Cisco NCS 560 シリーズ ルータ（Cisco IOS XR リリース 7.0.x）MPLS コンフィギュレーション ガイド

初版：2019 年 8 月 30 日

シスコシステムズ合同会社
〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー
http://www.cisco.com/jp
お問い合わせ先：シスコ コンタクトセンター
0120-092-255 （フリーコール、携帯・PHS含む）
電話受付時間：平日 10:00～12:00、13:00～17:00
http://www.cisco.com/jp/go/contactcenter/

目 次

第 1 章 **MPLS Label Distribution Protocol の実装** 1

MPLS Label Distribution Protocol の実装に関する前提条件 1

MPLS LDP の制約事項 2

ラベル配布プロトコルの概要 2

Label Distribution Protocol の設定 2

Label Distribution Protocol の設定 3

Label Distribution Protocol 検出パラメータの設定 3

Label Distribution Protocol の targeted hellos の検出 4

ラベル アドバタイズメント コントロール 4

ローカル ラベル割り当てコントロールの設定 5

ダウンストリーム オン デマンドの設定 6

明示的ヌル ラベルの設定 6

Label Distribution Protocol の自動設定 7

セッション保護の設定 8

Label Distribution Protocol 内部ゲートウェイ プロトコル（IGP）同期の設定 8

Label Distribution Protocol のグレースフル リスタートの設定 9

Label Distribution Protocol ノンストップ ルーティングの設定 10

MPLS Label Distribution Protocol：詳細 11

ラベル スイッチド パスのセットアップ 11

Label Distribution Protocol のグレースフルリスタートの詳細 13

セッション保護の詳細 16

第 2 章 **MPLS の静的なラベル付け** 19

MPLS の制約事項 19

ラベル範囲の定義および MPLS カプセル化のイネーブル化 20
ラベル不一致の特定と解消 21
再帰的プレフィックスによるスタティック LSP ネクスト ホップ解決の設定 22
BVI 経由の MPLS スタティックの設定 23

第 3 章 MPLS トラフィック エンジニアリングの実装 25
MPLS-TE 機能の概要 26
MPLS-TE の動作の仕組み 27
MPLS-TE の設定 28
MPLS-TE トポロジの作成 28
MPLS-TE トンネルの作成 29
Fast Reroute の設定 30
自動トンネル バックアップの設定 32
ネクスト ホップ バックアップ トンネルの設定 33
SRLG ノード保護の設定 33
先行標準 DS-TE の設定 34
RDM を使用した IETF DS-TE トンネルの設定 35
MAM を使用した IETF DS-TE トンネルの設定 36
柔軟な名前ベースのトンネル制約の設定 37
自動帯域幅の設定 37
自動トンネル メッシュの設定 39
MPLS トラフィック エンジニアリング エリア間トンネリングの設定 40
ポリシーベース トンネル選択の設定 41
LDP over MPLS-TE の設定 42
MPLS-TE パス保護の設定 44
MPLS-TE 機能の詳細 46
ポリシーベース トンネル選択 50

第 4 章 MPLS-TE への RSVP の実装 53
RSVP を使用した MPLS LSP の設定 53
MPLS-TE 用 RSVP の機能の概要 54

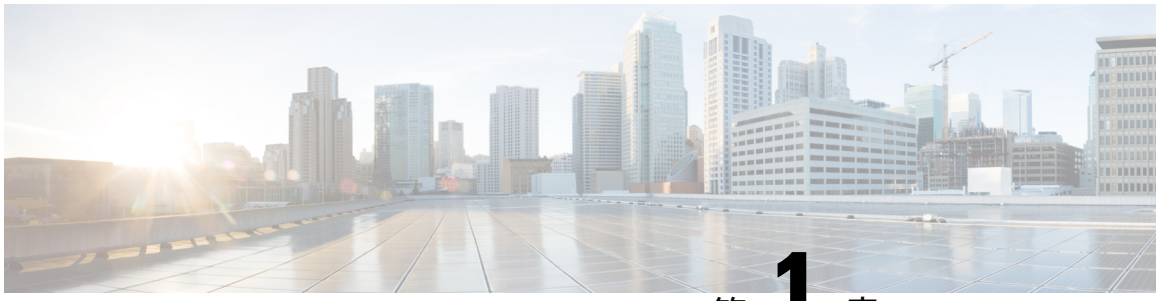
MPLS-TE 用 RSVP の設定 54
RSVP メッセージ認証のグローバル設定 55
インターフェイスでの RSVP 認証の設定 56
ネイバーでの RSVP 認証の設定 57
グレースフル リスタートの設定 57
リフレッシュ削減の設定 59
ACL ベース プレフィックス フィルタリングの設定 59
RSVP パケット ドロップの設定 60
RSVP トラップの有効化 61
MPLS-TE 用 RSVP の機能の詳細 61

第 5 章 MPLS OAM の実装 65
MPLS LSP ping 65
MPLS LSP traceroute 67

第 6 章 グローバル重み付け SRLG 保護の設定 69

第 7 章 自動帯域幅バンドル TE++ の設定 73

第 8 章 Point to Multipoint Traffic Engineering（ポイントツーマルチポイント トラフィック エンジニアリング）の設定 77



第 1 章

MPLS Label Distribution Protocol の実装

MPLS（マルチ プロトコル ラベル スイッチング）は、ラベル スイッチングに基づいた転送メカニズムです。MPLS ネットワークでは、データ パケットにラベルが割り当てられ、ラベルの内容に基づいてパケット転送の決定が行われます。ラベル付きパケットを MPLS ネットワーク上で切り替えるために、さまざまな送信元と宛先のペアに対して所定のパスが確立されます。これらの所定のパスは、ラベル スイッチド パス（LSP）と呼ばれます。LSP を確立するために、MPLS シグナリング プロトコルが使用されます。Label Distribution Protocol（LDP）は、LSP を確立するために使用される MPLS シグナリング プロトコルです。このモジュールでは、MPLS LDP の設定方法について説明します。

- MPLS Label Distribution Protocol の実装に関する前提条件 （1 ページ）
- MPLS LDP の制約事項 （2 ページ）
- ラベル配布プロトコルの概要 （2 ページ）
- Label Distribution Protocol の設定 （2 ページ）
- MPLS Label Distribution Protocol：詳細 （11 ページ）

MPLS Label Distribution Protocol の実装に関する前提条件

次に、MPLS LDP を実装するための前提条件を示します。

- 適切なタスク ID を含むタスク グループに関連付けられているユーザ グループに属している必要があります。このコマンド リファレンスには、各コマンドに必要なタスク ID が含まれます。ユーザ グループの割り当てが原因でコマンドを使用できないと考えられる場合、AAA 管理者に連絡してください。
- Cisco IOS XR ソフトウェアを実行している必要があります。
- 複合ミニイメージおよび MPLS パッケージをインストールする必要があります。

（注） この点は、Cisco NCS 540 シリーズ ルータには適用されません。

- IGP をアクティブにする必要があります。

- セッションダウンがネイバーで隣接のダウンよりも前に発生するように、ネイバーなど、セッション保持時間の短い帯域幅を使用することを推奨します。次に、hello タイムのデフォルト値を示します。
  - 保持時間は 15 秒です。
  - 間隔は 5 秒です。

  たとえば、**holdtime** コマンドを使用して LDP セッション保持時間を 30 秒に設定できます。

MPLS LDP の制約事項

- LDP 統計情報は、**show mpls forwarding command** の出力に表示されません。

ラベル配布プロトコルの概要

IP 転送では、パケットがルータに到達すると、そのルータは IP ヘッダー内の宛先アドレスを確認し、ルート検索を実行してパケットをネクストホップに転送します。MPLS は、パケットがラベルに基づいて転送される転送メカニズムです。Label Distribution Protocol は、MPLS 環境でラベルを割り当て、配布、およびインストールします。これは、ネットワークとレイヤ間のルーティング情報をデータリンク レイヤのスイッチド パスに直接マッピングすることで、ラベル スイッチドルータ（LSR）がネットワークを通じて LSP を確立する一連の手順とメッセージです。これらの LSP には直接接続されたネイバーにエンドポイントを持たせたり（IP のホップバイホップ転送に対応）、ネットワーク出力ノードにエンドポイントを設定し、すべての中間ノードを介してスイッチングを可能にしたりできます。

LSP は、RSVP トラフィック エンジニアリング（TE）または LDP によりスタティックに作成できます。LDP により作成される LSP は、エンドツーエンド パスではなく、ホップバイホップ セットアップを実行します。LDP を使用すると、LSR で潜在的ピア ルータを検出し、これらのピアとの LDP セッションを確立して、ラベル バインディング情報を交換できます。ラベル バインディングを学習すると、LDP は MPLS フォワーディング プレーンを設定できるようになります。

LSP の設定方法については、「MPLS Label Distribution Protocol：詳細 （11 ページ）」を参照してください。

Label Distribution Protocol の設定

要件に応じて、LDP では、次のトピックで説明するいくつかの基本設定作業が必要です。

Label Distribution Protocol の設定

この項では、基本的な LDP 設定について説明します。ルータを LDP ピアとなる可能性があるルータに接続しているすべてのインターフェイス上で LDP をイネーブルにする必要があります。mpls ldp コンフィギュレーション モードでインターフェイスを指定することで、インターフェイス上で LDP を有効にできます。

設定例

次の例では、インターフェイス経由で LDP をイネーブルにする方法を示します。

```
RP/0/RP0/CPU0:Router(config)#  mpls ldp
RP/0/RP0/CPU0:Router(config-ldp)#  router-id 192.168.70.1
RP/0/RP0/CPU0:Router(config-ldp)#  interface HundredGigE 0/9/0/0
RP/0/RP0/CPU0:Router(config-ldp-if)#  commit
```

Label Distribution Protocol 検出パラメータの設定

LDP を実行している LSR は、すべての LDP 対応インターフェイスで hello メッセージを送信して、互いを検出します。したがって、インターフェイス上で LDP hello メッセージを受信する LSR は、そのインターフェイス上の LDP ルータの存在を認識しています。LDP hello メッセージがインターフェイス上で送受信される場合、LDP を実行している 2 つの LSR 間のリンクには LDP 隣接関係があります。デフォルトでは、hello メッセージは 5 秒ごとに送信され、15 秒の保留時間があります。保留時間が切れる前に LSR がピアから検出 hello を受信しない場合、LSR は検出された LDP ネイバー リストからピア LSR を削除します。LDP 検出パラメータは、デフォルト パラメータを変更するように設定できます。

直接接続されていない LSR 間の LDP セッションは、ターゲット LDP セッションと呼ばれます。ターゲット LDP セッションの場合、LDP はターゲット hello メッセージを使用して拡張ネイバーを検出します。デフォルトでは、ターゲット hello メッセージは 10 秒ごとに送信され、90 秒の保留時間があります。

設定例

次の例に、以下の LDP 検出パラメータを設定する方法を示します。

- hello hold time
- hello interval
- targeted hello hold time
- targeted hello interval

```
RP/0/RP0/CPU0:Router(config)# mpls ldp
RP/0/RP0/CPU0:Router(config-ldp)# router-id 192.168.70.1
RP/0/RP0/CPU0:Router(config-ldp)# discovery hello holdtime 30
RP/0/RP0/CPU0:Router(config-ldp)# discovery hello interval 10
RP/0/RP0/CPU0:Router(config-ldp)# discovery targeted-hello holdtime 120
RP/0/RP0/CPU0:Router(config-ldp)# discovery targeted-hello interval 15
RP/0/RP0/CPU0:Router(config-ldp)# commit
```

確認

このセクションでは、MPLS LDP 検出パラメータの設定を確認します。

```
RP/0/RP0/CPU0:Router# show mpls ldp parameters
LDP Parameters:
Role: Active
Protocol Version: 1
Router ID: 192.168.70.1
Discovery:
Link Hellos:     Holdtime:30 sec, Interval:10 sec
Targeted Hellos: Holdtime:120 sec, Interval:15 sec
Quick-start: Enabled (by default)
Transport address:       IPv4: 192.168.70.1
```

Label Distribution Protocol の targeted hellos の検出

直接接続されていない LSR 間の LDP セッションは、ターゲット LDP セッションと呼ばれます。直接接続されていない LDP ネイバーの場合は、両方のルータで LDP ネイバーシップを手動で設定する必要があります。

設定例

次に、直接接続されていないルータ Router1 および Router 2 に LDP を設定する例を示します。

```
RP/0/RP0/CPU0:Router1(config)#  mpls ldp
RP/0/RP0/CPU0:Router1(config-ldp)# router-id 192.168.70.1
RP/0/RP0/CPU0:Router1(config-ldp)# neighbor 172.20.10.10 targeted
RP/0/RP0/CPU0:Router1(config-ldp)# interface HundredGigE 0/9/0/0
RP/0/RP0/CPU0:Router1(config-ldp-if)# commit
```

```
RP/0/RP0/CPU0:Router2(config)#  mpls ldp
RP/0/RP0/CPU0:Router2(config-ldp)# router-id 172.20.10.10
RP/0/RP0/CPU0:Router2(config-ldp)# neighbor 192.168.70.1 targeted
RP/0/RP0/CPU0:Router2(config-ldp)# address-family ipv4
RP/0/RP0/CPU0:Router2(config-ldp-af)#discoverey targeted-hello accept
RP/0/RP0/CPU0:Router2(config-ldp-af)# commit
```

ラベル アドバタイズメント コントロール

LDP では、ラベルのアドバタイジングや受信を制御できます。ラベル アドバタイズメント コントロール（アウトバウンド フィルタリング）またはラベル受け入れコントロール（インバウンド フィルタリング）を使用して、ラベル バインディング情報の交換を制御できます。

ラベル アドバタイズメント コントロール（アウトバウンド フィルタリング）

Label Distribution Protocol はすべてのネイバーのすべてのプレフィックスのラベルをアドバタイズします。（拡張性やセキュリティが理由で）これが望ましくない場合、1 つ以上のピアに対する 1 つ以上のプレフィックスでローカル ラベル アドバタイズメントのアウトバウンド フィルタリングを実行するように LDP を設定できます。この機能は、LDP アウトバウンド ラベル フィルタリングまたはローカル ラベル アドバタイズメント コントロールと呼ばれています。

mpls ldp label advertise コマンドを使用すると、ラベル バインディング情報の交換を制御できます。オプション キーワードを使用すると、選択プレフィックスをすべてのネイバーにアドバタイズ、選択プレフィックスを定義済みネイバーにアドバタイズ、またはすべてのプレフィックスのすべてのピアへのラベル アドバタイズメントをディセーブルにできます。選択してアドバタイズされるプレフィックスおよびピアは、アクセス リストで定義されます。

