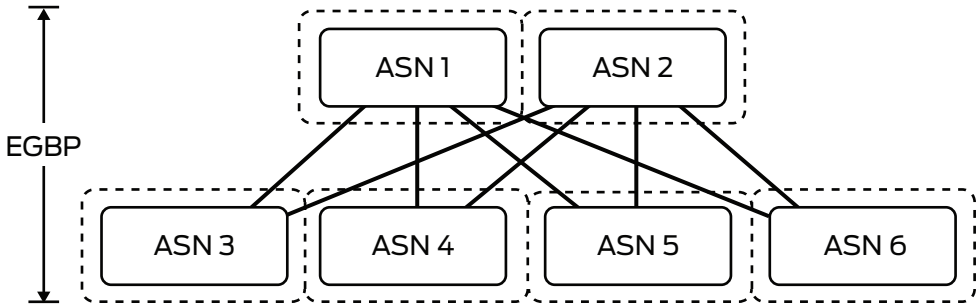

図 6：IP ファブリックでの EBGP の使用

IP ファブリックで EBGP を使用するのは、非常にシンプルで簡単です。また EBGP は、ローカル設定や AS パディング テクニックにより、トラフィック エンジニアリングに適したものになります。

IBGP の場合、すべてのスイッチが IP ファブリック内のその他すべてのデバイスとピアであることが要求されるため、IP ファブリックで IBGP を設計する場合は少々異なります。IP ファブリック内のその他すべてのデバイスとピアであることが要求されることによる負担を最小限にするため、図 7 に示すように、ネットワークのスパインでインライン BGP ルート リフレクタを使用できます。標準的な BGP ルート リフレクションを使用する場合の問題点は、最適なプレフィックスのみ反映し、ECMP には適していないということです。ECMP を完全に使えるようにするには、BGP AddPath 機能を使う必要があります。これはルート リフレクタとクライアント間の BGP アドバタイズメントへの追加 ECMP パスを提供します。

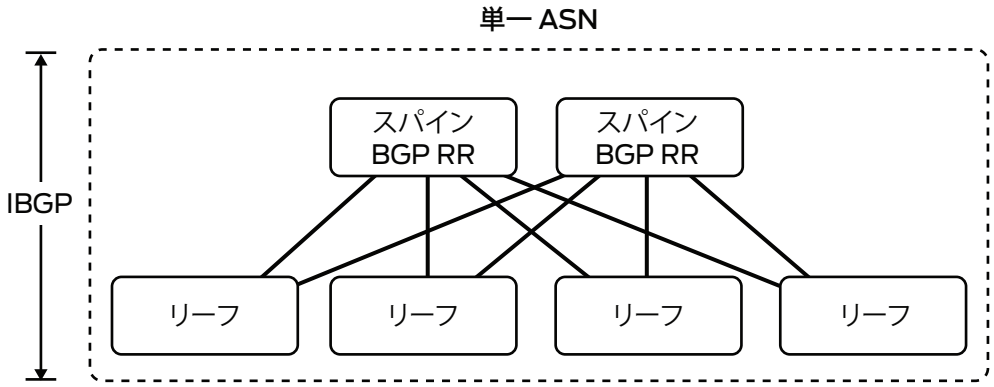

図 7：IP ファブリックでの IBGP の使用

QFX5100 スイッチは、IBGP と EBGP の両方の BGP 設計オプションに対応しています。どちらのオプションも同じように問題なく動作します。ただし、EBGP の設計と実装の方がシンプルです。一般的に、可動部品が少ないほどマシンの設計は安定します。単純な 3 階層 IP ファブリックを構築する際には、EBGP を使うことを推奨します。それは BGP ルート リフレクションや AddPath について考慮する必要がないためです。

実装要件

IP ファブリックの設計図を作成する際、満たさなければならない要件があります。高度なレベルでは、IP アドレス管理（IPAM）および BGP 割り当てが中心になります。ではこの要件を詳細レベルで見ていきましょう。

- **ベース IP プレフィックス**：IP ファブリック内のすべての IP アドレス割り当ては、共通のベース IP プレフィックスから行う必要があります。ベース IP プレフィックスが、IP ファブリック内の各スイッチのポイントツーポイント アドレス指定およびループバック アドレス指定に対応するのに十分なアドレス領域を保持していることが不可欠です。
- **ポイントツーポイント ネットワーク マスク**：各リーフは IP ファブリックのすべてのスパインに接続されます。この接続は、ポイントツーポイント リンクと呼ばれます。このポイントツーポイント リンクで使われるネットワーク マスクは、ベース IP プレフィックスがどの程度使われるかを定義します。たとえば、30 ビットのネットワーク マスクを使用する場合、31 ビットのネットワーク マスクを使用する場合の 2 倍の領域が必要です。
- **ポイントツーポイント IP アドレス**：各ポイントツーポイント接続では、各スイッチに IP アドレスを割り当てる必要があります。スパインが小さな番号の IP アドレス割り当てを受信するか、大きな番号の IP アドレス割り当てを受信するかを決定する必要があります。これは表面上の決定で、IP ファブリックの機能には影響しません。
- **サーバー向け IP プレフィックス**：VTEP にレイヤー 3 ゲートウェイを提供するためには、サーバー向けトラフィックに使用する一貫性のある IP プレフィックスをリーフに設定する必要があります。これは IP ファブリックを構成するために使われるベース IP プレフィックスとは独立しています。サーバー向け IP プレフィックスは、IP ファブリックの各リーフのアドレス要件に対応できる十分な大きさをもっていなければなりません。たとえば、各リーフが 24 ビットのサブネットを必要とし、512 個のリーフがある場合、サーバー向け IP プレフィックスは最低 15 ビット（192.168.0.0/15 など）必要です。この場合、512 個の 24 ビット サブネットが可能です。各リーフは 24 ビット サブネット（192.168.0.0/24 など）をもち、レイヤー 2 ゲートウェイ サービスに最初の IP アドレス（192.168.0.1/24 など）を使用できます。
- **ループバック アドレス指定**：IP ファブリックの各スイッチには、32 ビット マスクを使用した単一ループバック アドレスが必要です。ループバック アドレスは、トラブルシューティングやスイッチ間の接続状況確認のために使用できます。
- **BGP AS 番号**：IP ファブリックの各スイッチには、個別の AS 番号が必要です。各スパインとリーフは、個別の BGP AS 番号をもちます。これにより、EBGP をリーフとスパイン間で使用することができます。
- **BGP エクスポート ポリシー**：各リーフはローカル サーバー向け IP プレフィックスを IP ファブリックにアドバタイズし、他のサーバーからアクセスできるようにしなければなりません。各リーフはまた、ループバック アドレスを IP ファブリックにもエクスポートする必要があります。
- **BGP インポート ポリシー**：各リーフはサーバー向け IP プレフィックスおよびループバック アドレス指定にのみ注力するため、その他のポイントツーポイント リンクのアドレス指定はすべて除外することができます。
- **等価コスト マルチパス ルーティング**：各スパインおよびリーフは、等価ネクスト ホップ全体でロードバランス フロー機能をもつ必要があります。たとえば 4 個のスパイン スイッチがある場合、各リーフは各スパインと接続します。あるリーフ スイッチから出る各フローには、各スパイン向けに 4 個の等価ネクスト ホップがなければなりません。そのためには、ECMP（等価コスト マルチパス）ルーティングが有効である必要があります。

