APPLICATION NOTE

ブランチ向けSRXシリーズおよびJシリーズのシャーシクラスタ

ブランチ向けSRXシリーズ サービス・ゲートウェイおよびJシリーズ サービスルーターでのシャーシクラスタの設定

目次

図一覧

はじめに

最新のネットワークでは、高可用性が求められています。この要件に対処するため、ジュニパーネットワークスSRXシリーズ サービス・ゲートウェイおよびJシリーズ サービスルーターは、クラスタモードで動作するよう設定できます。クラスタモードでは、デバイスのペアを相互接続して、単一のノードとして動作するよう設定できます。この設定により、デバイス、インタフェース、およびサービスレベルで冗長性を実現します。ジュニパーネットワークスJunos® OSリリース9.0以降では、シャーシクラスタ機能によるHA（High Availability）対応のジュニパーネットワークスJシリーズ サービスルーターおよびSRXシリーズ サービス・ゲートウェイを導入できます。Jシリーズでは、この機能を利用できるのは、Junos OSのフロー対応バージョンに限定されます。Junos OSリリース9.5では、ブランチ向けSRXシリーズ サービス・ゲートウェイの導入に伴い、あらゆるブランチ向けSRXシリーズ デバイスでHAがサポートされています。

本書の目的

本書の目的として、制限事項や設計上の考慮事項とともに、HAシャーシクラスタ機能について確認することが挙げられます。また、一般的なユースケースや、ジュニパーネットワークスScreenOS®ソフトウェアNSRP（NetScreen Redundancy Protocol）の同様の機能との関連性についても説明します。

設計上の考慮事項

デバイス間の高可用性は、エンタープライズ環境の設計に簡単に組み込むことができます。特に関連性があるのは、大規模な企業オフィス環境に支店・営業所のサイトを接続する設計事例です。HA機能を活用することで、エンタープライズ環境では、デバイスや接続の障害が発生した場合でも接続を維持できます。

ハードウェア要件

- クラスタ単位に同じモデルのJシリーズ セキュアルーターが2台（ジュニパーネットワークスJ2320サービスルーター、J2350サービスルーター、J4350サービスルーター、J6350サービスルーター）または
- クラスタ単位に同じモデルのSRXシリーズ ゲートウェイが2台（ジュニパーネットワークスSRX100サービス・ゲートウェイ、SRX110サービス・ゲートウェイ、SRX210サービス・ゲートウェイ、SRX220サービス・ゲートウェイ、SRX240サービス・ゲートウェイ、SRX550サービス・ゲートウェイ、SRX650サービス・ゲートウェイ）

ソフトウェア要件

- フロー対応のJunos OS 9.0以降（Jシリーズ セキュアルーターの場合）
- Junos OSリリース9.5以降（SRXシリーズ サービス・ゲートウェイの場合）

説明と導入シナリオ

デバイス間のシャーシクラスタは、アクティブ/パッシブ型またはアクティブ/アクティブ型のどちらのシナリオでも導入できます。Junos OSでは、非対称ルーティングのシナリオにHAクラスタを追加して使用できます。本書全体にわたって、コードの例を紹介するとともに、導入シナリオについて説明しています。

機能説明

HA機能は、ジュニパーネットワークスMシリーズ マルチサービスエッジルーターおよびTシリーズ コアルーターに最初に導入された冗長性機能に続いてモデル化されました。まず、Junos OSの冗長性の仕組みについて簡単に説明します。この説明を踏まえて、デバイスのクラスタ化に際して、このモデルがどのように適用されるか理解を深めます。Junos OSはコントロールプレーンとデータプレーンを分離する設計を採用しているので、冗長性はこの両方のプレーンで機能する必要があります。Junos OSでは、コントロールプレーンはRE（Routing Engine）によって管理されます。REは、（さまざまな機能の中でも特に）ルーティングおよび転送の計算全般を実行します。コントロールプレーンが集約されると、Junos ルーター上で実装されている各PFE（Packet Forwarding Engine）に転送エントリがプッシュされます。次に、PFEがルートベースのルックアップを実行して、REに依存することなく、各パケットの適切な宛先を決定します。図1に、このJunos OSの転送パラダイムを簡潔にまとめています。

Junos OSの冗長性モデル

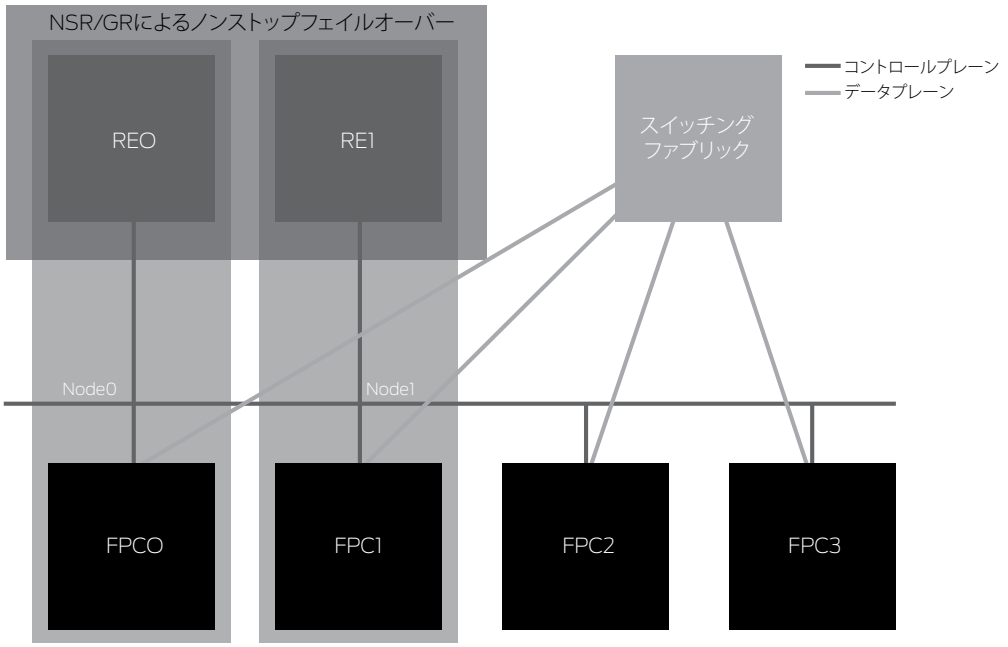

図1：Junos OSの冗長性モデル

Junos OSでは、グレースフルリスタートやNSR（Nonstop Active Routing）により、コントロールプレーンのフェイルオーバーが可能です。前者の場合、ルーターは（コントロールプレーンの障害は転送プレーンに影響を及ぼさないので）データプレーンでトラフィックの転送を続行しながら、コントロールプレーンの障害をネットワークに通知します。再起動中のルーターが新しい隣接関係を形成している間も、ネットワークでは、再起動中のルーターの使用が継続されます（Grace Peroid）。このシナリオでは、バックアップREで構成全体が検出されますが、コントロールプレーンの実行状態は検出されません。障害時には、バックアップREでルーティング/転送テーブルをすべて再計算する必要があります。ノンストップルーティングでは、ルーティングエンジン間のステートレプリケーション機能が使用されます。この場合、バックアップREはネットワークからの支援を受けずにルーターをコントロールするので、再起動中のルーターがコントロールプレーンの障害を透過的に処理します。ルーティングプロトコルがデータプレーンの障害を処理する一方で、従来のルーティングプロトコル、VRRP（Virtual Router Redundancy Protocol）、または集約インタフェースを使用して、他のインタフェースからのトラフィックを迂回することで、インタフェース、PFE、およびFPCのフェイルオーバーが処理されます。Jシリーズ ルーターでシャーシクラスタを実現する際に、Junos OSでは、同様のモデル（ただし、ノンストップルーティングステートレプリケーションを除く）を採用して、図2に示すように、コントロールプレーンの冗長性を確保しています。

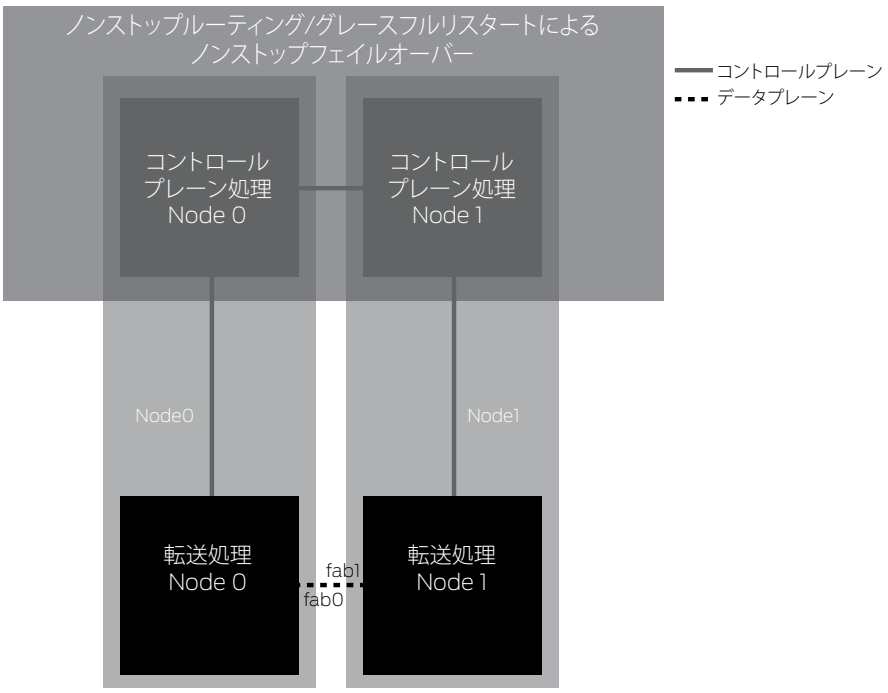

図2：デバイスクラスタ

シャーシクラスタ機能では、2台のデバイスのクラスタ化をサポートしており、前の図に示すように、デバイス間に2つの接続を必要とします。シャーシクラスタは、外部デバイスからもクラスタ管理者からも、一つのデバイスとして認識されます。クラスタ機能を有効にすると、node 0との衝突を回避するため、クラスタのnode 1側のインタフェース番号が再設定されます。使用するモデルに応じて（クラスタ化できるのは、同じモデルのデバイスが2台という制限）、システムFPCの合計数をインタフェースの元のFPC数に加算する形で、node 1側のインタフェース番号が再設定されます（SRXシリーズでは、オンボードのポートと、FPCに対応する各PIM（Physical Interface Module）スロット）。したがって、2台のSRXシリーズをクラスタ化する場合、J2320にはFPC0として機能するシステムボードに3つのPIMスロットと4つの標準GbEポートが搭載されているので、node 1のインタフェース番号はge-4/0/0～ge-7/0/0として再設定されます。以下の表に、インタフェース番号の再設定の体系をまとめています。