設定例：ラベル アドバタイズメント コントロール

次に、アウトバウンド ラベル アドバタイズメント コントロールを設定する例を示します。この例では、ネイバーはラベル アドバタイズメントをアドバタイズし受信するように指定されています。また、ラベル アドバタイズメントのためのインターフェイスも指定されています。

```
RP/0/RP0/CPU0:Router(config)# mpls ldp
RP/0/RP0/CPU0:Router(config-ldp)# address-family ipv4
RP/0/RP0/CPU0:Router(config-ldp-af)# label local advertise to 1.1.1.1:0 for pfx_ac11
RP/0/RP0/CPU0:Router(config-ldp-af)# label local advertise interface TenGigE 0/0/0/5

RP/0/RP0/CPU0:Router(config-ldp-af)# commit
```

ラベル受け入れコントロール（インバウンド フィルタリング）

LDP は、すべてのピアからのすべてのプレフィックスのラベルを（リモート バインディングとして）受け入れます。LDPは、リベラル ラベル保持モードで機能します。これは、LDPに、特定のプレフィックスのすべてのピアからのリモートバインディングを保持するように指示します。セキュリティ上の理由から、またはメモリを節約するため、特定のピアからのプレフィックスのセットのラベル バインディング受け入れを設定することで、この動作を上書きできます。プレフィックスの定義セットのリモート バインディングをフィルタリングする機能は、LDPインバウンドラベル フィルタリングまたはラベル受け入れコントロールとも呼ばれます。

設定例：ラベル受け入れコントロール（インバウンド フィルタリング）

次に、ラベル受け入れコントロールを設定する例を示します。この例では、LSRは、アクセス リストで定義されたプレフィックスについてネイバーからのラベル バインディングを受け入れて保持するように設定されています。

```
RP/0/RP0/CPU0:Router(config)#mpls ldp
RP/0/RP0/CPU0:Router(config-ldp)#address-family ipv4
RP/0/RP0/CPU0:Router(config-ldp-af)#label remote accept from 192.168.1.1:0 for acl_1
RP/0/RP0/CPU0:Router(config-ldp-af)#label remote accept from 192.168.2.2:0 for acl_2
RP/0/RP0/CPU0:Router(config-ldp-af)#commit
```

ローカル ラベル割り当てコントロールの設定

LDP は、すべての IGP プレフィックスのラベル バインディングを作成し、そのすべてのピアからすべての IGP プレフィックスのラベル バインディングを受信します。LSR が多数の IGP プレフィックス用に複数のピアからラベル バインディングを受信する場合、かなりのメモリと CPU が消費されます。いくつかのシナリオでは、LDP ラベル バインディングの大半はアプリケーションにとって有用ではない場合があり、場合によってはローカル ラベルの割り当てを制

限する必要があります。これは、アクセス リストを使用して、ローカル ラベルの割り当てをプレフィックスのセットに制限できる場合、LDP ローカル ラベル割り当てコントロールを使用して実行されます。ローカル ラベル割り当てを制限すると、メモリ使用要件の軽減、ローカル フォワーディングやネットワークおよびピアのアップデートの軽減など、いくつかのメリットがあります。

設定例

次の例に、IP アクセス リストを使用してローカル ラベル割り当てを設定し、ローカル ラベルが割り当ておよびアドバタイズできるプレフィックスのセットを指定する方法を示します。

```
RP/0/RP0/CPU0:Router(config)# mpls ldp
RP/0/RP0/CPU0:Router(config-ldp)# address-family ipv4
RP/0/RP0/CPU0:Router(config-ldp-af)# label local allocate for pfx_acl_1
RP/0/RP0/CPU0:Router(config-ldp-af)# commit
```

ダウンストリーム オン デマンドの設定

デフォルトでは、LDP は、すべてのルートのラベル アドバタイズメントがすべての LDP ピアから受信されるダウンストリーム未承諾モードを使用します。ダウンストリーム オン デマンド機能は、ダウンストリーム オンデマンド モードのサポートを強化します。このモードでは、ピアが明示的に要求しない限り、ラベルはそのピアにアドバタイズされません。同時に、ピアは自動的にラベルをアドバタイズしないため、ネクスト ホップが、リモート ラベルが割り当てられていないピアを示す場合、ラベル要求が必ず送信されます。

ダウンストリーム オンデマンド設定では、ACL を使用して、ダウンストリーム オンデマンド モードにピアのセットを指定します。ダウンストリーム オンデマンドを有効にするには、セッションの両方のピアで設定する必要があります。セッションの一方のピアだけでダウンストリーム オンデマンド機能が設定されている場合、そのセッションでは、ダウンストリーム オンデマンド モードを使用できません。

設定例

次に、LDP ダウンストリーム オン デマンドを設定する例を示します。

```
RP/0/RP0/CPU0:Router(config)# mpls ldp
RP/0/RP0/CPU0:Router(config-ldp)# session downstream-on-demand with ACL1
RP/0/RP0/CPU0:Router(config-ldp)# commit
```

明示的ヌル ラベルの設定

Cisco MPLS LDP は、暗黙的または明示的なヌル ラベルを、特定の LSR で終端するルートまたはプレフィックスのローカル ラベルとして使用します。これらのルートには、ローカルで接続またはアタッチされたすべてのネットワークが含まれます。デフォルトでは、ヌル ラベルは、LDP コントロール プレーンによる Penultimate Hop Popping（PHOP）メカニズムの実装を可能にする **implicit-null** です。これが望ましくない場合、LDP コントロール プレーンによる Ultimate Hop Popping（UHOP）メカニズムの実装を可能にする **explicit-null** ラベルを設定できます。明

示的ヌル機能は、最終ホップ LSR で設定できます。アクセスリストを使用して、PHP を必要とする IP プレフィックスを指定することができます。

デフォルトで非ヌル ラベルを割り当てる必要がある場合でも、**implicit-null-override** コマンドを使用することで、特定のプレフィックスに暗黙的ヌル ローカル ラベルを適用できます。たとえば、デフォルトでは、LSRは、IGPルートの非ヌルラベルを割り当て、アドバタイズします。LSR の最後から2番目のホップでこのルートのLSPを終端する場合、**implicit-null-override** コマンドを使用して、このプレフィックスに暗黙的ヌルラベルの割り当ておよびアドバタイズメントを適用できます。

設定例：明示的ヌル

次に、明示的ヌル ラベルを設定する例を示します。

```
RP/0/RP0/CPU0:Router(config)# mpls ldp
RP/0/RP0/CPU0:Router(config-ldp)# address-family ipv4
RP/0/RP0/CPU0:Router(config-ldp-af)# label local advertise explict-null
RP/0/RP0/CPU0:Router(config-ldp-af)# commit
```

設定例：暗黙的ヌルのオーバーライド

次に、プレフィックスのセットに対して暗黙的なヌルのオーバーライドを設定する例を示します。

```
RP/0/RP0/CPU0:Router(config)# mpls ldp
RP/0/RP0/CPU0:Router(config-ldp)# address-family ipv4
RP/0/RP0/CPU0:Router(config-ldp-af)# label local advertise implicit-null-override for
acl-1
RP/0/RP0/CPU0:Router(config-ldp-af)# commit
```

Label Distribution Protocol の自動設定

LDP 自動設定では、IGPプロトコルが有効になっているすべてのインターフェイスで自動的にLDPを設定できます。通常、LDPは、IGPルートのラベルを割り当て、アドバタイズします。LDP は、IGP によりすべてのアクティブ インターフェイスでイネーブルにする必要があります。LDPの手動構成時には、LDP下でインターフェイスのセットを定義する必要があります。この作業には時間がかかります。LDP 自動設定により、LDP 下で同じインターフェイスのリストを指定する必要がなくなり、設定作業が簡単になります。

設定例：OSPF に LDP 自動設定を有効にする

次に、指定した OSPF インスタンスに対して LDP 自動設定を有効にする例を示します。

```
RP/0/RP0/CPU0:Router(config)# router ospf 190
RP/0/RP0/CPU0:Router(config-ospf)# mpls ldp auto-config
RP/0/RP0/CPU0:Router(config-ospf)# area 8
RP/0/RP0/CPU0:Router(config-ospf-ar)# interface HundredGigE 0/9/0/0
RP/0/RP0/CPU0:Router(config-ospf-ar-if)# commit
```

セッション保護の設定

新しいリンクまたはノードがリンク障害後に起動すると、IP は、MPLS LDP よりも前の段階で速く収束し、MPLS 収束までに MPLS トラフィックが損失する可能性があります。リンクがフラップすると、リンク ディスカバリの損失のために LDP セッションもフラップします。LDP セッション保護により、トラフィックの損失が最小限に抑えられ、収束が迅速化され、既存の LDP（リンク）セッションが保護されます。ピアに対してセッション保護が有効になっている場合、LDP は基本的な検出リンク hello に加えて、ターゲット hello（転送検出）の送信を開始します。ダイレクトリンクがダウンすると、ターゲット hello は、代替パスが存在していれば、そのパスを介してピア LSR に引き続き転送されます。したがって、LDP セッションは、リンクがダウンした後でも維持されます。

LDP セッション保護を設定して、すべてのピアまたはピアの特定のセット（peer-acl で指定）でセッションを自動的に保護することができます。LDP は、設定されると、プライマリ リンク隣接がすでに存在するネイバーのバックアップ targeted hello を自動的に開始します。これらのバックアップ targeted hello は、プライマリ リンク隣接がダウンしても、LDP セッションを保持します。

設定例

次の例では、アクセス コントロール リスト peer-acl-1 で指定されたピアの LDP セッション保護を、最大期間 60 秒に設定する方法を示します。

```
RP/0/RP0/CPU0:Router(config)# mpls ldp
RP/0/RP0/CPU0:Router(config-ldp)# session protection for peer-acl-1 duration 60
RP/0/RP0/CPU0:Router(config-ldp)# commit
```

Label Distribution Protocol 内部ゲートウェイ プロトコル（IGP）同期の設定

LDP と内部ゲートウェイ プロトコル（IGP）間の同期が失われると、MPLS トラフィックが失われます。たとえば、リンク アップ時、IGP は LDP 収束が発生する前にリンクをアドバタイズして使用できます。または、LDP セッションがダウンした後でも IGP でリンクを使用し続けることができます。

LDP IGP の同期化では、MPLS LDP がそのリンクで収束される場合にのみ、IGP が通常のメトリックでリンクをアドバタイズできるように LDP と IGP が調整されます。LDP では、LDP が適切なラベル バインディングを送信し、ピアから少なくとも 1 つのラベル バインディングを受信するリンクで、少なくとも 1 つの LDP セッションがアップで実行中の場合だけリンクが収束されると見なされます。LDP は、リンク アップまたはセッション ダウン イベント時にこの情報を IGP に通信し、IGP は、同期ステートに応じて機能します。

LDP-IGP 同期は、OSPF および ISIS プロトコルの両方でサポートされ、対応する IGP プロトコル コンフィギュレーション モードで設定されます。状況によっては、設定可能なインターバルで、再同期化の宣言を遅延する必要があります。LDP は、同期化の宣言を最大 60 秒遅延できる設定オプションを提供します。LDP は、リンク アップまたはセッション ダウン イベント時にこの情報を IGP に通信します。

LDP IGP 同期の設定：Open Shortest Path First（OSPF）の例

次に、OSPF インスタンスに LDP-IGP 同期を設定する例を示します。同期遅延は 30 秒に設定されています。

```
RP/0/RP0/CPU0:Router(config)# router ospf 100
RP/0/RP0/CPU0:Router(config-ospf)# mpls ldp sync
RP/0/RP0/CPU0:Router(config-ospf)# mpls ldp igp sync delay 30
RP/0/RP0/CPU0:Router(config-ospf)# commit
```

LDP IGP 同期の設定：Intermediate System to Intermediate System（IS-IS）

次に、IS-IS に LDP-IGP 同期を設定する例を示します。

```
RP/0/RP0/CPU0:Router(config)# router isis 100
RP/0/RP0/CPU0:Router(config-isis)#  interface HundredGigE 0/9/0/0
RP/0/RP0/CPU0:Router(config-isis-if)# address-family ipv4 unicast
RP/0/RP0/CPU0:Router(config-isis-if-af)# mpls ldp sync
RP/0/RP0/CPU0:Router(config-isis-if-af)# commit
```

Label Distribution Protocol のグレースフル リスタートの設定

LDP グレースフル リスタートでは、LDP セッションが停止したときに LDP ピアが MPLS フォワーディング ステートを保持するメカニズムが提供されます。LDP グレースフル リスタートを使用しない場合、確立されたセッションで障害が発生すると、対応するフォワーディング ステートが、リスタートおよびピア ノードからすぐに消去されます。この場合、LDP フォワーディングは、最初から再起動する必要があるので、データおよび接続が失われる可能性があります。LDP グレースフル リスタートが設定されている場合、LDP セッションが再起動しても、トラフィックは中断することなく転送され続けます。LDP グレースフル リスタート機能は、セッション初期化中に 2 つのピア間でネゴシエーションされます。セッションの初期化中に、ルータはグレースフル リスタートの Typed Length Value（TLV）を送信することにより、LDP グレースフル リスタートを実行する機能をアドバタイズします。この TLV には、再接続時間と回復時間が含まれています。再接続時間と回復時間の値は、ルータでサポートされているグレースフル リスタート機能を示します。再接続時間は、再起動ルータが接続を確立するまでピア ルータが待機する時間です。ルータは、隣接ルータが再起動していることを検出すると、再接続を試みる前に回復時間の終了まで待機します。回復時間は、隣接ルータが再起動ルータに関する情報を維持する時間です。

設定例

次に、LDP グレースフル リスタートを設定する例を示します。この例では、隣接ルータがグレースフル リスタートするルータについてフォワーディング ステートを維持する時間を 180 に設定します。再接続時間は 169 秒に設定されています。

```
RP/0/RP0/CPU0:Router(config)# mpls ldp
RP/0/RP0/CPU0:Router(config-ldp)#  interface HundredGigE 0/9/0/0
RP/0/RP0/CPU0:Router(config-ldp-if)# exit
RP/0/RP0/CPU0:Router(config-ldp)# graceful-restart
RP/0/RP0/CPU0:Router(config-ldp)# graceful-restart forwarding-state-holdtime 180
```

```
RP/0/RP0/CPU0:Router(config-ldp)# graceful-restart reconnect-timeout 169
RP/0/RP0/CPU0:Router(config-ldp)# commit
```