これらの要件により、IP ファブリックの設計図を容易に描くことができます。ネットワークは初日から完全に構築できるものではありませんが、IP ファブリックの拡張可能な設計図を描いておくことによって、将来どのようにネットワークを拡張するかを思い悩む必要がなくなります。

判断ポイント

IP ファブリックを設計する際に考慮すべきいくつかの重要な判断ポイントがあります。1 つ目の判断ポイントは、IBGP と EBGP のどちらを使用するかです。一見これは簡単な選択のように見えますが、この判断を複雑にするいくつかの要素があります。2 つめの判断ポイントは、実は最初に選ぶべきものですが、16 ビットと 32 ビットのどちらの ASN を使うかです。では 2 つの判断ポイントを詳しく見てみましょう。

1 つ目の判断ポイントによって、JNCIE や CCIE の時代に引き戻されることでしょう。IBGP と EBGP の要件は何でしょうか。プレフィックスをトポロジー全体にプロパゲーションするには IBGP をフルメッシュ化する必要があります。一方、EBGP はフルメッシュ化が必要なくて、より柔軟です。この背後にある理由は、明らかにループの防止です。ループを防止するため、IBGP はある IBGP ピアから検出されたプレフィックスを他のピアにプロパゲーションしません。各 IBGP スイッチが完全にルートをプロパゲーションするには、他スイッチへの BGP セッションを設定する必要があります。一方、EBGP は単純に全 BGP プレフィックスを全 BGP ネイバーにプロパゲーションします。例外は、自身の AS 番号がドロップされたスイッチを含むプレフィックスです。

IBGP 設計

それでは、IP ファブリック構築の実装要件をすべて満たす IBGP 設計について見ていきます。最初の課題は、IBGP のフルメッシュ化の要件をどのように満たすかということです。その答えは、BGP コンフェデレーションまたはルート リフレクションです。IP ファブリックが固定トポロジーであれば、ルート リフレクションが適しています。各スパイン スイッチは BGP ルート リフレクタとして、各リーフは BGP ルートリフレクタ クライアントとして動作します。

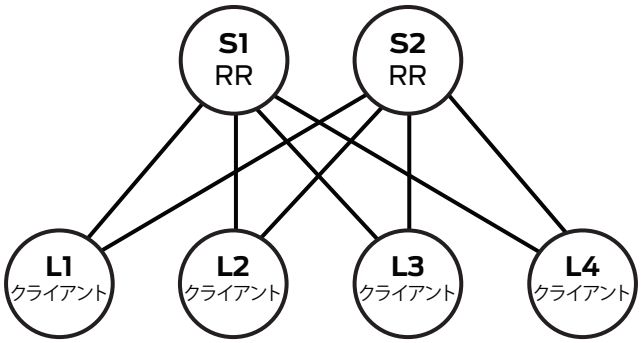

図 8：ルート リフレクタを使用した IBGP 設計

図 8 に示すように、各スパインは BGP ルート リフレクタとして動作します。L1 から L4 までの各リーフは、BGP ルート リフレクタ クライアントです。

> IBGP 設計を使用する場合、スパイン スイッチが BGP ルート リフレクションに対応しているかを確認することが重要です。幸いなことに、QFX5100 シリーズのスイッチは BGP ルート リフレクションに対応しています。

IBGP 設計で満たすべき別の重要な実装要件は、ECMP ルーティングです。デフォルトでは BGP ルート リフレクタは最適なルートのみ反映します。これはすなわち、4 つの ECMP ルートがある場合、最適なプレフィックス 1 つだけがクライアントに反映されるということです。これは明らかに ECMP 要件を満たさないため、何か対処しなければなりません。

この問題に対する答えは、最適なパスの代わりに複数のパスを送信する BGP ルート リフレクタを可能にすることです。現在、IETF にはこの動作を可能にするドラフト版があり、`draft-ietf-idr-add-paths` と呼ばれます。また、この機能は BGP AddPath と呼ばれます。これでルート リフレクタはすべての ECMP ルートを各クライアントに提供可能になります。

> IBGP を使ってネットワークを設計する場合、IP ファブリックのスパイン スイッチが BGP Add Path に対応しているかを確認してください。QFX5100 シリーズのスイッチは、BGP Add Path および BGP ルート リフレクションに対応しています。

まとめると、すべての IP ファブリック要件を満たすためには、スパイン スイッチは BGP ルート リフレクションおよび BGP Add Path に対応している必要があります。IBGP には他にもいくつかの要件がありますが、IBGP により IP ファブリック全体を 1 つの AS 番号として管理できます。

EBGP 設計

もう 1 つの選択肢は、EBGP を使って IP ファブリックを設計する方法です。EGBP は、デフォルトで IP ファブリック構築のすべての実装要件を満たしています。図 9 に示すように、EGBP では、BGP ルート リフレクションや BGP Add Path は必要ありません。