表1：インタフェース番号の再設定

デバイス	番号の再設定用の定数	node 0のインスタンス名	node 1のインスタンス名
J2320	4	ge-0/0/0	ge-4/0/0
J2350	5	ge-0/0/0	ge-6/0/0
J4350	7	ge-0/0/0	ge-7/0/0
J6350	7	ge-0/0/0	ge-7/0/0
SRX100/SRX110	1	fe-0/0/0	fe-1/0/0
SRX210	2	ge-0/0/0	ge-2/0/0
SRX220	3	ge-0/0/0	ge-3/0/0
SRX240	5	ge-0/0/0	ge-5/0/0
SRX650	9	ge-0/0/0	ge-9/0/0

クラスタ機能が有効になると、fxp0、fxp1、およびfabインタフェースがシステムによって作成されます。プラットフォームに応じて、fxp0とfxp1が物理インタフェースにマッピングされます。このマッピングは、ユーザー側で設定できません。fabインタフェースは、ユーザー側で設定できます。以下の表に、fxp0とfxp1のマッピングをまとめています。

表2：インタフェースfxp0およびfxp1のマッピング

デバイス	fxp0インタフェース	fxp1インタフェース	fabインタフェース
J2320	ge-0/0/2	ge-0/0/3	ユーザー定義
J2350	ge-0/0/2	ge-0/0/3	ユーザー定義
J4350	ge-0/0/2	ge-0/0/3	ユーザー定義
J6350	ge-0/0/2	ge-0/0/3	ユーザー定義
SRX100/SRX110	fe-0/0/6	fe-0/0/7	ユーザー定義
SRX210	ge-0/0/0	fe-0/0/7	ユーザー定義
SRX220	ge-0/0/6	ge-0/0/7	ユーザー定義
SRX240	ge-0/0/0	ge-0/0/1	ユーザー定義
SRX550	ge-0/0/0	ge-0/0/1	ユーザー定義
SRX650	ge-0/0/0	ge-0/0/1	ユーザー定義

図2で確認したように、fxp1（HAリンク）は、クラスタ内のノード間でコントロールプレーンの通信を可能にします。fxp0は管理アクセスを可能にしますが、対象はホストトラフィックに限定されます。つまり、イーサネットインタフェースであることが前提です（WANインタフェースはサポート対象外）。fxp0インタフェースから受信されたトラフィックは、システム内の他のインタフェースには転送されません。fabインタフェースは、データプレーン情報とトラフィックをデバイス間で交換する目的で使用されます。fxp0およびfxp1インタフェースとは異なり、fabインタフェースはシステム内の任意のイーサネットインタフェースにマッピングできます。

クラスタのコントロールプレーンの冗長性は、単一のMシリーズおよびTシリーズルーターで使用される場合と同様です。各デバイスは、冗長RE構成のシステムでルーティングエンジンとして機能します。グレースフルリスタートは、コントロールプレーンのフェイルオーバーを可能にし、ネットワーク上でトラフィックの影響を最小限に抑える目的で使用されます。コントロールプレーンの冗長性モデルは、アクティブ/パッシブ型です。この場合、クラスタ内のノードがアクティブデバイスとして指定され、クラスタルーティングの計算全般を実行します。クラスタの管理に必要ないくつかの重要な処理を除いて、大半の処理はマスターRE上だけで実行されます。プライマリノードで障害が発生すると、バックアップデバイスでルーティング処理とその他の処理がアクティブになり、コントロールプレーンの動作を引き継ぎます。

データプレーンの冗長性では、関係する要素がさらに増えます。ジュニパーネットワークスMシリーズおよびTシリーズルーターは、パケット単位にトラフィック転送を実行します。フローの概念は存在せず、アクティブなREから配信された転送テーブルのコピーを各PFEで保持しています。転送テーブルにより、各PFEは他のシステムのPFEに依存せずに、トラフィック転送を実行できます。PFEで障害が発生した場合でも、システム内の他のPFEは影響を受けず、コントロールプレーンは有効なPFEにトラフィックを再ルーティングできます。これに対して、Jシリーズ セキュアルーターおよびSRXシリーズ サービス・ゲートウェイでは、あらゆるトラフィックを検査して、全アクティブセッションのテーブルを保持しています。システムから新しい接続が許可されるたびに、デバイスは、特定の接続を識別する5-tuple（ソースおよび宛先のIPアドレス、ソースおよび宛先のポート、該当する場合、プロトコル）を記録します。さらに、ネクストホップ、セッションタイムアウト、シーケンス番号（プロトコルがTCPの場合）といったセッションの詳細情報や、不明な/望ましくないプロトコル（またはユーザー）からパケットが転送されていないことを保証する上で必要な、その他のセッション固有の情報でテーブルを更新します。トラフィックがデバイスを通過すると、セッション情報が更新されます。このセッション情報は、フェイルオーバーが発生したときに、確立済みのセッションがドロップされないようにするため、クラスタ内の両方のデバイスで必要になります。

図1に示すように、コントロールプレーンのREはアクティブ/バックアップモードで機能するのに対して、データプレーン（PFE）はアクティブ/アクティブモードで機能します。アクティブ/アクティブモードのPFEでは、トラフィックを一方のノードのクラスタで受信することも、もう一方のノードから送信することも可能です。つまり、両方のノードでセッションの作成と同期に対応できる必要があることを意味します。たとえば、初期セッションを記録していないノードにリターントラフィックが非対称に到達した場合、シャーシクラスタ機能では、トラフィックを元のノードに転送して処理することで、セキュリティ機能の侵害を防ぎます。

リダンダントイーサネットインターフェイス

前述のように、コントロールプレーンの障害がメンバーノードによって検出されると、バックアップノードがクラスタのコントロールを引き継ぎます。これに対して、データプレーンの障害時には、ルーティングプロトコルによるトラフィックの再ルーティングか、リダンダントイーサネットインターフェイスによるインタフェース障害の対処に依存しています。冗長イーサネットの概念は、極めてシンプルです。2つのイーサネットインタフェース（クラスタ内の各ノードから1つ）が同じリダンダントイーサネットインターフェイス（Junos OS用語では、RETHインタフェース）の一部として設定されます。さらに、RETHインタフェースは、リダンダントグループの一部として設定されます。リダンダントグループがアクティブになるのは、クラスタ内のいずれか1つのノード上に限定されます。このリダンダントグループのメンバーであるリダンダントイーサネットインターフェイスは、必ずアクティブノード上の物理インタフェースを経由して、トラフィックの送信を（通常は受信も）実行します。

リダンダントグループは、1つまたは複数の物理インタフェースを監視するよう設定できます。監視対象の各インタフェースには、ウェイトが割り当てられます。インタフェースの障害が発生した場合に、リダンダントグループのしきい値から、このウェイトが引かれます。インタフェースのフェイルオーバーにより、しきい値がゼロを下回った場合、リダンダントグループで状態の移行が発生して、クラスタ内の他のノードがリダンダントグループでアクティブになります。その結果、このリダンダントグループに含まれているリダンダントイーサネットインターフェイスはすべて、新しいノード上のインタフェースを使用してトラフィックの送信を（通常は受信も）実行します。したがって、障害時にもトラフィックのルーティングが維持されます。NSRPについて熟知している読者の方であれば、RETHインタフェースがジュニパーネットワークスScreenOS®のソフトウェアベースのデバイスVSI（Virtual Security Interface）と同様の機能であることをおわかりいただけるでしょう。RETHインタフェースは、VSIと同様に、VSI/RETHのメンバーである異なる物理インタフェース間で同じIPアドレスとMAC（Media Access Control）アドレスを共有します。リダンダントインタフェースはフェイルオーバー時にGratuitous ARP（Address Resolution Protocol）メッセージを送信し、ネットワークでは単一のインタフェースとして認識されます。ただし、RETHとVSIでは、以下に示すように、いくつかの大きな違いがあります。

- RETHインタフェースは常に、同じタイプの物理イーサネットインタフェース（fe-feやge-geなど）を含んでいます。
- アクティブなVSIの物理インタフェースがダウンした場合、VSIは常にフェイルオーバーを強制します。リダンダントイーサネットインターフェイスの状態は、RETHが関連付けられているリダンダントグループの機能状態とそのまま一致します。RETHインタフェースのアクティブな物理インタフェースがダウンした場合、RETHインタフェースはダウンします。

受信/送信インタフェースのタイプに関係なく、セッション情報は同期されます。従来のルーティングプロトコルは、障害時のルーティング用に使用できます。そして、ルーティングプロトコルをサポートしていないシンプルなデバイスへの接続時には、この制約を克服するため、リダンダントイーサネットインターフェイスが役立ちます。

リンクアグリゲーションインタフェースとLACP

Junos OS 11.2以降では、RETHインタフェースに、LAGインタフェースグループをメンバーとして含めることができます。さらに、LAGグループに含まれている物理インタフェースは、SRXシリーズ シャーシクラスタのクロスメンバーとして機能できます。この機能により、クラスタメンバー間で複数のアクティブな物理インタフェースがRETH（Redundant Ethernet）および冗長プロトコル（JSRP）に参加できます。

リモートパフォーマンス監視

Junos OSベースのデバイスには、RPM（Remote Performance Monitoring）を実行する機能があります。RPMは、ICMP、TCP、またはHTTPを使用してホストを監視するルーター上で動作するタスクであり、リモートホストを定期的にチェックして、パケットの損失や遅延の結果を含むログの履歴を保持します。この情報は、HA設計のアップストリームルーターを監視する目的で使用できます。さらに、IPモニタリングとの組み合わせで、（バックアップ）インタフェースを有効にしたり、RPMからのプローブ結果に基づいてアクティブルーティングテーブルに変更を加えたりすることが可能になります。

IPモニタリング

IPモニタリングは、ScreenOSのTrack-IP機能に相当するJunos OSの機能です。この機能により、SRXシリーズ デバイスはアップストリームホストを監視し、RPMで監視されているホストの可用性に基づいて、ルーティングテーブルに動的に変更を加えることが可能になります。

機能サポートと比較表

NSRPとJSRP（Junos OSで使用されるプロトコル）は、どちらも同じサービスを提供する目的で設計されたプロトコルですが、動作の仕組みも、提供する機能セットも異なります。この両プロトコルの主な違いについて、以下の表にまとめています。