Label Distribution Protocol ノンストップ ルーティングの設定

LDP ノンストップ ルーティング（NSR）機能は、ルート プロセッサ（RP）または分散型ルート プロセッサ（DRP）のフェールオーバーなどの障害をルーティング ピアに見えないようにして、収束パフォーマンスへの負荷を最小限に抑えたり、回避したりします。デフォルトでは、NSR は、AToM 以外、すべての LDP セッションでグローバルにイネーブルにされています。

サービスの中断では、次のイベントが発生している場合があります。

- ルート プロセッサ（RP）または分散ルート プロセッサ（DRP）フェールオーバー
- LDP プロセスの再開
- Minimum Disruption Restart（MDR）

（注） グレースフル リスタート機能とは異なり、LDP NSR では、プロトコル拡張機能は必要なく、ネットワークの他のルータでのソフトウェア アップグレードの必要もありません。また、LDP NSR によりピア ルータで NSR をサポートする必要もありません。L2VPN 設定は、NSR ではサポートされていません。アクティブ LDP のプロセス障害によりセッションが損失します。その結果、RP スイッチオーバーがリカバリ アクションとして設定されるまで、NSR は提供できません。

設定例

次に、LDP ノンストップ ルーティングを設定する例を示します。

```
RP/0/RP0/CPU0:Router(config)# mpls ldp
RP/0/RP0/CPU0:Router(config-ldp)# nsr
RP/0/RP0/CPU0:Router(config-ldp)# commit
```

確認

```
RP/0/RP0/CPU0:Router# show mpls ldp nsr summary
Mon Dec  7 04:02:16.259 UTC
Sessions:
Total: 1, NSR-eligible: 1, Sync-ed: 0
(1 Ready)
```

MPLS Label Distribution Protocol：詳細

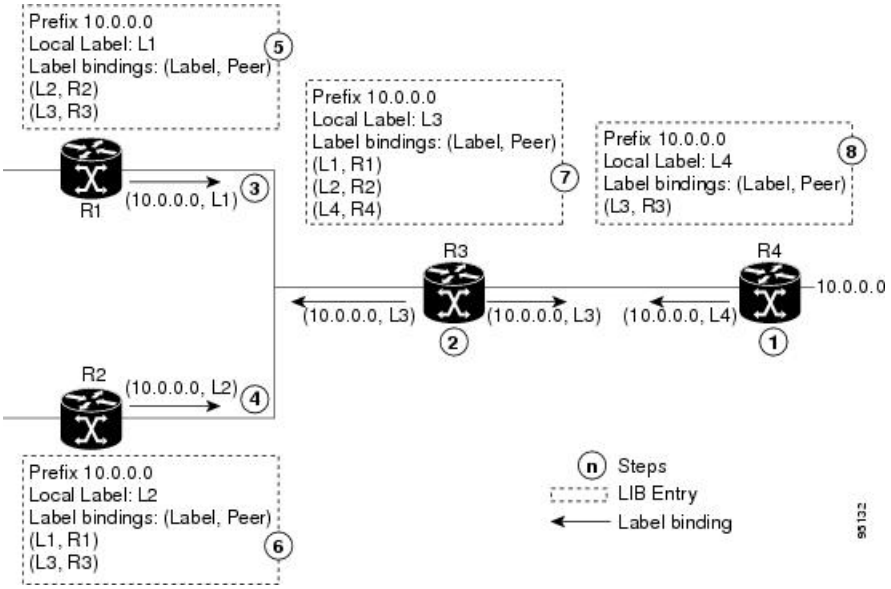
この項では、LSP の設定、LDP のグレースフル リスタート、および LDP セッション保護に関する詳細な概念情報について説明します。

ラベル スイッチド パスのセットアップ

MPLS パケットは、ラベル スイッチド パス（LSP）を使用して MPLS ネットワーク上のノード間で転送されます。LSP は、静的に、または LDP のような Label Distribution Protocol を使用して作成できます。LDP により作成されたラベル スイッチド パスはエンドツーエンド パスではなく、ホップバイホップ パスのセットアップを実行します。LDP により、ラベル スイッチ ルータ（LSR）は、潜在的なピア ルータを検出し、これらのピアとの LDP セッションを確立して、ラベル バインディング情報を交換します。

次の図は、LSP セットアップのためのラベル バインディングの交換プロセスを示します。

図 *1:* ラベル スイッチド パスのセットアップ



ネットワーク（10.0.0.0）では、ホップバイホップ LSP が各隣接ルータ（またはノード）間でセットアップされます。各ノードは、ローカル ラベルを割り当て、これをそのネイバーにバインディングとして渡します。

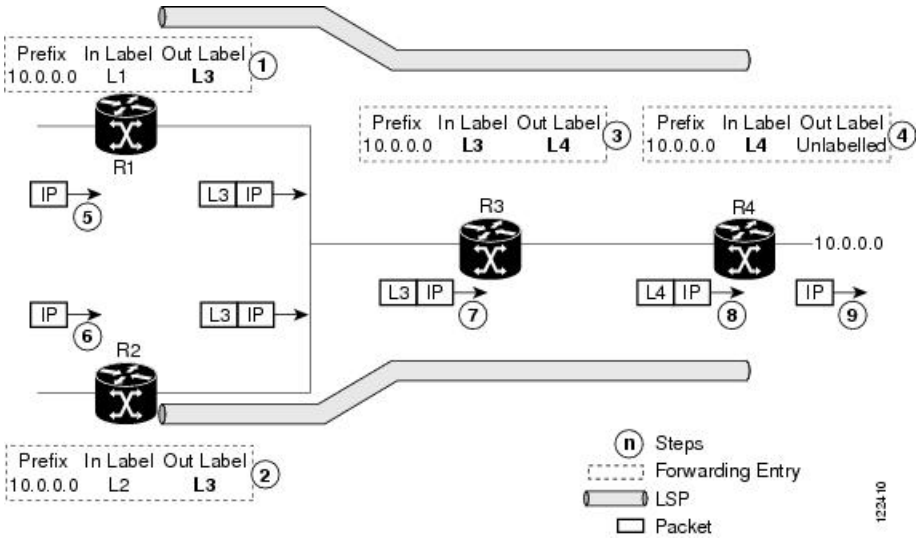
1. R4 は、ローカル ラベル L4 をプレフィックス 10.0.0.0 に割り当て、これをそのネイバー（R3）にアドバタイズします。
2. R3 は、ローカル ラベル L3 をプレフィックス 10.0.0.0 に割り当て、これをそのネイバー（R1、R2、R4）にアドバタイズします。
3. R1 は、ローカル ラベル L1 をプレフィックス 10.0.0.0 に割り当て、これをそのネイバー（R2、R3）にアドバタイズします。

4. R2 は、ローカル ラベル L2 をプレフィックス 10.0.0.0 に割り当て、これをそのネイバー（R1、R3）にアドバタイズします。
5. R1 のラベル情報ベース（LIB）は、ネイバーからのローカルおよびリモート ラベル バインディングを保持します。
6. R2 の LIB は、ネイバーからのローカルおよびリモート ラベル バインディングを保持します。
7. R3 の LIB は、ネイバーからのローカルおよびリモート ラベル バインディングを保持します。
8. R4 の LIB は、ネイバーからのローカルおよびリモート ラベル バインディングを保持します。

MPLS 転送

ラベルバインディングが学習されると、MPLS フォワーディングプレーンがセットアップされ、パケットは次の図に示すように転送されます。

図 *2 : MPLS* 転送



1. R3 は、FIB で通知されるように、10.0.0.0 のネクストホップなので、R1 は、ラベルバインディングを R3 から選択して、フォワーディングエントリ（レイヤ 1、レイヤ 3）をインストールします。
2. R3 は、10.0.0.0 のネクストホップなので（FIB で通知）、R2 は、ラベルバインディングを R3 から選択して、フォワーディングエントリ（レイヤ 2、レイヤ 3）をインストールします。
3. R4 は、10.0.0.0 のネクストホップなので（FIB で通知）、R3 は、ラベルバインディングを R4 から選択して、フォワーディングエントリ（レイヤ 3、レイヤ 4）をインストールします。

4. 10.0.0.0 のネクストホップは R4 外なので（FIB で通知）、R4 は、NO-LABEL をアウトバウンドとして使用して、フォワーディングエントリ（レイヤ 4）をインストールします。アウトバウンドパケットは IP のみで転送されます。
5. 入力 LSR R1 の着信 IP トラフィックは、ラベルインポーズされ、ラベル L3 の MPLS パケットとして転送されます。
6. 入力 LSR R2 の着信 IP トラフィックは、ラベルインポーズされ、ラベル L3 の MPLS パケットとして転送されます。
7. R3 は、ラベル L3 の MPLS パケットを受信し、MPLS ラベル フォワーディング テーブルで検索して、このパケットをラベル L4 の MPLS パケットとしてスイッチします。
8. R4 は、ラベル L4 の MPLS パケットを受信し、MPLS ラベル フォワーディング テーブルで検索して、ラベルを削除する必要があると判断します。次に、トップ ラベルをポップして、これを IP フォワーディング プレーンに渡します。
9. IP フォワーディングは、パケットを継承して、転送します。

Label Distribution Protocol のグレースフルリスタートの詳細

LDP（Label Distribution Protocol）グレースフルリスタートは、コントロール プレーン メカニズムを提供して、ハイアベイラビリティを保証し、ノンストップ フォワーディング（NSF）サービス中に障害を検出しリカバリできるようにします。グレースフルリスタートは、フォワーディングに影響を与えずに、シグナリングおよびコントロールプレーンの障害から回復する方法です。

LDP グレースフルリスタートを使用しない場合、確立されたセッションで障害が発生すると、対応するフォワーディングステートが、リスタートおよびピアノードからすぐに消去されます。この場合、LDP フォワーディングは、最初から再起動するので、データおよび接続が失われる可能性があります。

LDP グレースフルリスタート機能は、セッション初期化中に FT SESSION TLV で 2 つのピア間でネゴシエーションされます。この Typed Length Value（TLV）では、各ピアは、次の情報をピアにアドバタイズします。

再接続時間

この LSR がコントロールチャネル障害後に再接続するまで他のピアが待機する最大時間をアドバタイズします。

回復時間

他のピアがこの LSR を復元またはリフレッシュする最大時間をアドバタイズします。この時間は、先行のセッション障害後のセッション再確立中のみに使用されます。

FT フラグ

再起動により、このフラグの保存（ローカル）ノードのステートを復元できるかどうかを指定します。

グレースフル リスタート セッション パラメータが伝達され、セッションが起動し動作していると、グレースフルリスタート手順がアクティブになります。

マルチリンク、または同じネイバーの targeted LDP hello 隣接、あるいはこれら両方のネットワークで LDP グレースフル リスタート プロセスを設定する場合、ネイバー コントロール プレーン障害時に任意の hello 隣接がタイムアウトになる前に、グレースフルリスタートがセッションでアクティブになっていることを確認します。これをアクティブにするには、たとえば、セッションタイムアウトが hello 隣接タイムアウトの前に発生するように、ネイバー間のセッション保持時間を低く設定します。LDPセッション保持時間は、次の式を使用して設定することを推奨します。

```
Session Holdtime <= (Hello holdtime - Hello interval) * 3
```

たとえば、リンク hello の保持時間およびインターバルがそれぞれデフォルト値の 15 秒および 5 秒である場合、セッション保持時間は、30 秒以下に設定します。

グレースフルリスタートのフレーズ

グレースフル リスタート メカニズムは、次のフェーズに分かれます。

制御通信障害の検出

システムが次のいずれかの状況を検出したときに、制御通信障害が検出されます。

- LDP hello ディスカバリ メッセージの欠落
- LDP キープアライブ プロトコル メッセージの欠落
- ピアとの Transmission Control Protocol（TCP）切断の検出

障害時のフォワーディング ステート メンテナンス

各 LSR での永続的フォワーディングステートは、LDP コントロールプレーンにより、永続的ストレージ（チェックポイント）を介してアーカイブされます。コントロールプレーンのリカバリ中、フォワーディングプレーンは、フォワーディングステートを保持しますが、ステイルマークを付けます。同様に、ピアコントロールプレーンも（ステイルマークを付けて）再起動中のノードに関連付けられているインストール済みフォワーディングを保持します。ローカル ノード フォワーディングとリモート ノード フォワーディングのプレーンステートを組み合わせることで、NSF を保証し、トラフィックの損失を防ぎます。

制御ステートのリカバリ

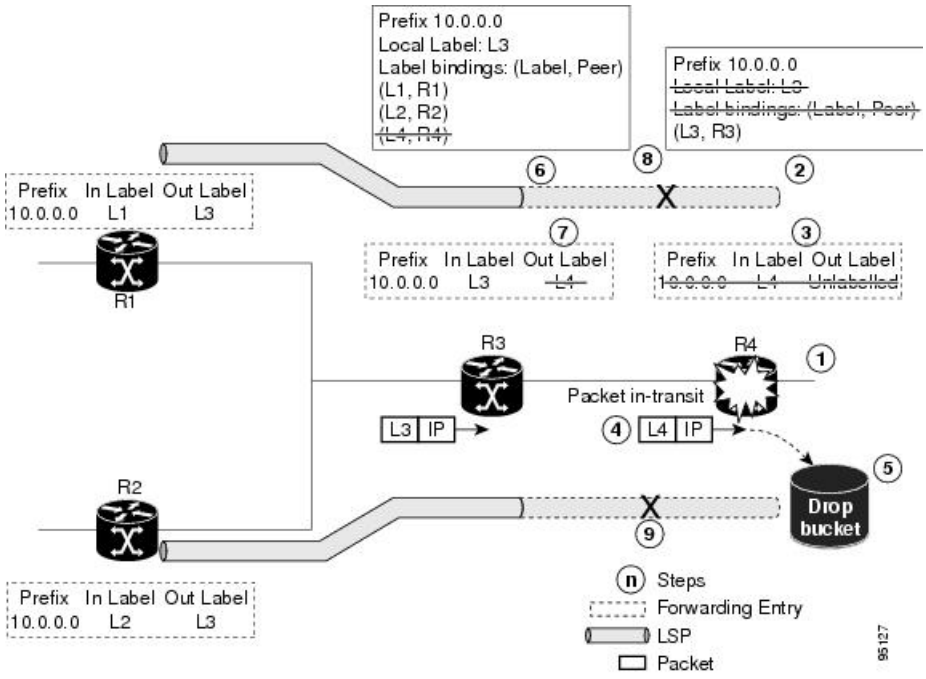
リカバリは、セッションが再確立され、ラベルバインディングが再び交換されるときに発生します。このプロセスにより、ピアノードは、ステイル フォワーディング ステートを同期化およびリフレッシュできます。