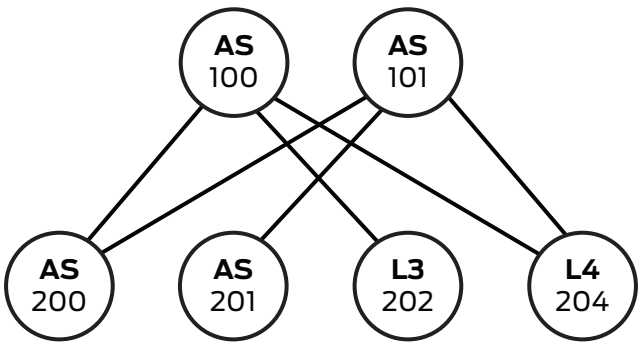

図 9：EBGP ではスイッチごとに BGP AS 番号が必要

実際に考慮しなければならない唯一の問題は、IP ファブリックで BGP AS 番号がいくつ必要かということです。各スイッチには個別の BGP AS 番号を設定します。技術的には、BGP プライベートの範囲は 64,512 から 65,535 で、これはすなわち 1023 の BGP AS 番号が使えることになります。IP ファブリックのスイッチが 1023 個以上の場合、パブリック BGP AS 番号の範囲への移行、または 32 ビット AS 番号への移行を検討する必要があります。

> 前述のように、EBGP は非常に単純な設計になっていて、すべての実装要件を満たします。そのため、マルチベンダー IP ファブリックを構築する際に適しています。

エッジ接続性

もう 1 つの重要な判断ポイントは、データ センターを世界中の他のオフィスやデータ センターとどのように接続するかという点です。エンドポイントやオプションの数に応じて、複数の判断ポイントがあります。図 10 に示すような、単純なデータ センターの例を見てみましょう。

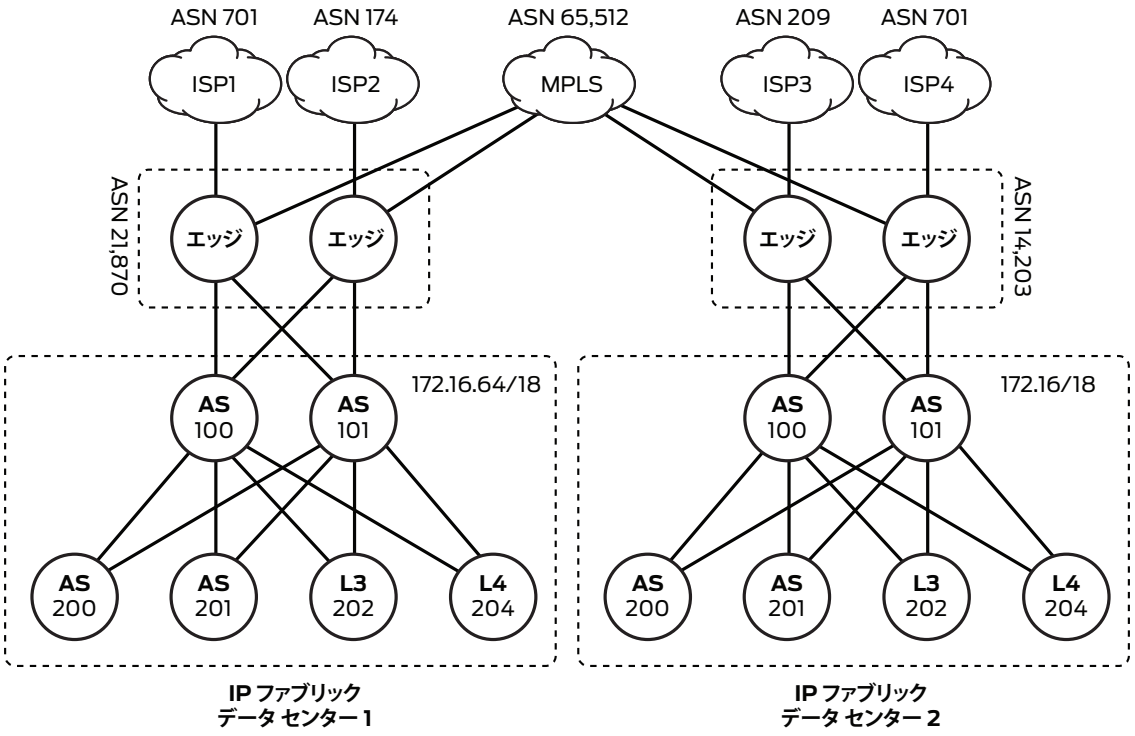

図 10：IP ファブリックで構築された 2 つのデータ センターのエッジ接続性

各データ センターには以下のことが求められます。

- 同一の BGP AS 番号とスキームを使った IP ファブリックで構築されている
- 固有の BGP AS 番号をもつエッジ ルーターが 2 つある
- 2 つの ISP に接続されている
- プライベート MPLS ネットワークに接続されている

各データ センターで同一の EBGP 設計を再利用することで、新規データ センター立ち上げ時の運用負荷を減らすことができます。またどのデータ センターにいても、一貫した運用が可能になります。短所は、設計全体で同一の AS 番号を使うため、MPLS コアで混乱するということです。たとえば AS 200 の BGP プレフィックスは何かという質問をした場合、答えはどのデータ センターにいるかによって異なります。

シンプルなソリューションは、BGP AS 優先機能を使うことです。これにより、各データ センターのエッジ ルーターで使う AS を MPLS ネットワークの PE ルーターで変更することができます。ここでは単純に、AS 番号 21,870 が集約ネットワーク 172.16.64/18 を、そして AS 番号 14,203 が集約ネットワーク 172.16/18 を所有しています。データ センター 1 から見ると、172.16/18 へのルートは BGP AS 番号 65,512 を通り、次に 14,203 を通ります。これを実現するには、すべての IP ファブリックのプレフィックスを拒否し、代わりに 1 つの BGP 集約をアドバタイズする BGP エクスポート ポリシーを各データ センターのエッジ ルーターに作成します。