表3：機能比較

機能	JSRP	NSRP
セッションレプリケーション	はい	はい
ALG（Application-Level Gateway）レプリケーション	はい	はい
NAT（Network Address Translation）セッションレプリケーション	はい	はい
IPsecセッションレプリケーション（ポリシーベースVPN）	はい	はい
IPsecセッションレプリケーション（ルートベースVPN）	はい	はい
ルート同期	N/A	はい
インタフェースモニタリング	はい	はい
ゾーンモニタリング	いいえ	はい
IP追跡/IPモニタリング	はい	はい
RPM（Remote Performance Monitoring）	はい	いいえ
非対称ルーティング	はい	いいえ
負荷分散	はい	いいえ
グレースフルリスタート	はい	いいえ
レイヤー2モード	はい	はい

クラスタ構成

このセクションでは、Jシリーズのシャーシクラスタを設定する場合に必要な手順の概要について説明します。ステップ1から3は、最低限必要な手順です。この最低限の設定後は、2台のJシリーズ セキュアルーターが単一のデバイスとして機能し、この両方のノードの全インタフェースをコントロールします。ステップ4から6では、管理インタフェース（fxp0）のIPアドレスと各クラスタノードのホスト名を指定する場合に必要な設定ステートメントについて詳しく説明します（node 0とnode 1には、異なる管理IPアドレスとホスト名が設定されます）。ステップ7では、リダンダントイーサネットインターフェイスとその関連するリダンダントグループを追加する場合に必要な設定について説明します。

この例では、J2320デバイスのペアであるnode leftとnode rightでシャーシクラスタを有効にするという想定です。node leftとnode rightは、インタフェースge-0/0/1およびge-0/0/3を使用してバックツーバック接続されています。

注：Factory Configを使用している場合、インタフェースユニットの削除、VLANの削除、およびセキュリティゾーンの変更（またはセキュリティの削除）が必要になります。

1.FXPおよびHAコントロールリンクとFabリンクのインタフェースで使用される設定をすべて削除します。以下の例では、SRX210デバイス用のインタフェース名とデフォルト設定を使用しています。

```
On both nodes:
root@left> cli
root@left# set chassis cluster cluster-id 1 node 0 reboot
root@left# delete interface interface-range interfaces-trust members ge-0/0/1
root@left# delete interface interface-range interfaces-trust members ge-0/0/2
root@left# delete interface ge-0/0/0
root@left# delete security zone security-zone trust interface ge-0/0/0

set interface fab0 fabric
```

各デバイスにログインして、EEPROMに適切なクラスタIDを設定することでクラスタ機能を有効にします。この設定を有効にするには、再起動が必要です。現在の実装はクラスタ内の2つのノードに限定されているので、設定できるのは、node 0とnode 1だけです。この例では、表1に示すように、node 0（left）とnode 1（right）の番号が再設定されます。

```
set chassis cluster cluster-id <n> node <m> reboot
On node left:
root@left> set chassis cluster cluster-id 1 node 0 reboot
On node right:
root@right> set chassis cluster cluster-id 1 node 1 reboot
```

注：ステップ1は、設定モードではなく、運用モードで実行する必要があります。

ノードの再起動後、ノードによってクラスタが形成されます。これ以降、クラスタの設定はノードメンバー間で同期されます。以下のコマンドは、いずれかのデバイスで設定モードから入力します。

再起動後、CLIに移動したときに、プロンプトの変化に注意してください。

2.fab接続に使用するインタフェースを定義します。該当するインタフェースでは、図2に示すように、バックツーバック接続か、レイヤー2インフラストラクチャ経由での接続が必要になります。当然、fab0がnode0のファブリックインタフェースであるのに対して、fab1がnode1のファブリックインタフェースです。

```
set interface fab0 fabric-options member-interfaces <interface>
set interface fab1 fabric-options member-interfaces <interface>
```

3.設定グループを使用して、各デバイスの管理インタフェースを設定します。

```
set groups node0 system host-name <node0 hostname>
set groups node0 interfaces fxp0 unit 0 family inet address <node0 mgmt
IP>/<netmask>
set groups node1 system host-name <node1 hostname>
set groups node1 interfaces fxp0 unit 0 family inet address <node1 mgmt
IP>/<netmask>
```

4.（オプション）デバイス固有のオプションを設定します。

```
set groups node0 snmp description <node0 snmp sysDesc>
set groups node1 snmp description <node1 snmp sysDesc>
```

5.グループ設定を適用します。

```
set apply-groups "${node}"
```

6.（オプション）リダンダントイーサネットインターフェイスを使用する場合、リダンダントグループとRETHインタフェースを定義します。

```
set chassis cluster reth-count <n>
set chassis cluster redundancy-group 1 node 0 priority <n>
set chassis cluster redundancy-group 1 node 1 priority <n>
set interface <interface name> gigether-options redundant-parent reth.<n>
```

この結果のサンプル設定を以下に示します。

```
#The following declares int ge-0/0/1 in node 0 as the fab interface for the node
set interface fab0 fabric-options member-interfaces ge-0/0/1
#The following declares int ge-4/0/1 in node 1 as the fab interface for the node
set interface fab1 fabric-options member-interfaces ge-4/0/1
#Groups configuration.Configuration parameters specific to each node are set here.
set groups node0 system host-name left
set groups node0 interfaces fxp0 unit 0 family inet address 192.168.3.10/24
set groups node1 system host-name right
set groups node1 interfaces fxp0 unit 0 family inet address 192.168.3.11/24
set apply-groups “${node}”
#Define a single RETH interface for the cluster
set chassis cluster reth-count 1
#Define node 0 as the primary node for reth0
set chassis cluster redundancy-group 1 node 0 priority 100
set chassis cluster redundancy-group 1 node 1 priority 1
#Add interfaces ge-0/0/0 (in node 0) and ge-4/0/0 (ge-0/0/0 in node 1) to the
reth
set interface ge-0/0/0 gigether-options redundant-parent reth0
set interface ge-4/0/0 gigether-options redundant-parent reth0
set interfaces reth0 unit 0 family inet address <reth0-ip-address>
set interfaces reth1 redundant-ether-options redundancy-group <rg-id>
#Define node 0 as the primary node for the control path
set chassis cluster redundancy-group 0 node 0 priority 100
set chassis cluster redundancy-group 0 node 1 priority 1
```

シャーシクラスタの無効化

シャーシクラスタは、簡単に無効化できます。最初に各ノードのクラスタIDを0に設定してから、ノードを再起動します。

```
set chassis cluster cluster-id 0 node 0 reboot
```

クラスタモニタリング

以下のコマンドは、クラスタのステータスを確認して、ノード側から見たクラスタの状況を表示する場合に使用できます。統計情報はクラスタ内のノード間で同期されません。クラスタのデバッグ時には、各メンバーノードにログインして、それぞれの出力を分析することを推奨します。

シャーシクラスタのステータスの表示

以下のコマンドは、クラスタ内に設定されている各リダンダントグループを、各リダンダントグループに指定されているプライオリティと、各ノードのステータスとともに表示します。このコマンドは、各ノードでアクティブなRETHインタフェースを確認したい場合に役立ちます。特別なリダンダントグループ0は、コントロールプレーンのステータスを参照します。この例では、node 0がこのリダンダントグループのプライマリノードです。したがって、node 0がコントロールプレーンの計算全般を処理します（マスターREとして機能し、rpd、kmd、dhcpd、pppdといったコントロールプレーンの処理を実行します）。

```
show chassis cluster status
Cluster:1, Redundancy-Group:0
    Device name             Priority    Status    Preempt  Manual failover

    node0                   100         Primary   No       No
    node1                   1           Secondary No       No

Cluster:1, Redundancy-Group:1
    Device name             Priority    Status    Preempt  Manual failover

    node0                   100         Primary   Yes      No
    node1                   1           Secondary Yes      No
```

クラスタ統計情報の表示

以下のコマンドは、同期対象の各オブジェクトの統計情報、ファブリックおよびコントロールインタフェースのhelloメッセージ、およびクラスタ内で監視されているインタフェースのステータスを表示します。

```
show chassis cluster statistics
Initial hold:5

Reth Information:
    reth     status      redundancy-group
    reth0    up          1
Services Synchronized:
    Service-name                              Rtos-sent    Rtos-received
    Translation Context                       0            0
    Incoming NAT                              0            0
    Resource Manager                          10           0
    Session-create                            225          10592
    Session-close                             222          10390
    Session-change                            0            0
    Gate-create                               0            0
    Session-Ageout-refresh-request            149          1
    Session-Ageout-refresh-reply              0            0
    VPN                                       0            0
    Firewall User Authentication              0            0
    MGCP Alg                                  0            0
    H323 Alg                                  0            0
    SIP Alg                                   0            0
    SCCP Alg                                  0            0
    PPTP Alg                                  0            0
    RTSP Alg                                  0            0
Interface Monitoring:
    Interface         Weight    Status    Redundancy-group
    ge-4/0/0          255       up        1
    ge-0/0/0          255       up        1
    fe-5/0/0          255       up        1
    fe-1/0/0          255       up        1
chassis-cluster interfaces:
    Control link: up
    244800 heart beats sent
    244764 heart beats received
    1000 ms interval
    3  threshold
chassis-cluster interfaces:
    Fabric link: up
    244786 heartbeat packets sent on fabric-link interface
    244764 heartbeat packets received on fabric-link interface
```

コントロールリンクのステータスの表示

このコマンドは、特定のノードのコントロールインタフェース（fxp1）のステータスを表示します。

```
show chassis cluster interface
Physical Interface: fxp1.0, Enabled, Control interface, Physical link is Up
```

セッションの表示

以下に示すコマンドは、ノード番号を指定することで、各ノードのセッションテーブルに含まれているセッションを表示します。同期されたセッションは両方のノードで認識され、一方のノードではアクティブ、もう一方のノードではバックアップとして表示されます。セッションIDを指定することで、セッションの詳細なビューを表示できます。

```
show security flow session node0

Session ID:2, Policy name: self-traffic-policy/1, State:Active, Timeout:1800
  In:172.24.241.53/50045 --> 172.19.101.34/22;tcp, If: ge-0/0/0.0
  Out:172.19.101.34/22 --> 172.24.241.53/50045;tcp, If:.local..0

1 sessions displayed


show security flow session session-identifier 2
Session ID:2, Status:Normal, State:Active
Flag:0x40
Virtual system: root, Policy name: self-traffic-policy/1
Maximum timeout:1800, Current timeout:1800
Start time:1900, Duration:256
   In:172.24.241.53/50045 --> 172.19.101.34/22;tcp,
    Interface: ge-0/0/0.0,
    Session token:0xa, Flag:0x4097
    Route:0x20010, Gateway:172.19.101.1, Tunnel:0
    Port sequence:0, FIN sequence:0,
    FIN state:0,
   Out:172.19.101.34/22 --> 172.24.241.53/50045;tcp,
    Interface:.local..0,
    Session token:0x4, Flag:0x4112
    Route:0xffffb0006, Gateway:172.19.101.34, Tunnel:0
    Port sequence:0, FIN sequence:0,
    FIN state:0,

1 sessions displayed
```