コントロールプレーンの障害

コントロールプレーン障害は、接続に影響します。ルータ コントロール プレーンによりインストールされたフォワーディングステートが失われ、転送中パケットがドロップされ、NSF が損失する可能性があります。次の図に、コントロールプレーン障害とグレースフルリスタート

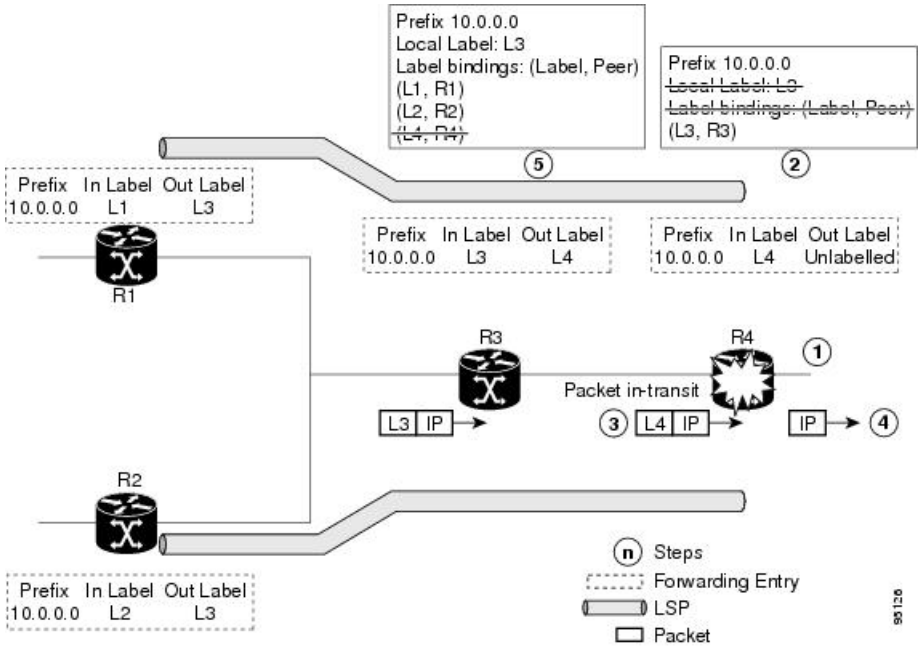
によるリカバリを示し、接続の損失につながるコントロールプレーン障害の処理と結果、およびグレースフルリスタートを使用したリカバリについて説明します。

図 *3:* コントロールプレーンの障害



グレースフルリスタートによるリカバリ

図 *4:* グレースフルリスタートでのリカバリ



1. R4 LSR コントロールプレーンが再起動します。
2. コントロールプレーンが再起動すると、LIB が失われます。
3. R4 LDP コントロールプレーンによりインストールされたフォワーディングステートがすぐに削除されます。
4. R3 から R4（ラベルはまだ L4）への転送中の任意のパケットが R4 に到着します。
5. R4 の MPLS フォワーディングプレーンが、ローカルラベル L4 でルックアップを実行しますが失敗します。これにより、パケットがドロップされ、NSF が満たされなくなります。
6. R3 LDP ピアが、コントロール プレーン チャネルの障害を検出して、そのラベルバインディングを R4 から削除します。
7. R3 コントロールプレーンは、R4 からの出ラベルの使用を停止し、対応するフォワーディング ステート（リライト）を削除します。これにより、フォワーディングが失敗します。
8. R4 に接続されている確立済み LSP は、R3 で終端し、R1 から R4 へのエンドツーエンド LSP が終了します。
9. R4 に接続されている確立済み LSP は、R3 で終端し、R2 から R4 へのエンドツーエンド LSP が終了します。

LDP コントロールプレーンがリカバリすると、リスタート LSR は、そのフォワーディングステートの保持タイマーを開始し、フォワーディングステートをチェックポイントデータから復元します。これにより、フォワーディングステートおよびエントリが復元され、オールドマークが付けられます。

リスタート LSR は、正常に復元されたかどうかに関係なく、FT セッション TLV に示されているピアに再接続します。ステートが復元できた場合、バインディングは再び同期化されます。

リスタートピアが接続し、ネイバーリカバリタイマーを開始すると、ピア LSR は、（リスタート LSR により開始された）ネイバー再接続タイマーを停止します。ピア LSR は、リスタートピアがそのステートを正常に復元できた場合、FT セッション TLV をチェックします。次に、対応するフォワーディング ステート エントリを復元し、リスタートピアからバインディングを受信します。リカバリタイマーが失効すると、任意のフォワーディングステート（この段階ではステイルマークが付いています）が削除されます。

リスタート LSR が復元（再起動）に失敗した場合、リスタート LSR フォワーディングステートおよびエントリは、タイムアウトになり削除されます。ネイバー関連のフォワーディングステートまたはエントリは、再接続またはリカバリタイマーが失効すると、ピア LSR により削除されます。

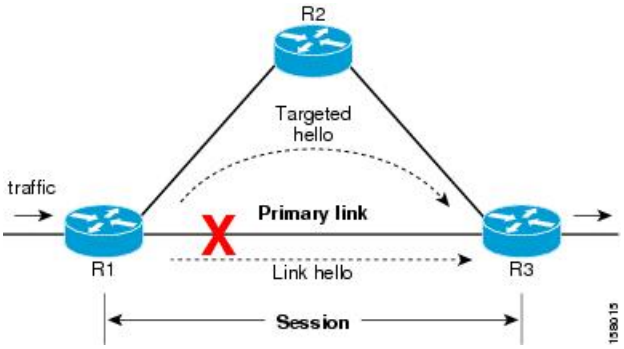
セッション保護の詳細

LDP セッション保護により、すべてのピアまたはピアの特定のセット（peer-acl で指定）でセッションを自動的に保護するように LDP を設定できます。LDP は、設定されると、プライマリリンク隣接がすでに存在するネイバーのバックアップ targeted hello を自動的に開始します。こ

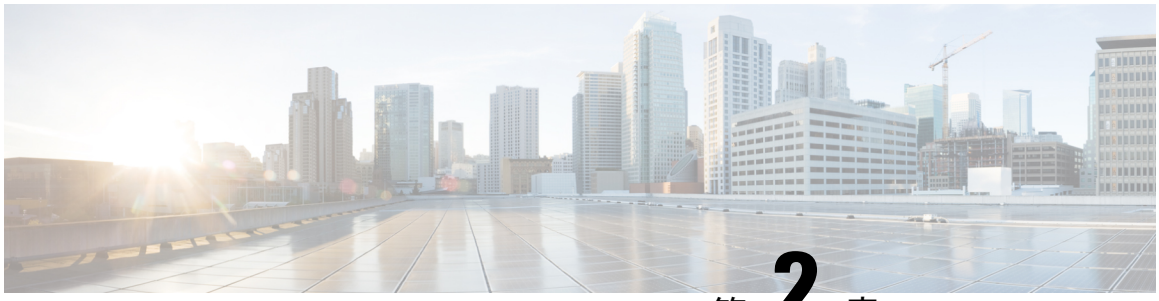
れらのバックアップ targeted hello は、プライマリリンク隣接がダウンしても、LDP セッションを保持します。

セッション保護の図は、ネイバー R1 と R3 の間の LDP セッション保護を示します。R1 および R3 間でのプライマリリンク隣接は、リンクとバックアップが直接接続されます。ターゲット隣接は、R1 と R3 間で保守されます。ダイレクトリンクが失敗すると、LDP リンク隣接が破棄されますが、セッションは、targeted hello 隣接を使用してアップのまま実行します（R2 を介します）。ダイレクトリンクが再びアップになっても、LDP セッションステートは変わらず、LDP は、すばやく収束し、MPLS トラフィックの転送を開始します。

図 *5:* セッション保護



（注） LDP セッション保護が（リンク障害時に）アクティブの場合、保護は無制限で保守されます。



第 2 章

MPLS の静的なラベル付け

MPLS の静的機能では、IPv4 プレフィックスにローカルラベルを静的に割り当てることができます。また、静的ラベルを含むパケットの転送に必要なネクストホップ情報を指定することで、ラベルスイッチドパス（LSP）をこれらの静的ラベル用にプロビジョニングできます。

静的に割り当てられたラベルと動的に割り当てられたラベルに不一致がある場合は、ルータによって警告メッセージがコンソールログに発行されます。この警告メッセージによって、不一致を特定して解決できます。

動的ラベル上での静的ラベルをアドバタイズすると、次の利点が得られます。

- ピアから不要なラベルを受信するリスク（侵害された MPLS 動的ラベリングプロトコルの実行）が軽減され、セキュリティが向上します。
- 定義済みの LSP 上でユーザが完全に制御できるようになります。
- 動的なラベル付けが処理されないため、システムリソースの使用が最適化されます。

制約事項

- IPv6 パケットでの静的なラベル付けはサポートされていません。
- ルータは静的ラベルの設定時にラベルの不一致を回避しません。生成されたすべての不一致は後で解消する必要があります。
- 等コスト マルチパス ルーティング（ECMP）はサポートされていません。
- インターフェイスを明示的に設定して、静的 MPLS ラベルを持つトラフィックを処理する必要があります。

- MPLS の制約事項 （19 ページ）
- ラベル範囲の定義および MPLS カプセル化のイネーブル化 （20 ページ）
- ラベル不一致の特定と解消 （21 ページ）
- 再帰的プレフィックスによるスタティック LSP ネクスト ホップ解決の設定 （22 ページ）
- BVI 経由の MPLS スタティックの設定 （23 ページ）

MPLS の制約事項

- MPLS 統計はサポートされていません。

ラベル範囲の定義および MPLS カプセル化のイネーブル化

デフォルトでは、MPLS カプセル化はすべてのインターフェイス上でディセーブルになっています。MPLS カプセル化は、静的 MPLS ラベル付きのトラフィックが通過するすべての入力 MPLS インターフェイスおよび出力 MPLS インターフェイス上で明示的にイネーブルになっている必要があります。

また、動的ラベルの範囲を定義する必要もあります。この動的範囲外にあるすべてのラベルが、静的ラベルとして手動による割り当てに使用できます。ルータは、静的に設定されたラベルを指定したラベルの範囲と照合して確認しません。したがって、ラベルの不一致を防止するには、静的 MPLS ラベルを動的ラベルの範囲に入らないように設定することが必要です。

設定例

MPLS の静的なラベル付けの設定を実行するには、以下を実行する必要があります。値は例として示しています。

1. 動的ラベル範囲を定義します。このタスクでは 17000 ~ 18000 で設定されます。
2. 必要なインターフェイス上で MPLS カプセル化をイネーブルにします。
3. 特定の入力ラベル 24035 に対して静的な MPLS LSP をセットアップします。
4. ラベル 24035 で受信したパケットの場合に、MPLS プロトコルがラベルをスワップし、24036 のラベルを適用するように転送情報を指定します。新しいラベルを適用した後、指定されたインターフェイスを通じてネクスト ホップである 10.2.2.2 にパケットを転送します。

```
RP/0/RP0/cpu 0: router(config)#mpls label range table 0 17000 18000
RP/0/RP0/cpu 0: router(config)#commit

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

RP/0/RP0/cpu 0: router(config-mpls-static)# interface HundredGigE 0/9/0/0
RP/0/RP0/cpu 0: router(config-mpls-static)#address-family ipv4 unicast
RP/0/RP0/cpu 0: router(config-mpls-static-af)#local-label 24035 allocate

RP/0/RP0/cpu 0: router(config-mpls-static-af-lbl)#forward
RP/0/RP0/cpu 0: router(config-mpls-static-af-lbl-fwd)#

RP/0/RP0/cpu 0: router(config-mpls-static-af-lbl-fwd)# commit
```

確認

MPLS がイネーブルになっているインターフェイスを確認します。

```
RP/0/RP0/cpu 0: router# show  mpls interfaces
Mon May 12 06:21:30.937 DST
Interface                  LDP      Tunnel   Static   Enabled
-------------------------- -------- -------- -------- --------
```

```
TenGigE0/0/0/5             No       No       Yes      Yes
```

指定したラベル値のステータスが「Created」であることを確認します。

```
RP/0/RP0/cpu 0: router#show mpls static local-label all
Tue Apr 22 18:21:55.764 UTC
Label   VRF             Type         Prefix           RW Configured   Status
------- --------------- ------------ ---------------- --------------- --------
24035   default         X-Connect    NA               Yes             Created
```

動的範囲をチェックし、指定したローカル ラベル値がこの範囲にないことを確認します。

```
RP/0/RP0/cpu 0: router#show mpls label range
Mon Apr 28 19:56:00.596 IST
Range for dynamic labels: Min/Max: 17000/18000
```

MPLS の動的設定が適用され、ラベル転送が行われていることを確認します。

```
RP/0/RP0/cpu 0: router#show mpls lsd forwarding
Wed Nov 25 21:40:57.918 UTC
In_Label, (ID), Path_Info: <Type>
24035, (Static), 1 Paths
   1/1: IPv4, 'default':4U, BE1.2, nh=10.20.3.1, lbl=35001, flags=0x0, ext_flags=0x0
```

関連コマンド

- mpls static
- mpls label range
- show mpls interfaces

ラベル不一致の特定と解消

静的ラベルまたはラベルの範囲の設定時または設定解除時は、次の場合にラベルの不一致が発生する可能性があります。

- 動的ラベルとすでにバインディングされている IP プレフィックスに静的ラベルを設定した。
- 同じラベル値が別の IP プレフィックスに動的に割り当てられている場合に、その IP プレフィックスに静的ラベルを設定した。