インターネットへの接続については、設計が多少異なります。目的は IP ファブリックが 0/0 のデフォルト ルートをもつことですが、エッジ ルーターが完全なインターネット テーブルをもつ必要があります。各データ センターには、各 ISP 向けにアドバタイズされるべき固有のパブリック IP 範囲もあります。まとめると、エッジ ルーターは以下のアクションを実行します。

- IP ファブリックへのデフォルト ルートをアドバタイズする
- 各 ISP へのパブリック IP 範囲をアドバタイズする
- その他のプレフィックスをすべて拒否する

BGP 設計のまとめ

IP ファブリックにより、データ センターのオーバーレイ ネットワーク アーキテクチャを簡単にサポートできる非常に大規模なネットワークを構築することができます。IP ファブリックを構築する際、注意しなければならない判断ポイントがあります。それは、「何個のスイッチを実装するか ?」、「BGP 機能を使ってマルチベンダー環境を設計するか ?」、「互いにいくつのデータ センターを接続するか ?」です。これらは、各データ センターの全体設計時に考えるべき項目です。

BGP の実装

では核心に入っていきましょう。設計フェーズから実装フェーズに入るには、物理デバイスや設定、確認が必要になります。このセクションでは、Junos OS を使った実装について詳細に見ていきます。図 11 に示すように、このラボには 2 つのスパインと 3 つのリーフがあります。

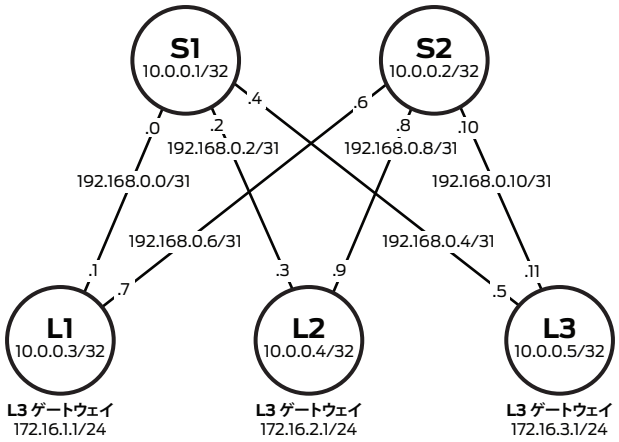

図 11：IP ファブリックの BGP 実装

- ループバック アドレス：各スイッチは 10/8 範囲からの 32 ビット ループバック アドレスを使用します。
- ポイントツーポイント アドレス：各スイッチは 192.168/24 から始まる各ポイントツーポイント リンクで 31 ビット ネットワークを使用します。
- レイヤー 3 サーバー ゲートウェイ：IP ファブリックに接続するサーバーにはデフォルト ゲートウェイが必要です。各リーフには 172.16.1/24 で始まるゲートウェイ サービスを設定します。

トポロジーの設定

まず始めにトポロジーについて学び、各スイッチの接続方法、BGP 属性、IP アドレス スキームについて理解しましょう。各スイッチは、ホスト名、ループバック、L3 ゲートウェイ、BGP AS 番号をもちます。表 2 でこれを確認します。

表 2：BGP 実装の詳細

スイッチ	ループバック	L3 ゲートウェイ	BGP AS 番号
S1	10.0.0.1/32	None	100
S2	10.0.0.2/32	None	101
L1	10.0.0.3/32	172.16.1.1/24	200
L2	10.0.0.4/32	172.16.2.1/24	201
L3	10.0.0.5/32	172.16.3.1/24	202

インターフェイスおよび IP の設定

では各スイッチの物理接続について詳細に見ていきます。表 3 で、インターフェイス名、ポイントツーポイント ネットワーク、IP アドレスを確認します。

表 3：インターフェイスと IP 実装の詳細

送信元スイッチ	送信元インターフェイス	送信元 IP	ネットワーク	宛先スイッチ	宛先インターフェイス	宛先 IP
L1	xe-0/0/14	.1	192.168.0.0/31	S1	xe-0/0/14	.0
L1	xe-0/0/15	.7	192.168.0.6/31	S2	xe-0/0/15	.6
L2	xe-0/0/16	.3	192.168.0.2/31	S1	xe-0/0/16	.2
L2	xe-0/0/17	.8	192.168.0.8/31	S2	xe-0/0/17	.8
L3	xe-0/0/18	.11	192.168.0.10/31	S1	xe-0/0/18	.10
L3	xe-0/0/19	.1	192.168.0.0/31	S2	xe-0/0/19	.0

各リーフは各スパインと接続されていますが、スパインは互いに接続されていません。IP ファブリックでは、スパインを直接接続する必要はありません。どれか 1 つのリンクで障害が起きたとしても、すべてのリーフは互いに接続されています。もう 1 つの詳細事項は、IP ファブリックはすべてレイヤー 3 ということです。従来のレイヤー 2 ネットワークでは、適切なフラッディングとブロードキャスト ドメインのプロパゲーションのために、スパイン間の接続が必要です。レイヤー 3 IP ファブリックが望ましいもう 1 つの理由は、スパインを相互接続しなくてもよい点です。

BGP の設定

最初のステップの 1 つは、EBGP を介した各スパインを各リーフのピアとして設定することです。Junos OS で BGP 処理を高速化するコツは、すべてのネイバーを 1 つの BGP グループに入れることです。インポート ポリシーとエクスポート ポリシーは同じですが、ピアの AS とネイバーの IP のみがリーフによって異なるため、これが可能です。では、S1 の BGP 設定を見てみましょう。

```
protocols {
    bgp {
        log-updown;
        import bgp-clos-in;
        export bgp-clos-out;
        graceful-restart;
        group CLOS {
            type external;
            mtu-discovery;
            bfd-liveness-detection {
                minimum-interval 350;
                multiplier 3;
                session-mode single-hop;
            }
            multipath multiple-as;
            neighbor 192.168.0.1 {
                peer-as 200;
            }
            neighbor 192.168.0.3 {
                peer-as 201;
```

```
            }
            neighbor 192.168.0.5 {
                peer-as 202;
            }
        }
    }
}
```

各リーフには、適切な IP アドレスを含む固有のネイバー ステートメントがあります。さらに、各ネイバーには固有のピア AS があります。これによって、IP ファブリックのすべてのリーフが CLOS という 1 つの BGP グループに含まれます。