TCPシーケンス番号は同期されません。ただし、特定のセッションのアクティブノードで、シーケンス番号が追跡されます。障害が原因でセッションが移行する場合（たとえば、セッション/セッションのグループの送信インタフェースが障害前と異なるノードに移行するような場合）、シーケンス番号のカウントは、セッションの新しいアクティブノードを通過するパケットのシーケンス番号に基づいて、新しいノードで再開します。

導入シナリオ

NSRPがいくつかのトポロジーに基づいて複数のネットワークで使用されています。このセクションでは、このような一般的なシナリオ向けに、SRXシリーズ サービス・ゲートウェイまたはJシリーズ ルーターで同等の機能を実現します。

アクティブ/パッシブ型クラスタ

この場合、クラスタ内の単一のデバイスがあらゆるトラフィックのルーティングに使用され、その他のデバイスは障害時に限定して使用されます。障害が発生した場合には、バックアップデバイスがマスターになり、あらゆる転送処理を引き継ぎます。

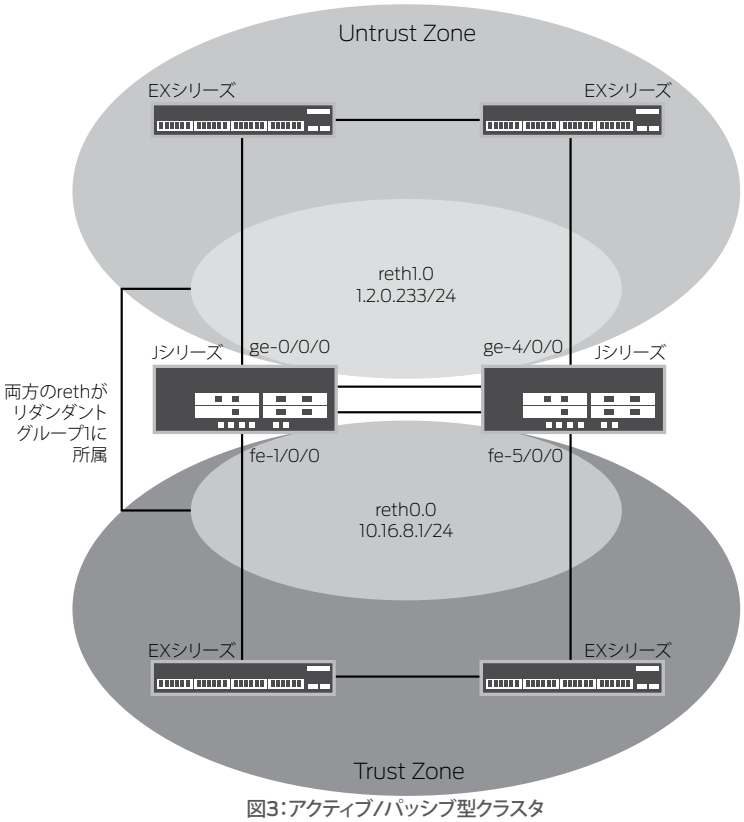

図3：アクティブ/パッシブ型クラスタ

アクティブ/パッシブ型は、VSIを使用する場合と同様に、RETHインタフェースを使用して実現します。リダンダントグループは、reth0およびreth1の物理インタフェースの状態を監視することで、RETH状態を判別します。このいずれかのインタフェースで障害が発生した場合、障害元のインタフェースをホスティングしているシステムによって、グループは非アクティブであると宣言されます。両方のRETHインタフェースが同じリダンダントグループに属しているので、障害時には、両方のRETHインタフェースが同時にフェイルオーバーを実行します。この設定により、トラフィックを転送するのは常にクラスタ内の1つのノードに限定されるので、ファブリックリンクのトラフィックは最小限に抑えられます。

```
#Groups Definitions
set groups node0 system host-name J2320-A
set groups node0 interfaces fxp0 unit 0 family inet address 192.168.3.110/24
set groups node1 system host-name J2320-B
set groups node1 interfaces fxp0 unit 0 family inet address 192.168.3.111/24
set apply-groups “${node}”

#Cluster Configuration, redundancy-group 0 determines the status of the RE
mastership, while redundancy-group 1 is used to control the reth interfaces
set chassis cluster reth-count 2
set chassis cluster heartbeat-threshold 3
set chassis cluster node 0
set chassis cluster node 1
set chassis cluster redundancy-group 0 node 0 priority 100
set chassis cluster redundancy-group 0 node 1 priority 1

#The ge-0/0/0 interface on each node is used as the fabric interface between the
nodes
set interfaces fab0 fabric-options member-interfaces ge-0/0/1
set interfaces fab1 fabric-options member-interfaces ge-4/0/1
```

```
#Note how the redundancy-group 1 is configured to monitor all the physical
interfaces forwarding traffic.The preempt keyword causes the mastership to be
reverted back to the primary node for the group (node 0, which has a higher
priority) when the failing interface causing the switchover comes back up
set chassis cluster redundancy-group 1 node 0 priority 100
set chassis cluster redundancy-group 1 node 1 priority 1
set chassis cluster redundancy-group 1 preempt
set chassis cluster redundancy-group 1 interface-monitor fe-1/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor fe-5/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-0/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-4/0/0 weight 255

#(Optionally) If both data processing and control plane functions want to
be performed in the same node, then redundancy-group 0 must monitor also
the physical interfaces.If control and data planes are allowed to fail over
independently, the following four commands should not be set.
set chassis cluster redundancy-group 0 interface-monitor fe-1/0/0 weight 255
set chassis cluster redundancy-group 0 interface-monitor fe-5/0/0 weight 255
set chassis cluster redundancy-group 0 interface-monitor ge-0/0/0 weight 255
set chassis cluster redundancy-group 0 interface-monitor ge-4/0/0 weight 255

set interfaces ge-0/0/0 gigether-options redundant-parent reth1
set interfaces fe-1/0/0 fastether-options redundant-parent reth0
set interfaces ge-4/0/0 gigether-options redundant-parent reth1
set interfaces fe-5/0/0 fastether-options redundant-parent reth0
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth1 redundant-ether-options redundancy-group 1

#Just as regular interfaces, reth interfaces must be part of a security zone
set security zones security-zone Untrust interfaces reth1.0
set security zones security-zone Trust interfaces reth0.0
```

非対称ルーティングのシナリオ

このシナリオでは、Junos OS with enhanced servicesの非対称ルーティング機能を利用します。ノードで受信されたトラフィックは、そのノードのセッションテーブルに対してマッチングされます。このルックアップの結果は、そのノードでセッションを処理するか、ファブリックリンクを介して他のノードに転送するかどうかを意味します。これで、セッションはクラスタ内の任意のデバイスに固定可能になり、セッションテーブルが複製されている限り、トラフィックは適切に処理されます。ファブリックトラフィックを最小限に抑えるため、特定の接続の送信インタフェースをホスティングしているノードに、セッションは常に固定されます。

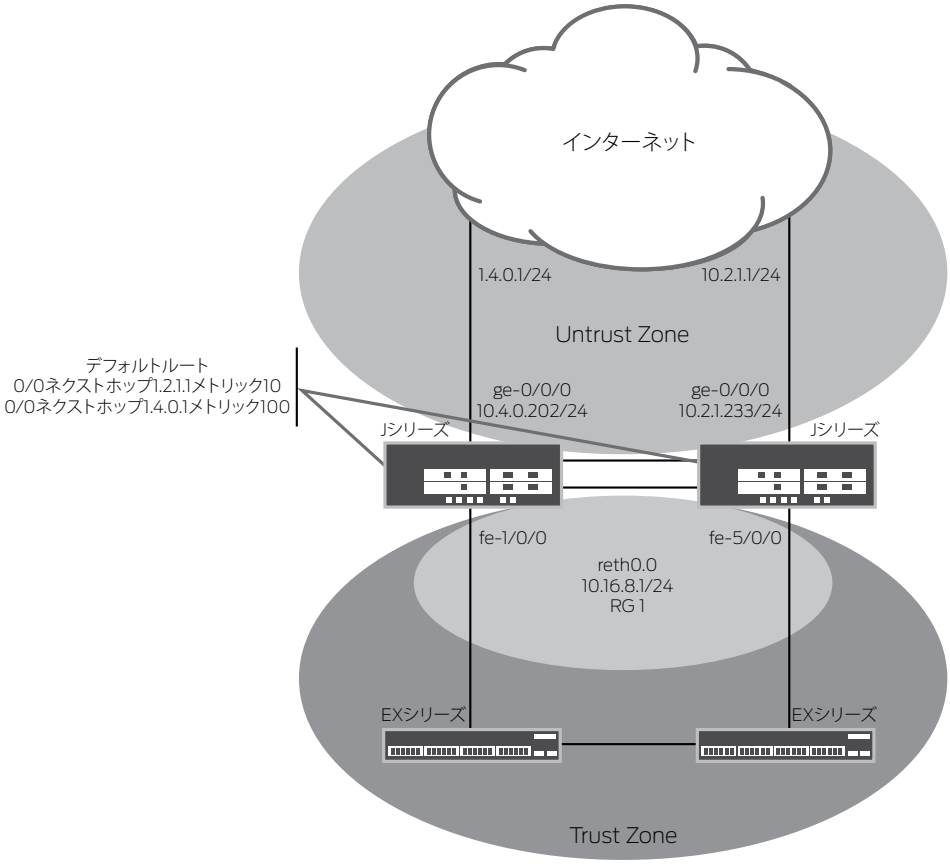

図4：非対称ルーティングのシナリオ

図4は、非対称ルーティングをサポートする仕組みの例を示しています。このシナリオでは、2つのインターネット接続を使用しており、その一方を優先接続として設定しています。Trust Zoneへの接続では、RETHインタフェースを使用して、Trust Zone内のデバイス向けにLANの冗長性を実現しています。説明のため、2つのフェイルオーバーのケースを取り上げます。それぞれのケースでは、セッションがTrust Zoneで開始され、インターネット（Untrust Zone）を宛先に設定しています。