確認

次の show コマンドを使用してラベルの不一致を特定します。

```
Router#show mpls static local-label discrepancy
Tue Apr 22 18:36:31.614 UTC
Label   VRF             Type         Prefix           RW Configured   Status
------- --------------- ------------ ---------------- --------------- --------
24000   default         X-Connect     NA              Yes             Discrepancy

Router#show mpls static local-label all
Tue Apr 22 18:36:31.614 UTC
```

```
Label   VRF             Type         Prefix           RW Configured   Status
------- --------------- ------------ ---------------- --------------- --------
24000   default         X-Connect    N/A              Yes             Discrepancy
24035   default         X-Connect    N/A              Yes             Created
```

```
RP/0/RP0/cpu 0: router#show log
Thu Apr 24 14:18:57.655 UTC
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Console logging: level warnings, 199 messages logged
    Monitor logging: level debugging, 0 messages logged
    Trap logging: level informational, 0 messages logged
    Buffer logging: level debugging, 2 messages logged

Log Buffer (307200 bytes):

RP/0/RSP0/CPU0:Apr 24 14:18:53.743 : mpls_static[1043]:
%ROUTING-MPLS_STATIC-7-ERR_STATIC_LABEL_DISCREPANCY :
The system detected 1 label discrepancies (static label could not be allocated due to
conflict with other applications).
Please use 'clear mpls static local-label discrepancy' to fix this issue.
RP/0/RSP0/CPU0:Apr 24 14:18:53.937 : config[65762]: %MGBL-CONFIG-6-DB_COMMIT :
Configuration committed by user 'cisco'.
Use 'show configuration commit changes 1000000020' to view the changes.
```

訂正

ラベルの不一致は、動的ラベルが割り当てられている IP プレフィックスに新しいラベルを割り当てることによって解消されます。不一致解消中は、静的ラベル設定が優先されます。不一致解消中は、トラフィックが影響を受けます。

```
Router# clear mpls static local-label discrepancy all
```

不一致が解消されたことを確認します。

```
Router# show mpls static local-label all
Wed Nov 25 21:45:50.368 UTC
Label   VRF             Type         Prefix           RW Configured   Status
------- --------------- ------------ ---------------- --------------- --------
24000   default         X-Connect    N/A              Yes             Created
24035   default         X-Connect    N/A              Yes             Created
```

関連コマンド

- show mpls static local-label discrepancy
- clear mpls static local-label discrepancy all

再帰的プレフィックスによるスタティック LSP ネクスト ホップ解決の設定

ルーティング テーブル エントリが直接接続された出口インターフェイスではなく別の IP アドレスを参照する場合、ネクストホップ IP アドレスは出口インターフェイスを持つ別のルートを使用して解決されます。ネクストホップ IP アドレスを解決するには複数のルックアップが

必要であるため、これは再帰ルックアップと呼ばれます。再帰的プレフィックスによるスタティック LSP ネクスト ホップ解決機能は、スタティック LSP の再帰的ルートの解決をサポートしています。この機能では、スタティック LSP に **resolve-nexthop** コマンドを使用して直接接続されていないネクスト ホップを指定できます。

制約事項

この機能には、次の制限があります。

- eBGP ルートのみがサポートされています。

設定例

次に、再帰的プレフィックスによるスタティック LSP ネクスト ホップ解決を設定する例を示します。ここで、192.168.2.1 は eBGP を通じて学習された再帰的ルートです。

```
RP/0/0/CPU0:Router# configure terminal
RP/0/0/CPU0:Router(config)# mpls static
RP/0/0/CPU0:Router(config-mpls-static)# lsp anycast_5001
RP/0/0/CPU0:Router(config-mpls-static-lsp)# in-label 5001 allocate
RP/0/0/CPU0:Router(config-mpls-static-lsp)# forward
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# path 1 resolve-nexthop 192.168.2.1
out-label pop
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# exit
```

確認

次に、再帰的プレフィックスによるスタティック LSP ネクスト ホップ解決の設定を検証する例を示します。

```
RP/0/0/CPU0:Router# show  mpls static lsp anycast_5001 detail
Tue Sep 12 20:00:09.248 UTC
LSP Name             Label   VRF             AFI  Type         Prefix           RW
Configured   Status
-------------------- ------- --------------- ---- ------------ ----------------
--------------- --------
anycast_5001         5001    default         N/A  X-Connect    N/A              Yes
         Created
  PRIMARY SET:
    [resolve-mode: nexthop 192.168.2.1]
    Path 0 : nexthop BVI1 1.1.1.3, out-label Pop, Role: primary, Path-id: 0, Status:
valid
    Path 1 : nexthop BVI1 1.1.1.4, out-label Pop, Role: primary, Path-id: 0, Status:
valid
    Path 2 : nexthop BVI1 1.1.1.5, out-label Pop, Role: primary, Path-id: 0, Status:
valid
    Path 3 : nexthop BVI1 1.1.1.6, out-label Pop, Role: primary, Path-id: 0, Status:
valid
```

BVI 経由の MPLS スタティックの設定

ブリッジグループ仮想インターフェイス（BVI）は、ブリッジされる一連のインターフェイスを表すルーテッドインターフェイスです。BVIを使用すると、複数のインターフェイスを共通のブロードキャスト ドメインのメンバーとして変換できます。BVI 経由の MPLS スタティッ

ク機能では、スタティック LSP を設定する際に、ネクストホップとして BVI インターフェイスを指定できます。

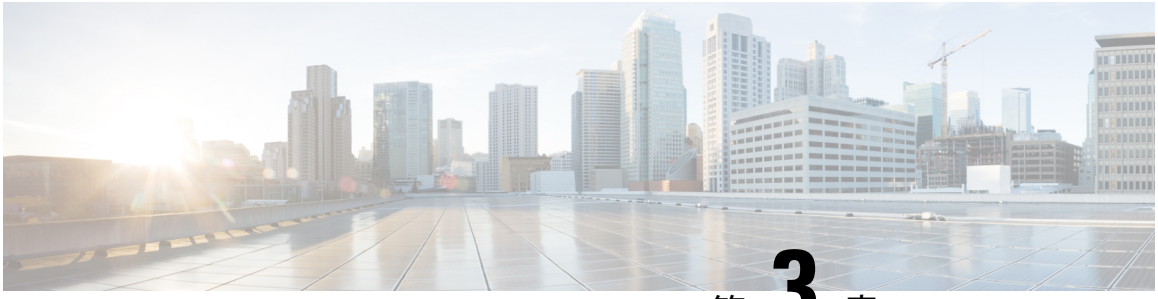
制約事項

- BVI の直接接続されたサブネット プレフィックスへの解決ネクストホップによる MPLS スタティックはサポートされていません。
- BVI 経由のバックアップ パスはサポートされていません。
- Fast Reroute（FRR）はサポートされていません。

設定例

次の例では、スタティック LSP のネクスト ホップとして BVI インターフェイスを設定する方法を示します。

```
RP/0/0/CPU0:Router# configure terminal
RP/0/0/CPU0:Router(config)# mpls static
RP/0/0/CPU0:Router(config-mpls-static)# interface TenGig 0/0/0/0
RP/0/0/CPU0:Router(config-mpls-static)# lsp bvi
RP/0/0/CPU0:Router(config-mpls-static-lsp)# in-label 5001 allocate
RP/0/0/CPU0:Router(config-mpls-static-lsp)# forward
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# path 1 nexthop BVI1 192.168.2.1 out-label
 pop
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# path 1 nexthop BVI1 192.168.2.1 out-label
 4444
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# exit
```



第 3 章

MPLS トラフィック エンジニアリングの実装

従来の IP ルーティングは、トラフィックをできるだけ速く宛先に転送することを重視しています。その結果、ルーティング プロトコルは、ネットワーク内の各宛先へのメトリックに従って最小コストのルートを見つけ出し、すべてのルータが宛先 IP アドレスに基づいてパケットを転送し、パケットはホップバイホップで転送されます。したがって、従来の IP ルーティングでは、リンクの利用可能な帯域幅は考慮されません。これにより、一部のリンクが他のリンクと比べて過剰に使用され、帯域幅が効率的に利用されなくなる可能性があります。トラフィック エンジニアリング（TE）は、ネットワーク リソースへのトラフィック ストリームの非効率的なマッピングによって問題が生じる場合に使用されます。トラフィック エンジニアリングでは、データ パケットが追随するパスを制御し、自動的に計算された宛先ベースの最短パスでは不可能であろう、輻輳したリンクから輻輳していないリンクにトラフィック フローを移動させることができます。

ラベル スイッチング機能を備えたマルチプロトコル ラベル スイッチング（MPLS）により、IP ルートの検索の必要がなくなり、仮想回線（VC）スイッチング機能が提供されます。これにより企業は、フレーム リレーや非同期転送モード（ATM）などの従来のネットワークの場合と同様のパフォーマンスを IP ベースのネットワーク サービスで実現できます。MPLS トラフィック エンジニアリング（MPLS-TE）は MPLS バックボーンに依存し、レイヤ 2 ATM およびフレーム リレー ネットワークの TE 機能を複製および拡張します。

MPLS TE はトポロジとネットワーク内で使用可能なリソースを学習し、帯域幅などのリソース要件とネットワーク リソースに基づいてトラフィック フローを特定のパスにマッピングします。MPLS TE では、ラベル スイッチド パス（LSP）の形で送信元から宛先への単方向トンネルが構築され、その後トラフィックの転送で使用されます。トンネルが開始される場所はトンネルのヘッドエンドまたはトンネルの送信元と呼ばれ、トンネルが終了するノードはトンネルのテールエンドまたはトンネルの宛先と呼ばれます。トンネルが通過するルータをトンネルの中点と呼びます。

MPLS は、Intermediate System-to-Intermediate System（IS-IS）や Open Shortest Path First（OSPF）などのリンクステート ベースの内部ゲートウェイ プロトコル（IGP）の拡張機能を使用します。MPLS は、必要なリソースと利用可能なリソースに基づいて、LSP ヘッドで TE トンネルを計算します（制約ベースのルーティング）。設定されている場合、IGP はこれらの LSP にトラフィックを自動的にルーティングします。通常、MPLS-TE バックボーンを通過するパケッ

トは、入力ポイントと出力ポイントを接続する単一の LSP 上を伝送されます。MPLS TE では、Resource Reservation Protocol（RSVP）を使用して、MPLS ネットワーク上で LSP を自動的に確立および維持します。

（注） ラベル付きのパスによって保護されるラベルなしのパスの組み合わせはサポートされていません。

- MPLS-TE 機能の概要 （26 ページ）
- MPLS-TE の動作の仕組み （27 ページ）
- MPLS-TE の設定 （28 ページ）
- MPLS-TE 機能の詳細 （46 ページ）

MPLS-TE 機能の概要

MPLS トラフィック エンジニアリングでは、IGP 拡張によって TE 情報がネットワーク全体にフラッディングされます。IGP がリンク属性と帯域幅情報を配信すると、ヘッドエンドルータは MPLS-TE トンネルの先頭から末尾までのベスト パスを計算します。このパスは明示的に設定することもできます。パスが計算されると、RSVP-TE を使用して TE LSP（ラベル付きスイッチ パス）が設定されます。

トラフィックを転送するには、自動ルート、転送隣接関係、またはスタティック ルーティングを設定します。自動ルート機能は、テールエンドルータによって割り当てられたルートとそのダウンストリーム ルートをヘッドエンド ルータのルーティング テーブルに通知し、トンネルはトンネルに直接接続されたリンクと見なされます。

転送隣接関係が有効になっている場合、MPLS-TE トンネルは IGP ネットワーク内にリンクとしてアドバタイズされ、リンクのコストが関連付けられます。TE ドメインの外側にあるルータは、TE トンネルを参照し、その TE トンネルを使用して、ネットワーク全体でトラフィックをルーティングするための最短パスを計算します。

MPLS-TE は、障害時のパケット損失を最小限に抑えるために、Fast Reroute と呼ばれる保護メカニズムを提供します。Fast Reroute 用に、バックアップ トンネルを作成する必要があります。自動トンネル バックアップ機能により、ルータは、各バックアップ トンネルを事前に設定するのではなく、バックアップ トンネルを必要なときに動的に構築し、保護されたインターフェイスにバックアップ トンネルを割り当てることができます。

DiffServ 対応トラフィック エンジニアリング（DS-TE）を使用すると、MPLS 対応インターフェイスで複数の帯域幅制約を設定して、さまざまなサービス クラス（CoS）をサポートできます。これらの帯域幅は、その制約を使用してトラフィック クラスの要件に基づいて扱うことができます。

MPLS トラフィック エンジニアリング自動トンネル メッシュ機能を使用すると、最小の MPLS トラフィック エンジニアリング設定で TE トンネルのフル メッシュを自動的に設定できます。MPLS-TE の自動帯域幅機能により、トラフィックの中断なしでトラフィック パターンに基づいて帯域幅を自動的に調整できます。

MPLS-TEエリア間トンネリング機能を使用すると、複数の内部ゲートウェイ プロトコル（IGP）エリアとレベルにまたがる TE トンネルを確立できます。そのため、ヘッドエンドおよびテールエンド ルータが単一のエリアに存在しなければならないという要件がなくなります。

MPLS-TE 機能の詳細については、MPLS-TE 機能の詳細 （46 ページ） を参照してください。

MPLS-TE の動作の仕組み

MPLS-TE では、RSVP を使用して、バックボーン上でラベル スイッチド パス（LSP）を自動的に確立および維持します。LSP で使用されるパスは、LSP リソース要件とネットワーク リソース（帯域幅など）によって決まります。利用可能なリソースは、リンク ステートに基づく内部ゲートウェイ プロトコル（IGP）の拡張機能によってフラッディングされます。MPLS-TE トンネルは、必要なリソースと使用可能なリソースの適合の度合いに基づいてLSPヘッドエンドルータで計算されます（制約ベースのルーティング）。IGPは、これらのLSPにトラフィックを自動的にルーティングします。通常、MPLS-TE バックボーンを通過するパケットは、入力ポイントと出力ポイントを接続する単一の LSP 上を伝送されます。

次の項では、MPLS-TE のコンポーネントについて説明します。

トンネル インターフェイス

レイヤ 2 の観点では、MPLS トンネル インターフェイスは LSP のヘッドエンドを表します。これは、帯域幅要件、メディア要件、プライオリティなどの一連のリソース要件を使用して設定されます。レイヤ3の観点では、LSP トンネル インターフェイスはトンネル宛先への単一方向仮想リンクのヘッドエンドです。

MPLS-TE パス計算モジュール

この計算モジュールはLSPヘッドエンドで動作します。このモジュールは、LSP で使用するパスを決定します。パス計算では、フラッディングされたトポロジおよびリソース情報を含むリンク ステート データベースが使用されます。

TE 拡張機能を備えた RSVP

RSVP は各 LSP ホップで動作し、計算されたパスに基づいて LSP のシグナリングおよび維持のために使用されます。

MPLS-TE リンク管理モジュール

このモジュールは各 LSP ホップで動作し、RSVP シグナリング メッセージに対するリンク コール アドミッションを実行して、フラッディングされるトポロジおよびリソース情報を追跡します。