有効にすべき BGP オプションは少ないため、それらを各グループやネイバーで指定する必要はありません。

- `log-updown`：すべての BGP セッションの状態を追跡可能にします。すべてのグループとネイバーがこのオプションを継承します。これにより、各スイッチの視点から IP ファブリック全体を追跡し続けることができます。
- インポート ポリシーとエクスポート ポリシー：IP ファブリック全体で共通のインポート ポリシーとエクスポート ポリシーが使われます。リーフかスパインかによって違いはありません。ポリシー ステートメントについては、当ホワイト ペーパーの後半でより詳しく見ていきます。
- `graceful-restart`：既存のセッションを切断することなく、BGP のポリシーを変更する機能が必要です。これを可能にするため、Junos OS には `graceful-restart` 機能があります。

CLOS BGP グループ配下には、高度な機能がいくつかあります。これについて 1 つずつ見ていきます。

- `type external`：EBGP を BGP グループ全体で有効にします。IP ファブリックが EBGP 設計に基づく場合、この情報を各ネイバーで繰り返す必要はありません。
- `mtu-discovery`：物理インターフェイス上でジャンボ フレームを実行します。これにより、コントロール プレーンの更新処理において、BGP がより多くの MTU ヘルプを検出できます。
- BFD：高速コンバージェンスを可能にし、フォワーディング検出を BFD にオフロードします。この例では、3 の乗数と共に 350 ミリ秒間隔を使用しています。
- `multipath multiple-as`：ECMP を EBGP ネイバー全体で可能にするため、`multipath multiple-as` オプションを有効にする必要があります。

BGP ポリシーの設定

重要なコツは、IP ファブリック全体でプレフィックスをインポートおよびエクスポートする BGP ポリシーを記述することです。これは非常に簡単です。単純なコピー アンド ペースト操作で、スパインとリーフ全体で使用する共通の BGP ポリシーを作成できます。ではこれについて見ていきます。

まず始めは BGP エクスポート ポリシーです。

```
policy-options {
    policy-statement bgp-clos-out {
        term loopback {
            from {
                protocol direct;
                route-filter 10.0.0.0/24 orlonger;
            }
            then {
                next-hop self;
                accept;
            }
        }
        term server-L3-gw {
            from {
                protocol direct;
                route-filter 172.16.0.0/12 orlonger;
            }
            then {
                next-hop self;
```

```
                accept;
            }
        }
    }
}
```

このポリシーでは多くの項目を指定できます。各項目について見ていきます。

- `term loopback`：最初に、スイッチのループバック アドレスを識別し、それをその他すべての BGP ピアにエクスポートします。これは、10/24 またはこれより長いビットマスクに合致する、直接接続されたインターフェイスを見ることで可能です。これにより、IP ファブリック全体のすべてのループバック アドレスを簡単に識別できます。IP ファブリック全体のすべてのポイントツーポイント アドレスをプロパゲーションしないですむよう、next-hop self を使い、アドバタイズされた各プレフィックスのネクスト ホップをスイッチの送信インターフェイスに変更します。これで IP ファブリックの各スイッチは、フォワーディング テーブルのネイバー IP アドレスを直接使用します。
- `term server-L3-gw`：これまで見てきたように、各リーフには接続されたサーバー用のレイヤー 3 ゲートウェイ サービスがあります。範囲は 172.16/12 です。これは各リーフのすべてのサーバー ゲートウェイ アドレスに合致します。ここでも next-hop self を適用します。これは明らかにスパインには影響を与えず、リーフにのみ影響します。両スイッチ用に 1 つのポリシーを記述できるのは便利です。
- デフォルト：各 BGP ポリシーには最後にデフォルト項目があります。これは設定できず、EBGP のデフォルトのルールに従います。すべての EBGP および IBGP プレフィックスをネイバーにアドバタイズするか、その他すべてのプレフィックスを拒否します。これはすなわち、ルーティング テーブルの他の BGP プレフィックスが別のピアにアドバタイズされることを意味します。明示的に reject アクションを最後に設定することで、この動作を止めることができます。しかしこの場合は、IP ファブリックが全 BGP プレフィックスを全リーフにプロパゲーションするようにする必要があります。

では次にインポート ポリシー設定について見ていきます。

```
policy-options {
    policy-statement bgp-clos-in {
        term loopbacks {
            from {
                route-filter 10.0.0.0/24 orlonger;
            }
            then accept;
        }
        term server-L3-gw {
            from {
                route-filter 172.16.0.0/12 orlonger;
            }
            then accept;
        }
        term reject {
            then reject;
        }
    }
}
```

インポート ポリシーでも多くの項目を指定できます。高いレベルで、各スイッチのルーティング テーブルとフォワーディング テーブルで受け入れるプレフィックスのタイプについて自由に選択できると便利です。それでは各項目について詳しく見ていきます。

- `term loopbacks`：各スイッチがループバック アドレスを介して IP ファブリックの他のスイッチにアクセスできると便利です。明示的に 10/8 と合致させて、すべてのループバック アドレスをルーティング テーブルとフォワーディング テーブルに取り込みます。
- `term server-L3-gw`：これはサーバーのレイヤー 3 ゲートウェイ アドレスと同様です。IP ファブリックの各リーフが他のすべてのゲートウェイ アドレスを検出できる必要があります。これを可能にするため、明示的に 172.16/12 と合致させます。
- `term reject`：これについては前述のとおりです。他のすべてのプレフィックスを拒否します。問題は、reject ステートメントをインポート ポリシーの最後に設定しなかった場合、ルーティング テーブルとフォワーディング テーブルはすべてのポイントツーポイント ネットワークによって捨てられてしまうことです。この情報は各スイッチの隣のスイッチにのみ関連するため、これを各スイッチに設定する意味はありません。

ループバックとレイヤー 3 サーバー ゲートウェイをエクスポートおよびインポートするだけで、IP ファブリック全体ですべてのプレフィックスをプロパゲーションできます。便利な点は、単純にコピー アンド ペーストするだけで、IP ファブリック全体で同一のポリシー セットを再利用できることです。