ケースI：Trust Zone RETHでの障害

通常の運用条件では、トラフィックはTrust Zoneからnode 0のインタフェースge-0/0/0（reth0.0に所属）に送信されます。プライマリインターネット接続はnode 0に存在するので、セッションはnode 0とnode 1の両方で作成されますが、（各セッションの送信インタフェースはすべて、node 0に所属のfe-1/0/0であることから）アクティブになるのはnode 0のセッションだけです。

ge-0/0/0インタフェースで障害が発生すると、リダンダントグループのフェイルオーバーが発生し、インタフェースge-4/0/0（node 1のge-0/0/0）がアクティブになります。このフェイルオーバー後に、トラフィックはnode 1に到達します。（送信インタフェースfe-1/0/0がnode 0でホスティングされることから）セッションはnode 0でアクティブになるので、セッションルックアップによって、トラフィックはnode 0に送信されます。node 0でトラフィックが処理されて、インターネットに転送されます。リターントラフィックも同様に処理されます。トラフィックがnode 0に到達すると、（セッションはnode 0に固定されていることから）node 0で処理されて、ファブリックインタフェースを介してnode 1に送信されます。node 1では、ge-4/0/0インタフェースを介してトラフィックが転送されます。

ケースII：Untrust Zoneインタフェースでの障害

このケースは、前のケースとは異なり、セッションがノード間で移行されます。前の例と同様に、通常の運用条件では、トラフィックはnode 0だけで処理されます。インターネットに接続しているインタフェースfe-1/0/0で障害が発生すると、ルーティングテーブルに変更が加えられて、障害後のデフォルトルートとしてnode 1のインタフェースfe-5/0/0を参照するようになります。障害後、（送信インタフェースはnode 1に移行することから）node 0のセッションは非アクティブになり、node 1のバックアップセッションがアクティブになります。Trust Zoneから到達したトラフィックは、インタフェースge-0/0/0で受信されますが、node 1に転送されて処理されます。node 1で処理されたトラフィックは、インタフェースfe-5/0/0を介してインターネットに転送されます。

このシナリオでソースNATを使用していた場合、送信セッションのNAT処理はフェイルオーバー後に変化するので、別のプロバイダによって割り当てられるアドレス空間には、上記の仕組みでは対処できないことに注意してください（これはHA実装の制約ではありませんが、2つのISP（Internet Service Provider）を利用している場合、結果的にユーザーはパブリックなアドレス空間を保持しておらず、いずれかのISPで障害が発生すると、その障害元のサービスプロバイダに属しているIPからの接続はすべて失われます）。

```
#Cluster Configuration, redundancy-group 1 is used to control the RETH interface
connected to the trust zone.Note how the redundancy group (and therefore reth0)
will only failover if either fe-1/0/0 or fe-5/0/0 fail, but not if any of the
interfaces connected to the Internet fails.
set chassis cluster reth-count 1
set chassis cluster node 0
set chassis cluster node 1
set chassis cluster redundancy-group 1 node 0 priority 100
set chassis cluster redundancy-group 1 node 1 priority 1
set chassis cluster redundancy-group 1 preempt
set chassis cluster redundancy-group 1 interface-monitor fe-1/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor fe-5/0/0 weight 255

#Interface Definitions
set interfaces ge-0/0/0 unit 0 family inet address 1.4.0.202/24
set interfaces fe-1/0/0 fastether-options redundant-parent reth0
set interfaces fe-1/0/1 disable
set interfaces ge-4/0/0 unit 0 family inet address 1.2.1.233/24
set interfaces fe-5/0/0 fastether-options redundant-parent reth0
set interfaces reth0 unit 0 family inet address 10.16.8.1/24

#ge-0/0/1 one each node will be used for the fab interfaces
set interfaces fab0 fabric-options member-interfaces ge-0/0/1
set interfaces fab1 fabric-options member-interfaces ge-4/0/1

#We have two static routes, one to each ISP, but the preferred one is through ge-
0/0/0
set routing-options static route 0.0.0.0/0 qualified-next-hop 1.4.0.1 metric 10
set routing-options static route 0.0.0.0/0 qualified-next-hop 1.2.1.1 metric 100
#Zones Definitions
set security zones security-zone Untrust interfaces ge-0/0/0.0 host-inbound-
traffic system-services dhcp
set security zones security-zone Untrust interfaces ge-4/0/0.0 host-inbound-
traffic system-services dhcp
set security zones security-zone Trust interfaces reth0.0

#Finally a permit all security policy  from Trust to Untrust zone
set security policies from-zone Trust to-zone Untrust policy ANY match source-
address any
set security policies from-zone Trust to-zone Untrust policy ANY match
destination-address any
set security policies from-zone Trust to-zone Untrust policy ANY match
application any
set security policies from-zone Trust to-zone Untrust policy ANY then permit
```

アクティブ/アクティブ型のフルメッシュ

このシナリオは、中～大規模の導入事例で採用され、ソースルーターがルーターの2つのペアの間に配置されます。クラスタ内のノードからのトラフィックフローをコントロールする目的で、OSPFが使用されます。また、2つのノード間のセッションを同期する目的で、JSRPが使用されます。非対称ルーティングがサポートされていることから、トラフィックを双方向に特定のノードに転送する処理は必須ではありません。障害が発生して、セッション作成元のノードと異なるノードからセッションのリターントラフィックが到着した場合、アクティブなセッションが存在するノードにトラフィックを返送する目的でfabリンクが使用されます（このノードが特定のセッションの送信インタフェースをホスティングすることになります）。

このシナリオでは、デバイス間のフルメッシュ接続を利用することで、ネットワークの障害許容力が高まるというメリットがもたらされます。同時に、ファイアウォールとルーターの間にスイッチを追加する必要がなくなり、ネットワークの障害点が減少するという効果があります。

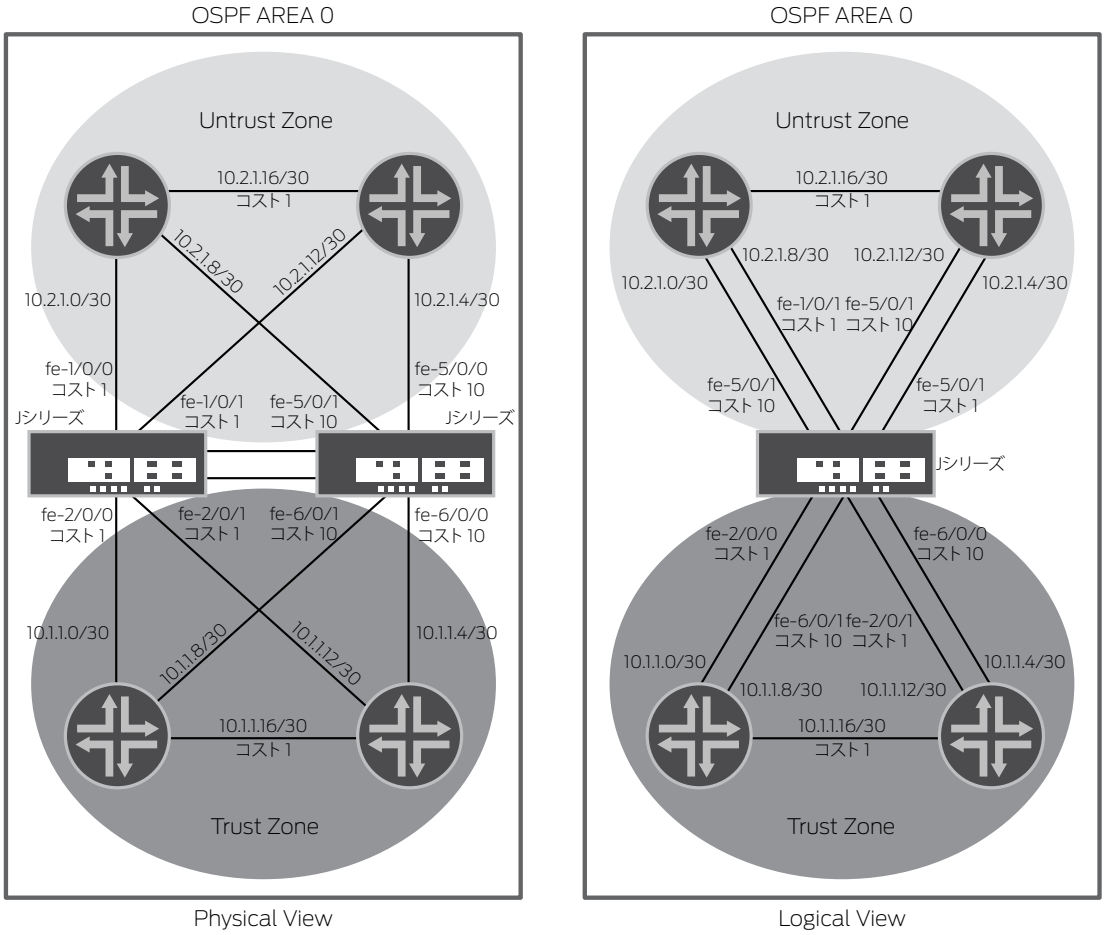

図5：アクティブ/アクティブ型のフルメッシュシナリオ

特別な考慮事項

Junos OS with enhanced servicesでシャーシクラスタ機能を使用する場合、以下の設計上の考慮事項について検討する必要があります。

- fabリンクまたはfxp1リンクの（両方ではなく）いずれか一方でエラーが発生すると、バックアップノードは無効になります（シングルポイントオブフェイラー）。バックアップノードがfabリンクとfxp1リンクの両方でエラーを検出した場合、バックアップノードはマスターになります。
- コントロールリンクの障害時には、システムはファブリックリンクを監視することで、デュアルマスターシップのシナリオを回避しようと試みます。このファブリックリンクを介してhelloメッセージが受信された場合は、セカンダリは無効になり、プライマリはアクティブな状態を維持します。コントロールリンクとファブリックリンクのどちらでもhelloメッセージが受信されなかった場合は、バックアップノードがアクティブになります。
- ファブリックリンクの障害が検出されたときには、コントロールリンクの障害時と同様に、ノードはスプリットブレインの回避手順を実行します。ファブリックリンクに障害が発生したときに、コントロールリンクがまだ動作している場合は、バックアップノードは無効になり、2つのマスターによる競合は避けられます。
- フェイルオーバーの時間は、ほぼ数秒です。（helloメッセージの最短時間は1000msであり、helloメッセージが連続で失われる回数を条件とするしきい値の最小値は3であることから）障害の検出には3秒以上かかります。
- ハイエンドモデルでは ISSU light（In-Service Software Upgrade light）をサポートしています。またブランチモデルではICU（In-band cluster upgrade）をサポートしています（一台ずつアップグレードする手順については次のセクションを参照）。
- シャーシクラスタ機能は、パケットモードベースのプロトコルをサポートしていません（MPLS、コネクションレス型ネットワークサービスはサポート対象外）。
- シャーシクラスタ機能の使用時には、Pseudo Interfaceはサポートされていません。Pseudo Interfaceを必要とする以下のサービスは、クラスタ構成では機能しません。