リンクステート IGP

Intermediate System-to-Intermediate System（IS-IS）または Open Shortest Path First（OSPF）のいずれかを IGP として使用できます。これらの IGP は、リンク管理モジュールからトポロジおよびリソース情報をグローバルにフラッディングするために使用されます。

ラベル スイッチング フォワーディング

この転送メカニズムは、レイヤ 2 と類似の機能をルータに提供し、RSVP シグナリングによって確立された LSP の複数のホップを経由してトラフィックを誘導できるようにします。

MPLS-TE の設定

MPLS-TE では、複数のグローバル ネイバー ルータ間で調整が必要です。RSVP、MPLS-TE、および IGP は、MPLS トラフィック エンジニアリング ネットワークのすべてのルータとインターフェイスで設定されます。明示パスおよび TE トンネル インターフェイスは、ヘッドエンド ルータでのみ設定されます。MPLS-TE には、この項で説明するいくつかの基本的な設定作業が必要です。

MPLS-TE トポロジの作成

MPLS-TE トポロジを構築し、MPLS-TE トンネルを作成するための環境を設定します。この手順には、MPLS-TE を有効にするための基本的なノードおよびインターフェイスの設定が含まれています。制約ベースのルーティングを実行するには、OSPF または IS-IS を IGP 拡張として有効にする必要があります。

はじめる前に

MPLS-TE トポロジの構築を開始する前に、次の前提条件が必要です。

- リンクを正常に行うには、リンクのいずれかの側に安定したルータ ID が必要です。ルータ ID を割り当てない場合、デフォルトでグローバル ルータ ID に設定されます。デフォルトのルータ ID は変更されることがあり、不安的なリンクの原因となる可能性があります。
- ポート インターフェイス上の RSVP を有効にします。

例

次の例では、ノード上で MPLS-TE を有効にしてから、MPLS-TE の一部であるインターフェイスを指定します。ここで、OSPF は情報配信のための IGP 拡張プロトコルとして使用されます。

```
RP/0/RP0/cpu 0: router# configure
RP/0/RP0/cpu 0: router(config)# mpls traffic-eng
RP/0/RP0/cpu 0: router(config-mpls-te)# interface hundredGigE0/9/0/0
RP/0/RP0/cpu 0: router(config)# router ospf area 1
RP/0/RP0/cpu 0: router(config-ospf)# area 0
RP/0/RP0/cpu 0: router(config-ospf-ar)# mpls traffic-eng
RP/0/RP0/cpu 0: router(config-ospf-ar)# interface hundredGigE0/9/0/0
```

```
RP/0/RP0/cpu 0: router(config-ospf-ar-if)# exit
RP/0/RP0/cpu 0: router(config-ospf)# mpls traffic-eng router-id 192.168.70.1
RP/0/RP0/cpu 0: router(config)# commit
```

例

次の例では、ノード上でMPLS-TEを有効にしてから、MPLS-TEの一部であるインターフェイスを指定します。ここで、IS-ISは情報配信のためのIGP拡張プロトコルとして使用されます。

```
# configure
RP/0/RP0/cpu 0: router(config)# mpls traffic-eng
RP/0/RP0/cpu 0: router(config-mpls-te)# interface hundredGigE0/9/0/0
RP/0/RP0/cpu 0: router(config)# router isis 1
RP/0/RP0/cpu 0: router(config-isis)# net 47.0001.0000.0000.0002.00
RP/0/RP0/cpu 0: router(config-isis)# address-family ipv4 unicast
RP/0/RP0/cpu 0: router(config-isis-af)# metric-style wide
RP/0/RP0/cpu 0: router(config-isis-af)# mpls traffic-eng level 1
RP/0/RP0/cpu 0: router(config-isis-af)# exit
RP/0/RP0/cpu 0: router(config-isis)# interface hundredGigE0/9/0/0
RP/0/RP0/cpu 0: router(config-isis-if)# exit
RP/0/RP0/cpu 0: router(config)# commit
```

関連項目

- MPLS-TE の動作の仕組み （27 ページ）
- MPLS-TE トンネルの作成 （29 ページ）

MPLS-TE トンネルの作成

MPLS-TE トンネルの作成は、ご使用のネットワーク トポロジに合うようにトラフィック エンジニアリングをカスタマイズするプロセスです。MPLS-TE トンネルは、ヘッドエンド ルータで作成されます。TE LSP の宛先とパスを指定する必要があります。

トラフィックをトンネルに誘導するには、次の方法を使用できます。

- スタティック ルーティング
- 自動ルート通知
- Forwarding Adjacency（FA）

はじめる前に

次に、MPLS-TE トンネルを作成するための前提条件を示します。

- 隣接ルータにはルータ ID が必要です。
- リンクを正常に行うには、リンクのいずれかの側に安定したルータ ID が必要です。ルータ IDをルータに割り当てない場合、デフォルトでグローバル ルータ IDに設定されます。デフォルトのルータ ID は変更されることがあり、不安的なリンクの原因となる可能性があります。

設定例

次の例では、宛先 IP アドレス 192.168.92.125 を持つヘッドエンド ルータに MPLS-TE トンネルを設定します。トンネルの帯域幅、パスオプション、およびトンネルの転送パラメータも設定されます。スタティック ルーティング、自動ルート通知、または Forwarding Adjacency（FA）を使用して、トラフィックをトンネルに誘導できます。

```
RP/0/RP0/cpu 0: router# configure
RP/0/RP0/cpu 0: router(config)# interface tunnel-te 1
RP/0/RP0/cpu 0: router(config-if)# destination 192.168.92.125
RP/0/RP0/cpu 0: router(config-if)# ipv4 unnumbered Loopback0
RP/0/RP0/cpu 0: router(config-if)# path-option 1 dynamic
RP/0/RP0/cpu 0: router(config-if)# autoroute announce  or forwarding adjacency
RP/0/RP0/cpu 0: router(config-if)# signalled-bandwidth 100
RP/0/RP0/cpu 0: router(config)# commit
```

確認

次のコマンドを使用して、MPLS-TE トンネルの設定を確認します。

```
RP/0/RP0/cpu 0: router# show mpls traffic-engineering tunnels brief

           Signalling Summary:
               LSP Tunnels Process:  running
                      RSVP Process:  running
                        Forwarding:  enabled
           Periodic reoptimization:  every 3600 seconds, next in 2538 seconds
            Periodic FRR Promotion:  every 300 seconds, next in 38 seconds
           Auto-bw enabled tunnels:  0 (disabled)
                       TUNNEL NAME          DESTINATION      STATUS  STATE


                     tunnel-te1       192.168.92.125      up       up
           Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
```

関連項目

- MPLS-TE の動作の仕組み （27 ページ）
- MPLS-TE トポロジの作成 （28 ページ）

Fast Reroute の設定

Fast Reroute（FRR）は、LSP にリンク保護を提供し、リンクで障害が発生した LSP によって送信されたトラフィックを、障害を避けて再ルーティングできるようにします。再ルーティングの決定は、障害の発生したリンクに接続されているルータによって、ローカルに制御されます。トンネルのヘッドエンド ルータには、IGP または RSVP からリンク障害が通知されます。リンク障害を通知されると、ヘッドエンド ルータは、障害をバイパスする新しい LSP を確立しようとします。これによって、障害が発生したリンクを再確立するためのパスが提供され、データ転送が保護されます。バックアップ トンネルのパスには、IP 明示パス、動的に計算されたパス、または準動的パスを使用できます。Fast Reroute の概念的な詳細については、次の項を参照してください。 MPLS-TE 機能の詳細 （46 ページ）

はじめる前に

次に、MPLS-TE トンネルを作成するための前提条件を示します。

- 隣接ルータにはルータ ID が必要です。
- リンクを正常に行うには、リンクのいずれかの側に安定したルータ ID が必要です。ルータ ID をルータに割り当てない場合、デフォルトでグローバル ルータ ID に設定されます。デフォルトのルータ ID は変更されることがあり、不安的なリンクの原因となる可能性があります。

設定例

次の例では、MPLS-TE トンネル上に Fast Reroute を設定します。ここで、tunnel-te 2 はバックアップ トンネルとして設定されています。**protected-by** コマンドを使用すると、別のパスで保護されている明示パスに対してパス保護を設定できます。

```
RP/0/RP0/cpu 0: router # configure
RP/0/RP0/cpu 0: router(config)# interface tunnel-te 1
RP/0/RP0/cpu 0: router(config-if)# fast-reroute
RP/0/RP0/cpu 0: router(config-if)# exit
RP/0/RP0/cpu 0: router(config)# mpls traffic-eng
RP/0/RP0/cpu 0: router(config-mpls-te)# interface HundredGigabitEthernet0/0/1/0
RP/0/RP0/cpu 0: router(config-mpls-te-if)# backup-path tunnel-te 2
RP/0/RP0/cpu 0: router(config)# interface tunnel-te 2
RP/0/RP0/cpu 0: router(config-if)# backup-bw global-pool 5000
RP/0/RP0/cpu 0: router(config-if)# ipv4 unnumbered Loopback0
RP/0/RP0/cpu 0: router(config-if)# destination 192.168.92.125
RP/0/RP0/cpu 0: router(config-if)# path-option l explicit name backup-path protected by
 10
RP/0/RP0/cpu 0: router(config-if)# path-option l0 dynamic
RP/0/RP0/cpu 0: router(config)# commit
```

確認

show mpls traffic-eng fast-reroute database コマンドを使用して、Fast Reroute 設定を確認します。

```
RP/0/RP0/cpu 0: router# show mpls traffic-eng fast-reroute database

Tunnel head FRR information:
Tunnel      Out intf/label                    FRR intf/label      Status
----------  ----------------                  ----------------    -------
tt4000       HundredGigabitEthernet 0/0/1/0:34     tt1000:34        Ready
tt4001       HundredGigabitEthernet 0/0/1/0:35     tt1001:35        Ready
tt4002       HundredGigabitEthernet 0/0/1/0:36     tt1001:36        Ready
```

関連項目

- MPLS-TE の設定 （28 ページ）
- 自動トンネル バックアップの設定 （32 ページ）
- ネクスト ホップ バックアップ トンネルの設定 （33 ページ）
- MPLS-TE 機能の詳細 （46 ページ）

自動トンネル バックアップの設定

MPLS トラフィック エンジニアリング自動トンネル バックアップ機能を使用すると、ルータは、MPLS-TE トンネルを静的に構築するのではなく、MPLS TE トンネルを使用して設定されたインターフェイス上でバックアップ トンネルを動的に構築できます。

MPLS-TE 自動トンネル バックアップ機能には、次の利点があります。

- バックアップ トンネルは自動的に構築されるため、ユーザが各バックアップ トンネルを事前に設定し、保護対象のインターフェイスにそのバックアップ トンネルを割り当てる必要はありません。
- 保護は拡張されます。ただし、TE トンネルを使用していない IP トラフィック、または TE トンネルを使用していないラベル配布プロトコル（LDP）ラベルは FRR で保護されません。

一連の TE トンネル属性を指定する TE Attribute-set テンプレートは、自動トンネルのヘッドエンドでローカルに設定されます。コントロール プレーンは、対応する TE トンネルの自動プロビジョニングをトリガーします。そのトンネルの特性がそれぞれの Attribute-set で指定されます。

設定例

次の例では、インターフェイスに自動トンネル バックアップを設定し、自動トンネル用の attribute-set テンプレートを指定します。この例では、未使用のバックアップ トンネルは、タイマーを使用して 20 分ごとに削除され、トンネル インターフェイス番号の範囲も指定されています。

```
RP/0/RP0/cpu 0: router # configure
RP/0/RP0/cpu 0: router(config)# mpls traffic-eng
RP/0/RP0/cpu 0: router(config-mpls-te)# interface HundredGigabitEthernet0/9/0/0
RP/0/RP0/cpu 0: router(config-mpls-te-if)# auto-tunnel backup
RP/0/RP0/cpu 0: router(config-mpls-te-if-auto-backup)# attribute-set ab
RP/0/RP0/cpu 0: router(config-mpls-te)# auto-tunnel backup timers removal unused 20
RP/0/RP0/cpu 0: router(config-mpls-te)# auto-tunnel backup tunnel-id min 6000 max 6500
RP/0/RP0/cpu 0: router(config)# commit
```

確認

次の例に、自動バックアップ トンネル設定のサンプル出力を示します。

```
RP/0/RP0/cpu 0: router# show mpls traffic-eng tunnels brief

              TUNNEL NAME          DESTINATION      STATUS  STATE
               tunnel-te0            200.0.0.3          up  up
               tunnel-te1            200.0.0.3          up  up
               tunnel-te2            200.0.0.3          up  up
               tunnel-te50           200.0.0.3          up  up
             *tunnel-te60         200.0.0.3          up  up
                *tunnel-te70         200.0.0.3          up  up
                *tunnel-te80         200.0.0.3          up  up
```

関連項目

- Fast Reroute の設定 （30 ページ）
- ネクスト ホップ バックアップ トンネルの設定 （33 ページ）
- MPLS-TE 機能の詳細 （46 ページ）

ネクスト ホップ バックアップ トンネルの設定

LSP パスの単一リンクのみをバイパスするバックアップ トンネルは、障害ポイントを超えた LSP のネクスト ホップで終了するため、ネクスト ホップ（NHOP）バックアップ トンネルと呼ばれます。パス上のリンクに障害が発生した場合、バックアップ トンネルは、LSP のトラフィックをネクスト ホップにリルートする（障害の発生したリンクをバイパスする）ことによって LSP を保護します。

設定例

次の例では、インターフェイスにネクスト ホップ バックアップ トンネルを設定し、自動トンネル用の attribute-set テンプレートを指定します。この例では、未使用のバックアップ トンネルは、タイマーを使用して 20 分ごとに削除され、トンネル インターフェイス番号の範囲も指定されています。

```
RP/0/RP0/cpu 0: router # configure
RP/0/RP0/cpu 0: router(config)# mpls traffic-eng
RP/0/RP0/cpu 0: router(config-mpls-te)# interface HundredGigabitEthernet0/0/1/00/9/0/0
RP/0/RP0/cpu 0: router(config-mpls-te-if)# auto-tunnel backup nhop-only
RP/0/RP0/cpu 0: router(config-mpls-te-if-auto-backup)# attribute-set ab
RP/0/RP0/cpu 0: router(config-mpls-te)# auto-tunnel backup timers removal unused 20
RP/0/RP0/cpu 0: router(config-mpls-te)# auto-tunnel backup tunnel-id min 6000 max 6500
RP/0/RP0/cpu 0: router(config)# commit
```