ECMP の設定

前のセクションで、`multipath multiple-as` コンフィギュレーション ノブを使用したのを覚えていますか。これ単体では ECMP プレフィックスをルーティング情報ベース（RIB）（ルーティング テーブルとも呼ばれます）に設定するだけです。RIB から全 ECMP を抽出して転送情報ベース（FIB）（フォワーディング テーブルとも呼ばれます）に設定するには、ECMP を有効にする別のポリシーを作成し、FIB に設定する必要があります。ではこれについて見ていきます。

```
routing-options {
    forwarding-table {
        export PFE-LB;
    }
}
policy-options {
    policy-statement PFE-LB {
        then {
            load-balance per-packet;
        }
    }
}
```

ここでは、スイッチで転送されるパケットがロード バランシングを有効にし、それが FIB 内の全 ECMP を有効にするということが、PFE-LB ポリシーに記述されています。ただし、PFE-LB ポリシーが存在するだけでは意味がありません。FIB に直接適用する必要があります。これはルーティング オプションのフォワーディング テーブル内で行われ、PFE-LB ポリシーを参照します。

BGP の確認

IP ファブリックの設定が完了したら、次にコントロール プレーンおよびデータ プレーンが動作することを確認します。Show コマンドを使って IP ファブリックを確認し、BGP セッションの状態、やり取りされているプレフィックス、ネットワークを通過するパケットをチェックできます。

BGP 状態

では `S1` にログインし、BGP セッションを確認することから始めましょう。

```
dhanks@S1> show bgp summary
Groups: 1 Peers: 3 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0
                     6          6          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
192.168.0.1          200      12380      12334       0        3 3d 21:11:35 2/2/2/2
0/0/0/0
192.168.0.3          201      12383      12333       0        2 3d 21:11:35 2/2/2/2
0/0/0/0
192.168.0.5          202      12379      12333       0        2 3d 21:11:35 2/2/2/2
0/0/0/0
```

すべて正しく動作しています。各リーフへの各 BGP セッションが接続され、プレフィックスがやり取りされています。各セッションには 2 つの active、received、accepted プレフィックスがあり、これらはループバック アドレスとレイヤー 3 ゲートウェイ アドレスであることが分かります。ここまですべて正しく動作しています。

ではさらに詳しく見ていきましょう。コントロール プレーンの観点から、ECMP、グレースフル リスタート、BFD を確認する必要があります。たとえばこのように表示されます。

```
dhanks@S1> show bgp neighbor 192.168.0.1
Peer: 192.168.0.1+60120 AS 200 Local: 192.168.0.0+179 AS 100
  Type: External    State: Established    Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: Cease
```

```
  Export: [ bgp-clos-out ] Import: [ bgp-clos-in ]
  Options: <Preference LogUpDown PeerAS Multipath Refresh>
  Options: <MtuDiscovery MultipathAs BfdEnabled>
  Holdtime: 90 Preference: 170
  Number of flaps: 3
  Last flap event: Stop
  Error: 'Cease' Sent: 1 Recv: 1
  Peer ID: 10.0.0.3        Local ID: 10.0.0.1          Active Holdtime: 90
  Keepalive Interval: 30         Group index: 1    Peer index: 0
  BFD: enabled, up
  Local Interface: xe-0/0/14.0
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Stale routes from peer are kept for: 300
  Peer does not support Restarter functionality
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 200)
  Peer does not support Addpath
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              2
    Received prefixes:            2
    Accepted prefixes:            2
    Suppressed due to damping:    0
    Advertised prefixes:          3
  Last traffic (seconds): Received 1    Sent 25   Checked 42
  Input messages:  Total 12381Updates 3Refreshes 0Octets 235340
  Output messages: Total 12334Updates 7Refreshes 0Octets 234634
  Output Queue[0]: 0
```

重要な箇所は強調表示してあります。オプションの 2 行について詳しく見てみると、次のことが分かります。

- BGP セッションの状態が記録されている
- ECMP に対応している
- グレースフル リスタートに対応している
- MTU 検出に対応している
- BFD が BGP に関連付けられている

BGP プレフィックス

BGP 自体は正しく設定されていることが確認できたので、次に BGP の動作について確認します。S1 について詳細を確認し、どのプレフィックスが L1 にアドバタイズされているかを見てみましょう。

```
dhanks@S1> show route advertising-protocol bgp 192.168.0.1 extensive

inet.0: 53 destinations, 53 routes (52 active, 0 holddown, 1 hidden)
* 10.0.0.1/32 (1 entry, 1 announced)
 BGP group CLOS type External
     Nexthop: Self
     Flags: Nexthop Change
     AS path: [100] I

* 10.0.0.4/32 (1 entry, 1 announced)
```

```
 BGP group CLOS type External
     Nexthop: Self (rib-out 192.168.0.3)
     AS path: [100] 201 I

* 10.0.0.5/32 (1 entry, 1 announced)
 BGP group CLOS type External
     Nexthop: Self (rib-out 192.168.0.5)
     AS path: [100] 202 I

* 172.16.2.0/24 (1 entry, 1 announced)
 BGP group CLOS type External
     Nexthop: Self (rib-out 192.168.0.3)
     AS path: [100] 201 I

* 172.16.3.0/24 (1 entry, 1 announced)
 BGP group CLOS type External
     Nexthop: Self (rib-out 192.168.0.5)
     AS path: [100] 202 I
```

すべて正しく動作しているようです。S1 は 5 つのプレフィックスを L1 にアドバタイズしています。それぞれ詳しく見てみます。

- 10.0.0.1/32：S1 自身のループバック アドレスです。このプレフィックスは L1 にアドバタイズされています。
- 10.0.0.4/32：これは L2 用のループバック アドレスです。このプレフィックスは単純に L1 に転送されています。AS パスは [100] 201 I であることが分かります。これはすなわち、ルート起源は内部で、単純に AS に従って L2 に戻れることを意味します。
- 10.0.0.5/32：同様に、L3 のループバック アドレスに関するものです。L1 に転送されています。
- 172.16.2.0/24：L2 用のレイヤー 3 ゲートウェイ アドレスです。L1 に転送されています。
- 172.16.3.0/24：同様に、L3 用のレイヤー 3 ゲートウェイ アドレスに関するものです。L1 に転送されています。