- MLPPP（Multilink Point-to-Point Protocol）、MLFR（Multilink Frame Relay）、CRTP（Compressed RTP）などのリンクサービス
- GRE（Generic Routing Encapsulation）トンネル
- IP/IPトンネル
- IPv4マルチキャスト
- WANインタフェースは、以下を例外として、サポートされています。
  - CH-T1、ISDN、およびxDSL
  - ISM 200モジュールは、HAモードではサポートされていません。*

注：ISMモジュールがサポートされているのは、Jシリーズに限定されます。

クラスタのアップグレード**

クラスタは簡単な手順でアップグレードできます。ただし、アップグレードの実行中に、サービスが3～5分間、中断することに注意してください。

1.新しいイメージファイルをnode 0にロードします。

2.Junos OS CLIから"request system software add <image name>"と入力して、ノードを再起動せずに、イメージアップグレードを実行します。

3.新しいイメージファイルをnode 1にロードします。

4.ステップ2で説明した手順どおりに、node 1のイメージアップグレードを実行します。

5.両方のノードを同時に再起動します。

注：より簡単な手順のLISSU、ICUをサポートしています。

シャーシクラスタのin-bandマネジメント

従来、SRXシリーズ クラスタを管理するには、管理ポートへの専用アクセスが必要なout-of-bandマネジメントネットワークを使用するしか方法がなく、レベニュートラフィックの転送には使用できませんでした。このセクションでは、in-bandマネジメント接続を使用してSRXシリーズ クラスタを管理・導入する場合に推奨される方法について詳しく見ていきます。

ブランチ向けSRXシリーズ サービス・ゲートウェイは、クラスタ構成で導入した場合、（fxp0インタフェースを介して）帯域内または帯域外で管理できます。この場合、クラスタにはレベニューポートだけを介して管理ステーションから到達できると想定しています。

問題の説明

SRXシリーズ ゲートウェイのJunos OSで利用可能なHA（High Availability）機能は、Junos OSベースのルーターに採用された冗長性機能に基づいてモデル化されています。コントロールプレーンとデータプレーンを分離する設計により、Junos OSベースのルーターは両方のプレーンで冗長性を実現しています。Junos OSでは、コントロールプレーンはルーティングエンジンによって管理されます。ルーティングエンジンは、（さまざまな機能の中でも特に）ルーティングおよび転送の計算全般を実行します。コントロールプレーンの集約後は、システム内の各PFE（Packet Forwarding Engine）に転送エントリがプッシュされます。次に、PFEがルートベースのルックアップを実行して、ルーティングエンジンによる操作を必要とせずに、各パケットの適切な宛先を決定します。

SRXシリーズ ゲートウェイでシャーシクラスタを有効化する場合、図6と同じモデルを使用して、コントロールプレーンの冗長性を実現します。

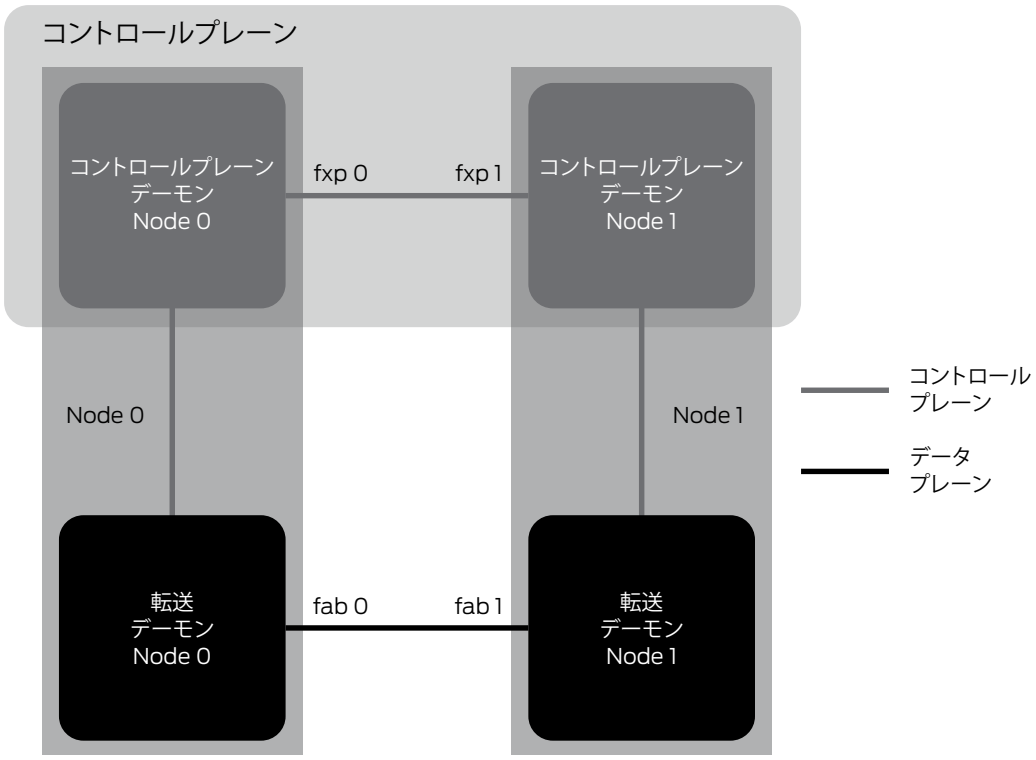

図6：SRXシリーズ クラスタ化モデル

2つのルーティングエンジンを搭載したルーターと同様に、SRXシリーズ クラスタのコントロールプレーンはアクティブ/パッシブモードで動作し、常に一方のノードだけがコントロールプレーンをアクティブに管理します。このため、転送プレーンはコントロールプレーンに送信されたすべてのトラフィック（ホストインバウンドトラフィック）を常にクラスタのプライマリノードに転送します。このトラフィックの例を以下に示します（ただし、これに限定されるわけではありません）。

- ルーティングデーモン用のトラフィック（BGPトラフィック、OSPF、IS-IS、RIP、PIMなど）
- IKE（Internet Key Exchange）ネゴシエーションメッセージ
- 管理デーモンにリダイレクトされるトラフィック（SSH、Telnet、SNMP、Netconf（NSM用）など）
- 監視プロトコル（BFD（Bidirectional Forwarding Detection (BFD)やRPM（Real-time Performance Monitoring）など）

この処理が適用されるのは、ホストインバウンドトラフィックに限定されることに注意してください。スルートラフィック（クラスタによって転送されるトラフィックのうち、クラスタのいずれかのインタフェースが宛先に指定されていないもの）は、クラスタの設定に基づいて、どちらのノードでも処理できます。

転送プレーンは常にホストインバウンドトラフィックをプライマリノードにリダイレクトするので、コントロールプレーンのステータスに関係なく、各ノードへの独立した接続を可能にするため、新しいタイプのインタフェースとしてfxp0を追加しました。fxp0インタフェースに送信されるトラフィックは、転送プレーンでは処理されません。ただし、Junos OSカーネルに送信されて、（セカンダリノードも含む）ノードのコントロールプレーンへの接続を可能にします。

Junos OS 10.1r2までは、NSM（およびその他の管理インタフェース）によるシャーシクラスタの管理では、クラスタの両方のメンバーのコントロールプレーンへの接続が必要でした。したがって、各ノードのfxp0インタフェースへのアクセスを必要としていました。

本書では、fxp0インタフェースを必要とせずに、プライマリノードからシャーシクラスタを管理する方法について説明します。

説明と導入シナリオ

SSH/Telnetによるクラスタへの接続

クラスタのプライマリノードへのアクセスは、ノードのいずれかのインタフェース（つまり、fxp0以外のインタフェース）への接続を確立する場合と同様に簡単です。L3またはRETH（Redundant Ethernet）インタフェースは常にトラフィックを（ノードの種類を問わず）プライマリノードにリダイレクトします。どちらの導入シナリオも一般的であり、詳細を以下の図に示します。

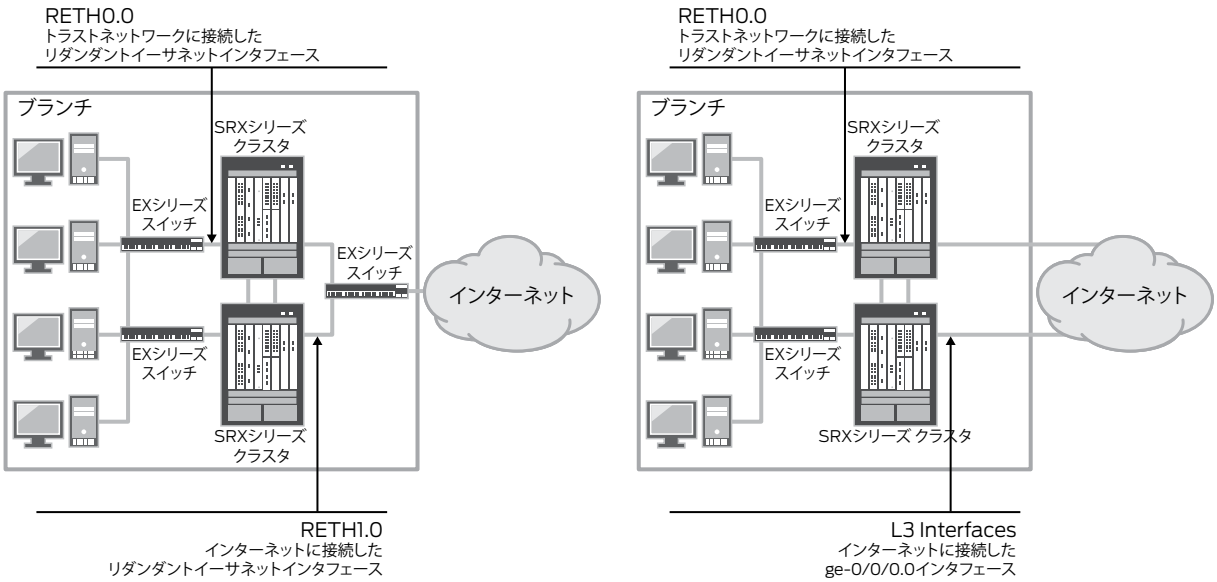

図7：一般的なブランチでのSRXシリーズ クラスタの導入シナリオ

どちらのケースでも、いずれかのローカルアドレスへの接続を確立することで、プライマリノードに接続します（厳密には、リダンダントグループ0のプライマリノードに接続）。たとえば、リダンダントグループ1のメンバーであるRETHインタフェースが別のノードでアクティブである場合にも、プライマリノードに接続できます（同じことは、L3インタフェースがバックアップノードに物理的に存在している場合にも当てはまります）。

```
$ssh 10.1.1.34
labuser@10.1.1.34's password:
--- JUNOS 10.2R1.3 built 2010-05-14 15:13:40 UTC
{primary:node1}
labuser@BranchGW> show chassis cluster status
Cluster ID:3
Node                  Priority          Status    Preempt  Manual failover

Redundancy group:0 , Failover count:3
    node0                   200         secondary      no       yes
    node1                   255         primary        no       yes

Redundancy group:1 , Failover count:4
    node0                   254         primary        yes      no
    node1                   1           secondary      yes      no
```