関連項目

- 自動トンネル バックアップの設定 （32 ページ）
- Fast Reroute の設定 （30 ページ）
- MPLS-TE 機能の詳細 （46 ページ）

SRLG ノード保護の設定

MPLS トラフィック エンジニアリング内の共有リスク リンク グループ（SRLG）は、ネットワーク内のリンクが共通のリソースを共有する状況を指します。これらのリンクには、共有リスクがあります。つまり、1 つのリンクで障害が発生すると、グループ内の別のリンクでも障害が発生する可能性があります。

OSPF と IS-IS は、サブタイプ、長さ、値（サブ TLV）を使用して、SRLG 値情報（帯域幅のアベイラビリティやアフィニティなどの他の TE リンク属性を含む）をフラッディングして、ネットワーク内のすべてのルータが各リンクの SRLG 情報を持つようにします。

MPLS-TE SRLG 機能は、バックアップ トンネルの作成中に保護しているインターフェイスと同じ SRLG 内のリンクを使用しないようにすることで、バックアップ トンネル パスの選択を強化します。

設定例

次の例では、バックアップ トンネルを作成し、保護されたノードの IP アドレスを明示パスから除外します。

```
RP/0/RP0/cpu 0: router # configure
RP/0/RP0/cpu 0: router(config)# mpls traffic-eng
RP/0/RP0/cpu 0: router(config-mpls-te)# interface HundredGigabitEthernet0/9/0/0
RP/0/RP0/cpu 0: router(config-mpls-te-if)# backup-path tunnl-te 2
RP/0/RP0/cpu 0: router(config-mpls-te-if)# exit
RP/0/RP0/cpu 0: router(config)# interface tunnel-te 2
RP/0/RP0/cpu 0: router(config-if)# ipv4 unnumbered Loopback0
RP/0/RP0/cpu 0: router(config-if)# path-option 1 explicit name backup-srlg
RP/0/RP0/cpu 0: router(config-if)# destination 192.168.92.125
RP/0/RP0/cpu 0: router(config-if)# exit
RP/0/RP0/cpu 0: router(config)# explicit-path name backup-srlg-nodep
RP/0/RP0/cpu 0: router(config-if)# index 1 exclude-address 192.168.91.1
RP/0/RP0/cpu 0: router(config-if)# index 1 exclude-srlg 192.168.92.2
RP/0/RP0/cpu 0: router(config)# commit
```

関連項目

- Fast Reroute の設定 （30 ページ）
- MPLS-TE 機能の詳細 （46 ページ）

先行標準 DS-TE の設定

通常のトラフィック エンジニアリングでは、異なるトラフィック クラスへの帯域幅保証は提供されません。単一帯域幅の制約は、すべてのトラフィックで共有される通常の TE で使用されます。MPLS DS-TE では、MPLS 対応インターフェイスに複数の帯域幅制約を設定できます。これらの帯域幅は、その制約を使用してトラフィック クラスの要件に基づいて扱うことができます。Cisco IOS XR ソフトウェアは、2 つの DS-TE モード、先行標準と IETF をサポートしています。先行標準 DS-TE では、RSVP シグナリングおよび IGP アドバタイズメントにシスコ独自のメカニズムを採用しています。この DS-TE モードには、サードパーティ ベンダー製機器との相互運用性はありません。先行標準 DS-TE をイネーブルにするには、MPLS 対応のインターフェイスでサブプール帯域幅の値を設定する必要があります。

先行標準 Diff-Serve TE モードでは、グローバルプールとサブプールの 2 つの帯域幅プールを持つ単一の帯域幅制約モデル Russian Doll Model（RDM）がサポートされます。

はじめる前に

次に、先行標準 DS-TE トンネルを設定するための前提条件を示します。

- 隣接ルータにはルータ ID が必要です。

- リンクを正常に行うには、リンクのいずれかの側に安定したルータ ID が必要です。ルータ ID をルータに割り当てない場合、デフォルトでグローバル ルータ ID に設定されます。デフォルトのルータ ID は変更されることがあり、不安的なリンクの原因となる可能性があります。

設定例

次の例では、先行標準の DS-TE トンネルを設定します。

```
RP/0/RP0/cpu 0: router # configure
RP/0/RP0/cpu 0: router(config)# rsvp interface HundredGigabitEthernet 0/9/0/0
RP/0/RP0/cpu 0: router(config-rsvp-if)# bandwidth 100 150 sub-pool 50
RP/0/RP0/cpu 0: router(config-rsvp-if)# exit
RP/0/RP0/cpu 0: router(config)# interface tunnel-te 2
RP/0/RP0/cpu 0: router(config-if)# signalled bandwidth sub-pool 10
RP/0/RP0/cpu 0: router(config)# commit
```

確認

show mpls traffic-eng topology コマンドを使用して先行標準 DS-TE トンネル設定を確認します。

関連項目

- RDM を使用した IETF DS-TE トンネルの設定 （35 ページ）
- MAM を使用した IETF DS-TE トンネルの設定 （36 ページ）
- MPLS-TE 機能の詳細 （46 ページ）

RDM を使用した IETF DS-TE トンネルの設定

IETF DS-TE モードは、RSVP および IGP に IETF 定義の拡張機能を使用します。このモードには、サードパーティ ベンダー製機器との相互運用性があります。

IETF モードは、Russian Doll Model（RDM）および Maximum Allocation Model（MAM）を含む複数の帯域幅制限モデルをサポートしており、どちらのモデルでも 2 つの帯域幅プールを使用します。IETF DS-TE ネットワークでは、すべてのノードで同一の帯域幅制約モデルを設定する必要があります。

はじめる前に

次に、RDM を使用して IETF モード DS-TE トンネルを作成するための前提条件を示します。

- 隣接ルータにはルータ ID が必要です。
- リンクを正常に行うには、リンクのいずれかの側に安定したルータ ID が必要です。ルータ ID をルータに割り当てない場合、デフォルトでグローバル ルータ ID に設定されます。デフォルトのルータ ID は変更されることがあり、不安的なリンクの原因となる可能性があります。

設定例

次の例では、RDM を使用して IETF DS-TE トンネルを設定します。

```
RP/0/RP0/cpu 0: router # configure
RP/0/RP0/cpu 0: router(config)# rsvp interface HundredGigabitEthernet 0/9/0/0
RP/0/RP0/cpu 0: router(config-rsvp-if)# bandwidth rdm 100 150
RP/0/RP0/cpu 0: router(config-rsvp-if)# exit
RP/0/RP0/cpu 0: router(config)# mpls traffic-eng
RP/0/RP0/cpu 0: router(config-mpls-te)# ds-te mode ietf
RP/0/RP0/cpu 0: router(config-mpls-te)# exit
RP/0/RP0/cpu 0: router(config)# interface tunnel-te 2
RP/0/RP0/cpu 0: router(config-if)# signalled bandwidth sub-pool 10 class-type 1
RP/0/RP0/cpu 0: router(config)# commit
```

確認

show mpls traffic-eng topology コマンドを使用して、RDM 設定を使用した IETF DS-TE トンネルを確認します。

関連項目

- 先行標準 DS-TE の設定 （34 ページ）
- MAM を使用した IETF DS-TE トンネルの設定 （36 ページ）
- MPLS-TE 機能の詳細 （46 ページ）

MAM を使用した IETF DS-TE トンネルの設定

IETF DS-TE モードは、RSVP および IGP に IETF 定義の拡張機能を使用します。このモードには、サードパーティ ベンダー製機器との相互運用性があります。IETF モードは、Russian Doll Model（RDM）および Maximum Allocation Model（MAM）を含む複数の帯域幅制限モデルをサポートしており、どちらのモデルでも 2 つの帯域幅プールを使用します。

設定例

次の例では、MAM を使用して IETF DS-TE トンネルを設定します。

```
RP/0/RP0/cpu 0: router # configure
RP/0/RP0/cpu 0: router(config)# rsvp interface HundredGigabitEthernet 0/9/0/0
RP/0/RP0/cpu 0: router(config-rsvp-if)# bandwidth mam max-reservable-bw 1000 bc0 600 bc1
 400
RP/0/RP0/cpu 0: router(config-rsvp-if)# exit
RP/0/RP0/cpu 0: router(config)# mpls traffic-eng
RP/0/RP0/cpu 0: router(config-mpls-te)# ds-te mode ietf
RP/0/RP0/cpu 0: router(config-mpls-te)# ds-te bc-model mam
RP/0/RP0/cpu 0: router(config-mpls-te)# exit
RP/0/RP0/cpu 0: router(config)# interface tunnel-te 2
RP/0/RP0/cpu 0: router(config-if)# signalled bandwidth sub-pool 10
RP/0/RP0/cpu 0: router(config)# commit
```

確認

show mpls traffic-eng topology コマンドを使用して、MAM 設定を使用した IETF DS-TE トンネルを確認します。

関連項目

- RDM を使用した IETF DS-TE トンネルの設定 （35 ページ）
- 先行標準 DS-TE の設定 （34 ページ）
- MPLS-TE 機能の詳細 （46 ページ）

柔軟な名前ベースのトンネル制約の設定

MPLS-TE の柔軟な名前ベースのトンネル制約は、MPLS-TE トンネルのパスを計算するために、リンク属性とパス アフィニティを簡単かつより柔軟に設定する方法を提供します。

従来の TE では、リンクは、Open Shortest Path First（OSPF）などの内部ゲートウェイ プロトコル（IGP）を使用して、TE リンクステート パラメータでフラッディングされる attribute-flags で設定されます。

MPLS-TE の柔軟な名前ベースのトンネル制約を使用すると、32 ビットの 16 進数値の代わりに、アフィニティと attribute-flag 属性のために最大 32 個のカラー名を割り当てる（マップする）ことができます。マッピングの定義後に、対応するカラー名で属性を参照することができます。

設定例

次の例では、トンネルをアフィニティ制約に関連付ける方法の割り当てを示しています。

```
RP/0/RP0/cpu 0: router# configure
RP/0/RP0/cpu 0: router(config)# mpls traffic-eng
RP/0/RP0/cpu 0: router(config-mpls-te)# affinity-map red 1
RP/0/RP0/cpu 0: router(config-mpls-te)# interface HundredGigabitEthernet0/9/0/0
RP/0/RP0/cpu 0: router(config-mpls-te-if)# attribute-names red
RP/0/RP0/cpu 0: router(config)# interface tunnel-te 2
RP/0/RP0/cpu 0: router(config-if)# affinity include red
RP/0/RP0/cpu 0: router(config)# commit
```

自動帯域幅の設定

自動帯域幅では、測定されたトラフィックに基づいて帯域幅予約を動的に調整できます。MPLS-TE の自動帯域幅は、トンネル インターフェイスのトラフィック レートを監視し、トンネルインターフェイスの帯域幅のサイズを変更して、トンネル内のトラフィックと厳密にそろえます。MPLS-TE 自動帯域幅は、すべてのヘッドエンドルータで個々のラベル スイッチドパス（LSP）上で設定されます。

次の表に、自動帯域幅設定の一部として設定できるパラメータを示します。

表 *1:* 自動帯域幅パラメータ

帯域幅のパラメータ	説明
適用の頻度	トンネル帯域幅をトンネルごとに変更する頻度を設定します。デフォルト値は 24 時間です。
帯域幅制限	トンネルに設定する最小および最大自動帯域幅を設定します。
帯域幅収集頻度	自動帯域幅を調整しないで帯域幅収集をイネーブルにします。デフォルト値は 5 分です。
オーバーフローしきい値	トンネルのオーバーフロー検出を設定します。
調整しきい値	調整をトリガーするトンネル帯域幅変更しきい値を設定します。

設定例

次の例では、MPLS-TE トンネル インターフェイスで自動帯域幅をイネーブルにし、次の自動帯域幅変数を設定します。

- 適用の頻度
- 帯域幅制限
- 調整しきい値
- オーバーフロー検出

```
RP/0/RP0/cpu 0: router# configure
RP/0/RP0/cpu 0: router(config)# interface tunnel-te 1
RP/0/RP0/cpu 0: router(config-if)# auto-bw
RP/0/RP0/cpu 0: router(config-if-tunte-autobw)# application 1000
RP/0/RP0/cpu 0: router(config-if-tunte-autobw)# bw-limit min 30 max 1000
RP/0/RP0/cpu 0: router(config-if-tunte-autobw)# adjustment-threshold 50 min 800
RP/0/RP0/cpu 0: router(config-if-tunte-autobw)# overflow threshold 100 limit 1
RP/0/RP0/cpu 0: router(config)# commit
```

確認

show mpls traffic-eng tunnels auto-bw brief コマンドを使用して、自動帯域幅設定を確認します。

```
RP/0/RP0/cpu 0: router# show mpls traffic-eng tunnels auto-bw brief

  Tunnel    LSP  Last appl  Requested  Signalled    Highest    Application
  Name       ID   BW(kbps)   BW(kbps)   BW(kbps)   BW(kbps)      Time Left
  ------------- ------ ---------- ---------- ---------- ---------- --------------
  tunnel-te1      5          500         300        420          1h 10m
```

関連項目

- MPLS-TE 機能の詳細 （46 ページ）

自動トンネル メッシュの設定

MPLS-TE 自動トンネル メッシュ（自動メッシュ）機能を使用すると、最小の MPLS トラフィック エンジニアリング設定で TE ポイントツーポイント（P2P）トンネルのフル メッシュを自動的に設定できます。1 つまたは複数のメッシュグループを設定でき、各メッシュ グループでは、宛先が示されている宛先リスト（IPv4 Prefix-List）が必要です。これは、そのメッシュグループにトンネルを作成するための宛先として使用されます。

MPLS TE 自動メッシュ タイプ attribute-set（テンプレート）を設定して、メッシュグループに関連付けることができます。ラベル スイッチング ルータ（LSR）は、この attribute-set で定義されたトンネル プロパティを使用してトンネルを作成できます。

自動トンネル メッシュ設定によって、ネットワークの初期設定が最小限に抑えられます。これらの LSR 間でさらに TE トンネルのフル メッシュが作成される TE LSR で、トンネル プロパティ テンプレートとメッシュグループまたは宛先リストを設定できます。新しい TE LSR がネットワークで追加されるたびに TE トンネルのフル メッシュを確立するために、既存の各 TE LSR を再設定する必要がなくなります。