では他のリーフから何を受信しているかを見てみましょう。

```
dhanks@S1> show route receive-protocol bgp 192.168.0.1

inet.0: 53 destinations, 53 routes (52 active, 0 holddown, 1 hidden)
  Prefix  Nexthop      MED     Lclpref    AS path
* 10.0.0.3/32            192.168.0.1                             200 I
* 172.16.1.0/24          192.168.0.1                             200 I

dhanks@S1> show route receive-protocol bgp 192.168.0.3

inet.0: 53 destinations, 53 routes (52 active, 0 holddown, 1 hidden)
  Prefix  Nexthop      MED     Lclpref    AS path
* 10.0.0.4/32            192.168.0.3                             201 I
* 172.16.2.0/24          192.168.0.3                             201 I

dhanks@S1> show route receive-protocol bgp 192.168.0.5

inet.0: 53 destinations, 53 routes (52 active, 0 holddown, 1 hidden)
  Prefix  Nexthop      MED     Lclpref    AS path
* 10.0.0.5/32            192.168.0.5                             202 I
* 172.16.3.0/24          192.168.0.5                             202 I
```

再度、各リーフはスパイン用のループバック アドレスとレイヤー 3 ゲートウェイ アドレスのみアドバタイズしていることが確認できます。

ルーティングテーブル

ここまで、スイッチ間でプレフィックスが正しくやり取りされていることが確認できました。次に RIB が正しく登録されているかを確認しましょう。これを確認する最も簡単な方法は、L1 にログインし、L3 のループバック アドレスへの ECMP を確認することです。

```
dhanks@L1> show route 172.16.3.1/24 exact

inet.0: 54 destinations, 58 routes (53 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both
172.16.3.0/24      *[BGP/170] 3d 10:55:14, localpref 100, from 192.168.0.6
                      AS path: 101 202 I
                     > to 192.168.0.0 via xe-0/0/14.0
                      to 192.168.0.6 via xe-0/0/15.0
                     [BGP/170] 3d 10:55:14, localpref 100
                      AS path: 100 202 I
                     > to 192.168.0.0 via xe-0/0/14.0
```

ここで確認できることは、L1 から L3 へのネクスト ホップが 2 つあるということです。これは multipath multiple-as ノブを使った適切な BGP 設定の結果です。

フォワーディング テーブル

次に、RIB でフォワーディング テーブルが正しくプログラムされているかを確認します。これまでと同様の方法で確認します。まず L1 から始め、L3 を確認します。

```
dhanks@L1> show route forwarding-table destination 172.16.3.1
Routing table: default.inet
Internet:
Destination        Type RtRef Next hop           Type Index NhRef Netif
172.16.3.0/24      user     0                    ulst 131070     5
                              192.168.0.0        ucst  1702     5 xe-0/0/14.0
                              192.168.0.6        ucst  1691     5 xe-0/0/15.0
```

ここで確認できるのは、S1 (xe-0/0/14) 向け、および S2 (xe-0/0/15) 向けの 2 つのネクスト ホップです。

Ping

データ プレーン接続を確認する単純な方法は、L1 にログインし、レイヤー 2 ゲートウェイ アドレスから ping source し、L3 に ping する方法です。これによりネットワークのスパインを介したトラフィックが強制実行されます。

```
dhanks@L1> ping source 172.16.1.1 172.16.3.1 count 5
PING 172.16.3.1 (172.16.3.1): 56 data bytes
64 bytes from 172.16.3.1: icmp_seq=0 ttl=63 time=3.009 ms
64 bytes from 172.16.3.1: icmp_seq=1 ttl=63 time=2.163 ms
64 bytes from 172.16.3.1: icmp_seq=2 ttl=63 time=2.243 ms
64 bytes from 172.16.3.1: icmp_seq=3 ttl=63 time=2.302 ms
64 bytes from 172.16.3.1: icmp_seq=4 ttl=63 time=1.723 ms

--- 172.16.3.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.723/2.288/3.009/0.414 ms
```

ここまで順調です。ここでのコツは、レイヤー 3 ゲートウェイ アドレスから ping source することです。これにより、L3 は L1 へのリターン ルートをもつことが分かります。

Traceroute

さらに詳細に確認するには、traceroute を使います。トラフィックが IP ファブリックのスパインを介して移動することが確認できます。これを混ぜ合わせ、代わりにループバック アドレスを使用します。

```
dhanks@L1> traceroute source 10.0.0.3 10.0.0.5
traceroute to 10.0.0.5 (10.0.0.5) from 10.0.0.3, 30 hops max, 40 byte packets
 1  192.168.0.6http://bb06-cclab-lo0.spglab.juniper.net/ (192.168.0.6)  2.031 ms
1.932 ms 192.168.0.0 (192.168.0.0)  2.121 ms
 2  10.0.0.5 (10.0.0.5)  2.339 ms  2.342 ms  2.196 ms
```

ここで興味深いのは、traceroute は S2 を通り、L3 に達することが確認できることです。これは、L1 のフォワーディング テーブルによって traceroute トラフィックがどのようにハッシュ化されたかを示しています。

設定

IP ファブリックを構築し、このラボを基盤として使いたい場合、次の設定を使用してください。ページ数に制限があるため、1 つのスパインとリーフ スイッチについて示します。賢明なみなさんは、残りを理解することができるでしょう。