プライマリノードからセカンダリノードへのログイン

ほとんどの監視コマンドは、両方のノードのステータスを表示します。必要な場合には、以下に示すように、プライマリノードからセカンダリノードに接続することも可能です。

```
labuser@BranchGW> request routing-engine login node 0
--- JUNOS 10.2R1.3 built 2010-05-14 15:13:40 UTC
```

```
{secondary:node0}
Exiting the session will bring us back to the primary node:
{secondary:node0}
labuser@BranchGW> exit
rlogin: connection closed
{primary:node1}
labuser@BranchGW>
```

管理プロトコルの動作全般を確認する上で、クラスタのSSH管理は、わかりやすい例です。プライマリノードへの接続はシンプルであり、セカンダリノードへの接続はプライマリを介して実行する必要があります。

クラスタのNSM管理もまったく同じです。2010.2よりも前のバージョンのNSMでは、両方のノードへのNETCONF接続が必要でした。このことが、古いバージョンでクラスタのin-bandマネジメントに問題が発生した理由です。この問題の解決策については、次のセクションで取り上げます。

ネットワークおよびセキュリティマネージャによるin-bandマネジメント

クラスタ構成のSRXシリーズ ゲートウェイのNSM管理は、NSRP（NetScreen Redundancy Protocol）を使用して接続されたScreenOSデバイスの管理に基づいてモデル化されました。このモデルでは、HAペアを形成する各メンバーにNSMが個別に接続します。ただし、HAモードで動作している他のJunos OSベースのデバイスは、単一の接続を使用してNSMから管理できます。特に、NSMは、マスターノードだけに接続することで、バーチャルシャーシテクノロジーを採用したジュニパーネットワークスEXシリーズ イーサネットスイッチを管理できます。この場合、シャーシの設定および監視は、この単一の接続を使用して実行されます。

NSMバージョン2010.2では、バーチャルシャーシテクノロジーを採用したEXシリーズと同様に、ブランチ向けSRXシリーズ クラスタを管理する機能が追加され、必要とされるのはプライマリノードへの単一の接続だけになりました。この変更に伴い、デバイスがNSMにバーチャルシャーシとして識別されるよう、デバイスとNSMの両方に変更が必要になりました。後方互換性を保つため、デフォルトでは、クラスタはNSMにシャーシクラスタとして識別され、fxp0インタフェースを介して管理されるという想定です。

デフォルトの動作は、以下の設定をクラスタに追加することで、デバイスで変更できます。

```
labuser@BranchGW# set chassis cluster network-management cluster-master
```

NSMへのデバイスの追加は、EXシリーズのバーチャルシャーシを追加する場合と同様です。クラスタを追加するときに[virtual-chassis]チェックボックスをオンにするだけです。クラスタをシャーシクラスタとしてではなく、単一のノードとして追加する必要があるので、その方法に注意してください。

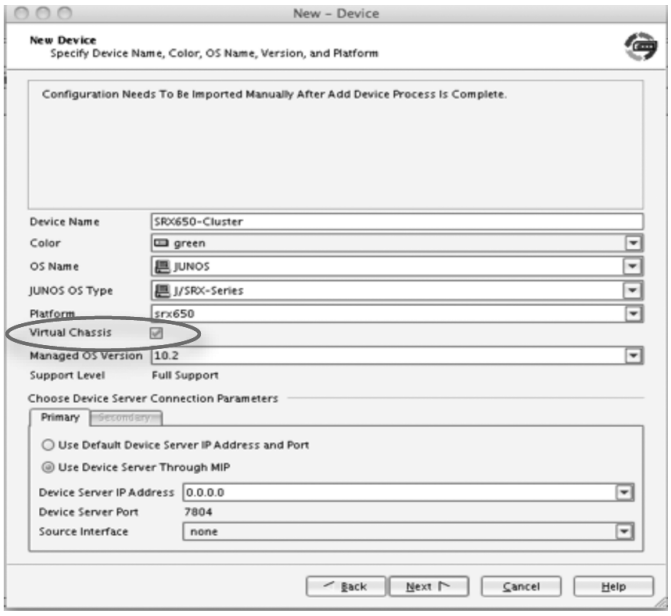

図8：NSMのバーチャルシャーシとしてのクラスタの追加

ハードウェアインベントリには、プライマリノードのシャーシのシリアル番号が表示されます。フェイルオーバーにより、更新が実行され、シリアル番号の変更が反映されます。

以下の例外（今後のリリースで解決される予定）を除いて、ほとんどの設定および監視オプションがサポートされています。

- シャーシインベントリには、"FPC"ではなく、"sub-component"が表示されます。
- NSMにキャッシュされたコピーから"get-system-information"によって取得される"chassis serial number"には古い情報が含まれており、正確ではありません。
- NSMによる両方のデバイスのソフトウェア更新はサポートされていません。
- バーチャルシャーシのステータスビューには、有効な情報が表示されません。
- ライセンスインベントリには、プライマリノードについての情報だけが表示されます。
- プライマリノードが再起動すると、ハードウェアインベントリの同期がずれます。
- NSMから送信される再起動コマンドは、プライマリノードだけ適用されます。
- プライマリノードからはコントロールプレーンのログだけがNSMに送信されます。
- データプレーンのログ（セッションログ、IDP攻撃など）は、structured-syslog形式で両方のノードからNSMに直接送信できます。NSMでのstructured-syslogメッセージのサポートには、2010.4R2以降のバージョンが必要です。

IDPシグネチャの更新時に、NSMはセキュリティパッケージをプライマリノードにプッシュします。その後、RPC（Remote Procedure Call）をクラスタに送信して、アップグレードをトリガします。通常の状況下では、プライマリノードだけが更新されます。この制約を克服するため、プライマリノードの更新後にセカンダリノードの自動更新を可能にするJunos OSスクリプトが開発されました。

IDPシグネチャの更新

シャーシクラスタが帯域内接続から管理されている場合、プライマリノードのコントロールプレーンだけが他のデバイスに接続できる状態になります。特に、更新サーバーから新しいセキュリティパッケージをダウンロードできるのは、プライマリノードだけです。

"request security idp security-package download node primary"および"request security idp security-package install node primary"コマンドは、現在でもプライマリノードにセキュリティノードをダウンロードしてインストールする場合に使用できます（ノードを指定せずに、この2つのコマンドを使用すると、プライマリノードではそのままでも機能しますが、セカンダリノードでは処理に失敗します）。

クラスタは"idp-update.xslt"イベントスクリプトをロードして有効にすることで、新しくインストールされたセキュリティパッケージをセカンダリノードに自動的にコピーしてインストールできます。スクリプトは、両方のノードの"/var/db/scripts/event"ディレクトリにコピーする必要があります。さらに、コピー後は、以下の設定を使用して有効にする必要があります。

```
set event-options policy idp-update events IDP_SECURITY_INSTALL_RESULT
set event-options policy idp-update attributes-match idp_security_install_result.
status matches successful
set event-options policy idp-update then event-script idp-update.xslt
```

スクリプトを有効にすると、NSM、CLI（Command-Line Interface）、自動更新など、IDPシグネチャの更新方法がすべてサポートされます。

ノード間でシグネチャパッケージを手動で同期させることも可能です。具体的には、プライマリノードの/var/db/idpd/sec-downloadディレクトリ内のファイルをセカンダリノードにコピーします。ノード間でファイルをコピーするには、"file copy"コマンドを使用して、バックアップノードをターゲットとして指定します（file copy /var/db/idpd/sec-download/<filename> nodeX:/var/db/idpd/sec-download）。この場合、nodeXには、バックアップに該当するノードに応じて、node0またはnode1を指定します。

同様に、IDPポリシーテンプレートを同期させるには、/var/db/scripts/commitディレクトリに格納されているテンプレートをセカンダリノードにそのままコピーします。

SNMPの使用

SSH/Telnetのケースと同様に、プライマリデバイスはSNMPクエリに応答して、両方のノードのSNMPトラップを生成できます。本書の執筆時点では、ブランチ向けSRXシリーズ デバイスでサポートされているMIBのうち、一部を除いて、ほとんどのMIBはクラスタで問題なく機能します。

例として、クラスタのインタフェース記述についてのクエリを実行すると、両方のノードのインタフェースを含むリストが返されます。

```
[labuser@centos-1 ~]$ snmpwalk -v 2c -c public 10.1.1.34 ifDescr
IF-MIB::ifDescr.1 = STRING: fxp0
IF-MIB::ifDescr.2 = STRING: fxp1
IF-MIB::ifDescr.4 = STRING: lsi
IF-MIB::ifDescr.5 = STRING: dsc
IF-MIB::ifDescr.6 = STRING: lo0
IF-MIB::ifDescr.7 = STRING: tap
IF-MIB::ifDescr.8 = STRING: gre
IF-MIB::ifDescr.9 = STRING: ipip
IF-MIB::ifDescr.10 = STRING: pime
IF-MIB::ifDescr.11 = STRING: pimd
IF-MIB::ifDescr.12 = STRING: mtun
IF-MIB::ifDescr.13 = STRING: fxp0.0
IF-MIB::ifDescr.14 = STRING: fxp1.0
IF-MIB::ifDescr.21 = STRING: lo0.16384
IF-MIB::ifDescr.22 = STRING: lo0.16385
IF-MIB::ifDescr.116 = STRING: pp0
IF-MIB::ifDescr.123 = STRING: st0
IF-MIB::ifDescr.159 = STRING: reth1.0
IF-MIB::ifDescr.160 = STRING: reth0.0
IF-MIB::ifDescr.162 = STRING: reth0
IF-MIB::ifDescr.163 = STRING: reth1
IF-MIB::ifDescr.172 = STRING: vlan
IF-MIB::ifDescr.501 = STRING: ge-0/0/0
IF-MIB::ifDescr.502 = STRING: ge-0/0/1
IF-MIB::ifDescr.503 = STRING: ge-0/0/1.0
IF-MIB::ifDescr.504 = STRING: ge-3/0/0
IF-MIB::ifDescr.505 = STRING: ge-3/0/0.0
IF-MIB::ifDescr.506 = STRING: ge-3/0/1
IF-MIB::ifDescr.507 = STRING: ge-3/0/1.0
IF-MIB::ifDescr.508 = STRING: ge-3/0/2
IF-MIB::ifDescr.509 = STRING: ge-3/0/3
IF-MIB::ifDescr.510 = STRING: ge-3/0/4
IF-MIB::ifDescr.511 = STRING: ge-3/0/5
IF-MIB::ifDescr.512 = STRING: ge-3/0/6
IF-MIB::ifDescr.513 = STRING: ge-3/0/7
IF-MIB::ifDescr.514 = STRING: fab1.0
IF-MIB::ifDescr.515 = STRING: fab1
IF-MIB::ifDescr.516 = STRING: ge-4/0/0
IF-MIB::ifDescr.517 = STRING: ge-4/0/1
IF-MIB::ifDescr.518 = STRING: ge-4/0/1.0
IF-MIB::ifDescr.519 = STRING: ge-7/0/0
IF-MIB::ifDescr.520 = STRING: ge-7/0/1
IF-MIB::ifDescr.521 = STRING: ge-7/0/0.0
IF-MIB::ifDescr.522 = STRING: ge-7/0/2
IF-MIB::ifDescr.523 = STRING: ge-7/0/3
IF-MIB::ifDescr.524 = STRING: ge-7/0/4
IF-MIB::ifDescr.525 = STRING: ge-7/0/5
IF-MIB::ifDescr.526 = STRING: ge-7/0/1.0
IF-MIB::ifDescr.527 = STRING: ge-7/0/6
IF-MIB::ifDescr.528 = STRING: ge-7/0/7
IF-MIB::ifDescr.529 = STRING: t1-6/0/0
IF-MIB::ifDescr.530 = STRING: t1-6/0/1
IF-MIB::ifDescr.531 = STRING: fab0
IF-MIB::ifDescr.532 = STRING: fab0.0
```

ソフトウェアアップグレード

各ノードに個別に接続して、"request system software add"コマンドを実行することで、Junos OSをアップグレードできます。(FTPまたはSSHが有効であるという前提で)FTPまたはSCPを使用してイメージをプライマリノードにコピーできます。イメージがプライマリノードにコピーされた時点で、"file copy"コマンドを使用して、ファイルをセカンダリノードにコピーできます。以下の手順では、in-bandマネジメントされているクラスタの両方のノードをアップグレードする方法について詳しく説明しています(この例はJシリーズのクラスタで実行していますが、手順およびコマンドは、ブランチ向けSRXシリーズとJシリーズの両方のクラスタで同じです)。

1.任意の方法で(この例では、/var/tmp内のファイルを使用)、ソフトウェアイメージをプライマリノードにコピーします。

2.file copyコマンドを使用して、ファイルをプライマリノードからバックアップノードにコピーします(処理には数分程度かかります。この例では、イメージを/var/tmp directory in node0にコピーしています)。

```
labuser@J2320-1# run file copy /var/tmp/junos-jsr-10.2R1.3-domestic.tgz node1:/
var/tmp
```

3.バックアップノードにログインして、新しいイメージをロードします。

```
labuser@J2320-1# run request routing-engine login node 1
--- JUNOS 10.1-20100515.0 built 2010-05-15 06:07:46 UTC
{secondary:node1}
labuser@J2320-2> request system software add /var/tmp/junos-jsr-10.2R1.3-
domestic.tgz no-copy unlink
NOTICE:Validating configuration against junos-jsr-10.2R1.3-domestic.tgz.
NOTICE:Use the 'no-validate' option to skip this if desired.
Checking compatibility with configuration
Initializing...
Verified manifest signed by PackageProduction_10_1_0
Verified junos-10.1-20100515.0-domestic signed by PackageProduction_10_1_0
Using /var/tmp/junos-jsr-10.2R1.3-domestic.tgz
Checking junos requirements on /
Saving boot file package in /var/sw/pkg/junos-boot-jsr-10.2R1.3.tgz
Verified manifest signed by PackageProduction_10_2_0
Hardware Database regeneration succeeded
Validating against /config/juniper.conf.gz
cp:/cf/var/validate/chroot/var/etc/resolv.conf and /etc/resolv.conf are identical
(not copied).
cp:/cf/var/validate/chroot/var/etc/hosts and /etc/hosts are identical (not
copied).
Network security daemon: warning:You have enabled/disabled inet6 flow.
Network security daemon:You must reboot the system for your change to take effect.
Network security daemon:If you have deployed a cluster, be sure to reboot all
nodes.
mgd: commit complete
Validation succeeded
Validating against /config/rescue.conf.gz
Network security daemon: warning:You have enabled/disabled inet6 flow.
Network security daemon:You must reboot the system for your change to take effect.
Network security daemon:If you have deployed a cluster, be sure to reboot all
nodes.
```

```
mgd: commit complete
Validation succeeded
Installing package '/var/tmp/junos-jsr-10.2R1.3-domestic.tgz' ...
Verified junos-boot-jsr-10.2R1.3.tgz signed by PackageProduction_10_2_0
Verified junos-jsr-10.2R1.3-domestic signed by PackageProduction_10_2_0
Available space:333778 require:4160
Saving boot file package in /var/sw/pkg/junos-boot-jsr-10.2R1.3.tgz
JUNOS 10.2R1.3 will become active at next reboot
WARNING:A reboot is required to load this software correctly
WARNING:     Use the 'request system reboot' command
WARNING:         when software installation is complete
Saving state for rollback ...
Removing /var/tmp/junos-jsr-10.2R1.3-domestic.tgz
{secondary:node1}
labuser@J2320-2> exit
```

4.プライマリノードをアップグレードします。

```
labuser@J2320-1# run request system software add /var/tmp/junos-jsr-10.2R1.3-
domestic.tgz no-copy unlink
NOTICE:Validating configuration against junos-jsr-10.2R1.3-domestic.tgz.
NOTICE:Use the 'no-validate' option to skip this if desired.
Checking compatibility with configuration
Initializing...
Verified manifest signed by PackageProduction_10_1_0
Verified junos-10.1-20100515.0-domestic signed by PackageProduction_10_1_0
Using /var/tmp/junos-jsr-10.2R1.3-domestic.tgz
Checking junos requirements on /
Saving boot file package in /var/sw/pkg/junos-boot-jsr-10.2R1.3.tgz
Verified manifest signed by PackageProduction_10_2_0
Hardware Database regeneration succeeded
Validating against /config/juniper.conf.gz
cp:/cf/var/validate/chroot/var/etc/resolv.conf and /etc/resolv.conf are identical
(not copied).
cp:/cf/var/validate/chroot/var/etc/hosts and /etc/hosts are identical (not
copied).
Network security daemon: warning:You have enabled/disabled inet6 flow.
Network security daemon:You must reboot the system for your change to take effect.
Network security daemon:If you have deployed a cluster, be sure to reboot all
nodes.
mgd: commit complete
Validation succeeded
Validating against /config/rescue.conf.gz
mgd: commit complete
Validation succeeded
Installing package '/var/tmp/junos-jsr-10.2R1.3-domestic.tgz' ...
```

```
Verified junos-boot-jsr-10.2R1.3.tgz signed by PackageProduction_10_2_0
Verified junos-jsr-10.2R1.3-domestic signed by PackageProduction_10_2_0
Available space:332709 require:4160
Saving boot file package in /var/sw/pkg/junos-boot-jsr-10.2R1.3.tgz
JUNOS 10.2R1.3 will become active at next reboot
WARNING:A reboot is required to load this software correctly
WARNING:     Use the 'request system reboot' command
WARNING:         when software installation is complete
Saving state for rollback ...
Removing /var/tmp/junos-jsr-10.2R1.3-domestic.tgz

{primary:node0}[edit]
labuser@J2320-1#
```

5.両方のノードを再起動します。

```
labuser@J2320-1# run request routing-engine login node 1
--- JUNOS 10.1-20100515.0 built 2010-05-15 06:07:46 UTC
{secondary:node1}
labuser@J2320-2> request system reboot
Reboot the system ?[yes,no] (no) yes

Shutdown NOW!
[pid 6456]

{secondary:node1}
labuser@J2320-2>
*** FINAL System shutdown message from labuser@J2320-2 ***
System going down IMMEDIATELY


{secondary:node1}
labuser@J2320-2> exit

rlogin: connection closed

{primary:node0}[edit]
labuser@J2320-1# run request system reboot
Reboot the system ?[yes,no] (no) yes

Shutdown NOW!
[pid 7048]

{primary:node0}[edit]
labuser@J2320-1#
*** FINAL System shutdown message from labuser@J2320-1 ***
System going down IMMEDIATELY
```

両方のノードを再起動すると、クラスタは新しいイメージで再起動します。

まとめ

ブランチ向けSRXシリーズ サービス・ゲートウェイおよびJシリーズのシャーシクラスタはシンプルな実装機能であり、企業本社と拠点・支社を結ぶ、信頼性に優れたエンタープライズ接続を実現します。2台のジュニパーネットワークスのセキュリティデバイス間でステートフルトラフィックフェイルオーバーを可能にするとともに、単一のデバイスの抽象化を維持することで、ネットワーク設計を簡素化します。非対称トラフィック、VPN、LAN/WAN混在環境など、さまざまな一般的な接続の課題を解決するため、この機能は設計上、十分に配慮されています。ブランチ向けジュニパーネットワークスSRXシリーズ サービス・ゲートウェイおよびJシリーズ サービスルーターはシャーシクラスタを導入することで、ハイパフォーマンスと信頼性が求められるネットワークの導入事例に最適な基盤を提供します。

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、ネットワークイノベーション企業です。デバイスからデータセンター、消費者からクラウド事業者にいたるまで、ジュニパーネットワークスは、ネットワークの利便性と経済性を変え、ビジネスを変革するソフトウェア、シリコン、システムを提供しています。ジュニパーネットワークスに関する詳細な情報は、以下をご覧ください。

http://www.juniper.net/jp/ 、**Twitter** 、**Facebook**

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

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

米国本社
Juniper Networks, Inc.

1194 North Mathilda Ave
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

3500132-003 JP Apr 2014