S1

```
interfaces {
    xe-0/0/14 {
        mtu 9216;
        unit 0 {
            family inet {
                mtu 9000;
                address 192.168.0.0/31;
            }
        }
    }
    xe-0/0/16 {
        mtu 9216;
        unit 0 {
            family inet {
                mtu 9000;
                address 192.168.0.2/31;
            }
        }
    }
    xe-0/0/18 {
        mtu 9216;
        unit 0 {
            family inet {
                mtu 9000;
                address 192.168.0.4/31;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.1/32;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.1;
    autonomous-system 100;
    forwarding-table {
```

```
        export PFE-LB;
    }
}
protocols {
    bgp {
        log-updown;
        import bgp-clos-in;
        export bgp-clos-out;
        graceful-restart;
        group CLOS {
            type external;
            mtu-discovery;
            bfd-liveness-detection {
                minimum-interval 350;
                multiplier 3;
                session-mode single-hop;
            }
            multipath multiple-as;
            neighbor 192.168.0.1 {
                peer-as 200;
            }
            neighbor 192.168.0.3 {
                peer-as 201;
            }
            neighbor 192.168.0.5 {
                peer-as 202;
            }
        }
    }
policy-options {
    policy-statement PFE-LB {
        then {
            load-balance per-packet;
        }
    }
    policy-statement bgp-clos-in {
        term loopbacks {
            from {
                route-filter 10.0.0.0/24 orlonger;
            }
            then accept;
        }
        term server-L3-gw {
            from {
                route-filter 172.16.0.0/12 orlonger;
            }
            then accept;
        }
        term reject {
            then reject;
        }
    }
    policy-statement bgp-clos-out {
        term loopback {
            from {
                protocol direct;
                route-filter 10.0.0.0/24 orlonger;
            }
            then {
                next-hop self;
```

```
                accept;
            }
        }
        term server-L3-gw {
            from {
                protocol direct;
                route-filter 172.16.0.0/12 orlonger;
            }
            then {
                next-hop self;
                accept;
            }
        }
    }
}
```

L1

```
interfaces {
    interface-range ALL-SERVER {
        member-range xe-0/0/0 to xe-0/0/13;
        member-range xe-0/0/16 to xe-0/0/47;
        unit 0 {
            family ethernet-switching {
                interface-mode access;
                vlan {
                    members SERVER;
                }
            }
        }
    }
    xe-0/0/14 {
        mtu 9216;
        unit 0 {
            family inet {
                mtu 9000;
                address 192.168.0.1/31;
            }
        }
    }
    xe-0/0/15 {
        mtu 9216;
        unit 0 {
            family inet {
                mtu 9000;
                address 192.168.0.7/31;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.3/32;
            }
        }
    }
    irb
        mtu 9216;
        unit 1 {
```

```
            family inet {
                mtu 9000;
                address 172.16.1.1/24;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.3;
    autonomous-system 200;
    forwarding-table {
        export PFE-LB;
    }
}
protocols {
    bgp {
        log-updown;
        import bgp-clos-in;
        export bgp-clos-out;
        graceful-restart;
        group CLOS {
            type external;
            mtu-discovery;
            bfd-liveness-detection {
                minimum-interval 350;
                multiplier 3;
                session-mode single-hop;
            }
            multipath multiple-as;
            neighbor 192.168.0.0 {
                peer-as 100;
            }
            neighbor 192.168.0.6 {
                peer-as 101;
            }
        }
    }
}
policy-options {
    policy-statement PFE-LB {
        then {
            load-balance per-packet;
        }
    }
    policy-statement bgp-clos-in {
        term loopbacks {
            from {
                route-filter 10.0.0.0/24 orlonger;
            }
            then accept;
        }
        term server-L3-gw {
            from {
                route-filter 172.16.0.0/12 orlonger;
            }
            then accept;
        }
        term reject {
            then reject;
        }
```

```
    }
    policy-statement bgp-clos-out {
        term loopback {
            from {
                protocol direct;
                route-filter 10.0.0.0/24 orlonger;
            }
            then {
                next-hop self;
                accept;
            }
        }
        term server-L3-gw {
            from {
                protocol direct;
                route-filter 172.16.0.0/12 orlonger;
            }
            then {
                next-hop self;
                accept;
            }
        }
        term reject {
            then reject;
        }
    }
}
vlans {
    SERVER {
        vlan-id 1;
        l3-interface irb.1;
    }
}
```

結論

当ホワイト ペーパーでは、IP ファブリックを構築する基本的な方法について説明しました。さらに重要な点は、IP ファブリックを構築する際に考慮すべき判断ポイントについても確認したということです。プラットフォームで必要な機能に影響を与えるコントロール プレーンには様々なオプションがあります。最後に、IP ファブリックに BGP を実装する方法について詳細に確認しました。すべてのインターフェイス、IP アドレス、BGP 設定、およびポリシーについて確認しました。まとめとして、BGP が IP ファブリック全体で動作していることを確認しました。また、データ プレーンでテストを行い、トラフィックがリーフからリーフに転送されていることを確認しました。

IP ファブリックの構築は非常に簡単で、VMware NSX やジュニパーネットワークス Contrail などのオーバーレイ技術の優れた基盤として機能します。レイヤー 3 のみの設計を基盤とすることで、IP ファブリックの障害回復力が高まり、BFD により非常に高速なエンドツーエンド コンバージェンスが実現します。ぜひ次回の IP ファブリックは、QFX5100 スイッチを使って構築してください。

その他の詳細については、2014 年第 4 四半期にオライリー メディアから発行される Juniper Networks Technical Library 書籍『The QFX Series』をご覧ください。この本の中で、当ホワイト ペーパーの内容が詳しく紹介される予定です。Technical Library のすべての本について、**www.juniper.net/books** から参照できます。

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、ネットワーク革新に取り組んでいます。デバイスからデータ センターまで、そしてコンシューマーからクラウド プロバイダまで、ジュニパーネットワークスが提供するソフトウェア、シリコン、システムは、ネットワークのエクスペリエンスと経済性を変革します。ジュニパーネットワークスは、世界中のお客様とパートナー企業のために尽力しています。詳細については、**www.juniper.net/jp/** をご覧ください。

日本

ジュニパーネットワークス株式会社

東京本社
〒163-1445
東京都新宿区西新宿3-20-2
東京オペラシティタワー45F
電話　03-5333-7400
FAX　03-5333-7401

西日本事務所
〒541-0041
大阪府大阪市中央区北浜1-1-27
グランクリュ大阪北浜

米国本社

Juniper Networks, Inc.

1194 North Mathilda Avenue
Sunnyvale, CA 94089
USA

電話　888-JUNIPER
（888-586-4737）
または 408-745-2000
FAX　408-745-2100

URL　http://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

URL http://www.juniper.net/jp/

2000565-002-JP 2014 年 3 月