設定例

次の例では、自動トンネル メッシュ グループを設定し、メッシュグループ内のトンネルに属性を指定します。

```
RP/0/RP0/cpu 0: router # configure
RP/0/RP0/cpu 0: router(config)# mpls traffic-eng
RP/0/RP0/cpu 0: router(config-mpls-te)# auto-tunnel mesh
RP/0/RP0/cpu 0: router(config-mpls-te-auto-mesh)# tunnel-id min 1000 max 2000
RP/0/RP0/cpu 0: router(config-mpls-te-auto-mesh)# group 10
RP/0/RP0/cpu 0: router(config-mpls-te-auto-mesh-group)# attribute-set 10
RP/0/RP0/cpu 0: router(config-mpls-te-auto-mesh-group)# destination-list dl-65
RP/0/RP0/cpu 0: router(config-mpls-te)# attribute-set auto-mesh 10
RP/0/RP0/cpu 0: router(config-mpls-te-attribute-set)# autoroute announce
RP/0/RP0/cpu 0: router(config-mpls-te-attribute-set)# auto-bw collect-bw-only
RP/0/RP0/cpu 0: router(config)# commit
```

確認

show mpls traffic-eng auto-tunnel mesh コマンドを使用して、自動トンネル メッシュ設定を確認します。

```
RP/0/RP0/cpu 0: router# show mpls traffic-eng auto-tunnel mesh

Auto-tunnel Mesh Global Configuration:
  Unused removal timeout: 1h 0m 0s
  Configured tunnel number range: 1000-2000

Auto-tunnel Mesh Groups Summary:
  Mesh Groups count: 1
  Mesh Groups Destinations count: 3
```

```
  Mesh Groups Tunnels count:
    3 created, 3 up, 0 down, 0 FRR enabled

Mesh Group: 10 (3 Destinations)
  Status: Enabled
  Attribute-set: 10
  Destination-list: dl-65 (Not a prefix-list)
  Recreate timer: Not running
       Destination    Tunnel ID    State  Unused timer
  ----------------  -----------  -------  ------------
       192.168.0.2         1000     up   Not running
       192.168.0.3         1001     up   Not running
       192.168.0.4         1002     up   Not running
  Displayed 3 tunnels, 3 up, 0 down, 0 FRR enabled

Auto-mesh Cumulative Counters:
  Last cleared: Wed Oct  3 12:56:37 2015 (02:39:07 ago)
                       Total
  Created:                 3
  Connected:               0
  Removed (unused):        0
  Removed (in use):        0
  Range exceeded:          0
```

MPLS トラフィック エンジニアリング エリア間トンネリングの設定

MPLS TEエリア間トンネリング機能を使用すると、複数の内部ゲートウェイプロトコル（IGP）のエリアとレベルにまたがる MPLS TE トンネルを確立できます。この機能により、トンネルのヘッドエンド ルータとテールエンド ルータの両方が同じエリア内になければならないという制限がなくなります。IGP は、Intermediate System-to-Intermediate System（IS-IS）または Open Shortest Path First（OSPF）のいずれかになります。エリア間トンネルを設定するには、ヘッドエンド ルータで、LSP が next-address loose コマンドを使用して通過する必要がある各エリア境界ルータ（ABR）を識別するトンネル ラベル スイッチド パス（LSP）への緩やかにルーティングされた明示パスを指定します。指定した明示パス上のヘッドエンド ルータと ABR は、ルーズ ホップを展開し、それぞれが次の ABR またはトンネル宛先へのパス セグメントを計算します。

設定例

次の例では、ABR をヘッドエンド ルータにルーズ アドレスとして設定した IPv4 明示パスを設定します。

```
RP/0/RP0/cpu 0: router# configure
RP/0/RP0/cpu 0: router(config)# explicit-path name interarea1
RP/0/RP0/cpu 0: router(config-expl-path)# index1 next-address loose ipv4 unicast
172.16.255.129
RP/0/RP0/cpu 0: router(config-expl-path)# index 2 next-address loose ipv4 unicast
172.16.255.131
RP/0/RP0/cpu 0: router(config)# interface tunnel-te1
RP/0/RP0/cpu 0: router(config-if)# ipv4 unnumbered Loopback0
RP/0/RP0/cpu 0: router(config-if)# destination 172.16.255.2
RP/0/RP0/cpu 0: router(config-if)# path-option 10 explicit name interarea1
RP/0/RP0/cpu 0: router(config)# commit
```

関連項目

- MPLS-TE 機能の詳細 （46 ページ）

ポリシーベース トンネル選択の設定

PBTS の設定は、着信トラフィックを分類基準（DSCP）に基づいて特定の TE トンネルに誘導するプロセスです。トラフィック転送の決定は、分類されたトラフィック クラスと宛先ネットワーク アドレスに基づいて行われます。次の項では、MPLS-TE トンネル ネットワーク上で PBTS を設定する手順を示します。

1. 分類基準に基づいてクラスマップを定義します。
2. 分類されたトラフィックのルールを作成して、ポリシーマップを定義します。
3. 各タイプの入力トラフィックに転送クラスを関連付けます。
4. このサービスポリシーを適用して、入力インターフェイスで PBTS を有効にします。
5. 宛先に 1 つ以上の出力 MPLS-TE トンネルを作成します（優先順位に基づいてパケットを伝送するため）。
6. 出力 MPLS-TE トンネルを転送クラスに関連付けます。

PBTS の詳細については、「*MPLS* トラフィック エンジニアリングの実装」の章の「ポリシーベース トンネル選択 （50 ページ）」を参照してください。

設定例

次のセクションでは、PBTS の実装を示しています。

```
RP/0/RP0/cpu 0: router#configure
/* Class-map; classification using DSCP */
RP/0/RP0/cpu 0: router(config)# class-map match-any AF41-Class
RP/0/RP0/cpu 0: router(config-cmap)# match dscp AF41
RP/0/RP0/cpu 0: router(config-cmap)# exit

/* Policy-map */
RP/0/RP0/cpu 0: router(config)# policy-map INGRESS-POLICY
RP/0/RP0/cpu 0: router(config-pmap)# class AF41-Class
/* Associating forward class */
RP/0/RP0/cpu 0: router(config-pmap-c)# set forward-class 1
RP/0/RP0/cpu 0: router(config-pmap-c)# exit
RP/0/RP0/cpu 0: router(config-pmap)# exit

RP/0/RP0/cpu 0: router(config)# interface GigabitEthernet0/9/0/0
/* Applying service-policy to ingress interface */
RP/0/RP0/cpu 0: router(config-if)# service-policy input INGRESS-POLICY
RP/0/RP0/cpu 0: router(config-if)# ipv4 address 10.1.1.1 255.255.255.0
RP/0/RP0/cpu 0: router(config-if)# exit

/* Creating TE-tunnels to carry traffic based on priority */
RP/0/RP0/cpu 0: router(config)# interface tunnel-te61
RP/0/RP0/cpu 0: router(config-if)# ipv4 unnumbered Loopback0
RP/0/RP0/cpu 0: router(config-if)# signalled-bandwidth 1000
RP/0/RP0/cpu 0: router(config-if)# autoroute announce
RP/0/RP0/cpu 0: router(config-if)# destination 10.20.20.1
RP/0/RP0/cpu 0: router(config-if)# record route
```

```
/* Associating egress TE tunnels to forward class */
RP/0/RP0/cpu 0: router(config-if)# forward-class 1
RP/0/RP0/cpu 0: router(config-if)# path-option 1 explicit identifier 61
RP/0/RP0/cpu 0: router(config-if)# exit
```

確認

show mpls forwarding tunnels コマンドを使用して PBTS の設定を確認します。

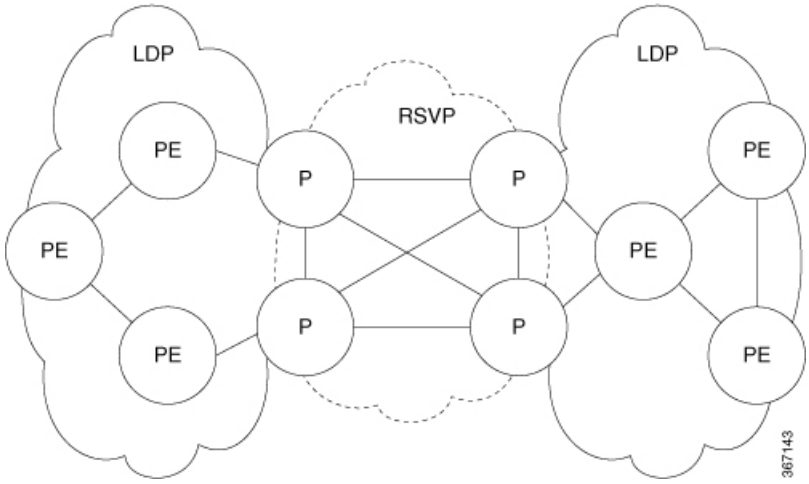
```
RP/0/RP0/CPU0:ios# show mpls forwarding tunnels 10 detail
Tue May 16 01:18:19.681 UTC

Tunnel        Outgoing    Outgoing     Next Hop        Bytes
Name          Label       Interface                    Switched
------------- ----------- ------------ --------------- ------------
tt10              Exp-Null-v4 Te0/0/0/16   20.20.17.21     0
     Updated: May 11 19:31:54.716
     Version: 483, Priority: 2
     Label Stack (Top -> Bottom): { 0 }
     NHID: 0x0, Encap-ID: N/A, Path idx: 0, Backup path idx: 0, Weight: 0
     MAC/Encaps: 14/18, MTU: 1500
     Packets Switched: 0

  Interface:
    Name: tunnel-te10 (ifhandle 0x0800005c)
    Local Label: 64016, Forwarding Class: 1, Weight: 0

    Packets/Bytes Switched: 0/0
```

LDP over MPLS-TE の設定

LDP および RSVP-TE は、MPLS ネットワークで LSP を確立するために使用されるシグナリング プロトコルです。LDP は設定が簡単で信頼性が高いですが、トラフィックの輻輳を回避するのに役立つ RSVP のトラフィック エンジニアリング機能はありません。LDP over MPLS-TE 機能は、LDP と RSVP 両方の利点を組み合わせています。LDP over MPLS-TE では、LDP シグナリング ラベルスイッチド パス（LSP）が RSVP-TE を使用して確立された TE トンネルを通過します。

次の図は、LDP over MPLS-TE の使用例を示しています。この図では、プロバイダー エッジ（PE）ルータとプロバイダー（P）ルータ間のシグナリング プロトコルとして LDP が使用されています。RSVP-TE は、P ルータ間のシグナリング プロトコルとして LSP を確立するために使用されます。LDP は RSVP-TE LSP 上でトンネリングされます。

図 *6 : LDP over MPLS-TE*



LDP over MPLS-TE の制約事項とガイドライン

Cisco IOS-XR リリース 6.3.2 では、この機能に次の制限事項とガイドラインが適用されます。

- BGP ネイバーが TE トンネルのヘッドまたはテール ノードにある場合、LDP over MPLS-TE 経由の MPLS サービスがサポートされます。
- TE ヘッドエンドルータがそのサービスの通過ポイントとして機能している場合、LDP over MPLS-TE 経由の MPLS サービスがサポートされます。
- TELS ヘッドエンドから MPLS サービスが発信されていても、TE トンネルが BGP ピアより前に終了している場合、LDP over MPLS-TE 機能はサポートされません。
- **hw-module fib mpls ldp lsr-optimized** コマンドを使用して LDP 最適化を有効にした場合は、次の制限が適用されます。
  - EVPN はサポートされません。
  - 任意のプレフィックスまたはラベルに対して、すべての発信パスを LDP 対応にする必要があります。
- すでに設定されている EVPN、MPLS VPN、L2VPN などの機能が正常に動作しない可能性があるため、プロバイダー エッジ（PE）ルータでは **hw-module fib mpls ldp lsr-optimized** コマンドを使用しないでください。

設定例

次の例に、MPLS-TE トンネルをプロバイダー ルータ P1 ～ P2 に設定し、LDP over MPLS-TE を有効にする方法を示します。この例では、P1 からのトンネルの宛先は P2 のループバックとして設定されています。

```
RP/0/RP0/CPU0:router # configure
RP/0/RP0/CPU0:router(config)# interface tunnel-te 1
RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0
```

```
RP/0/RP0/CPU0:router(config-if)# autoroute announce
RP/0/RP0/CPU0:router(config-if)# destination 4.4.4.4
RP/0/RP0/CPU0:router(config-if)# path-option 1 dynamic
RP/0/RP0/CPU0:router(config-if)# exit
RP/0/RP0/CPU0:Router(config)#  mpls ldp
RP/0/RP0/CPU0:Router(config-ldp)#  router-id 192.168.1.1
RP/0/RP0/CPU0:Router(config-ldp)#  interface TenGigE 0/0/0/0
RP/0/RP0/CPU0:Router(config-ldp-if)# interface tunnel-te 1
RP/0/RP0/CPU0:Router(config-ldp-if)# exit
```

MPLS-TE パス保護の設定

パス保護は、MPLS-TE トンネルのエンドツーエンド障害回復メカニズムを提供します。セカンダリ ラベル スイッチドパス（LSP）をあらかじめ確立しておくと、トンネルの TE トラフィックを伝送する保護 LSP を障害から保護できます。保護された LSP に障害がある場合、送信元ルータは、トンネルのトラフィックを一時的に伝送するセカンダリ LSP をすぐにイネーブルにします。フェールオーバーは、LSP ヘッドエンドに送信された RSVP エラー メッセージによってトリガーされます。ヘッド エンドはこのエラー メッセージを受信すると、セカンダリ トンネルに切り替えます。セカンダリ LSP で障害が発生した場合は、セカンダリ パスの障害がクリアされるまでトンネルのパス保護は機能しなくなります。パス保護は、単一のエリア（OSPF または IS-IS）、外部 BGP（eBGP）、およびスタティック ルート内で使用できます。明示的および動的なパスオプションはどちらも、MPLS-TE パス保護機能がサポートされています。保護されたオプションで同じ属性または帯域幅の要件が設定されていることを確認する必要があります。

はじめる前に

パス保護を有効にするには、次の前提条件が必要です。

- ネットワークで MPLS-TE、Cisco Express Forwarding、および Intermediate System-to-Intermediate System（IS-IS）または Open Shortest Path First（OSPF）がサポートされることを確認する必要があります。
- ルータで MPLS-TE を設定する必要があります。

設定例

次の例では、mpls-te トンネルにパス保護を設定する方法を設定します。パス保護を設定するには、プライマリ パスオプションが必要です。この構成では、R1 はヘッドエンド ルータであり、R3 はトンネルのテールエンド ルータであり、R2 と R4 は中間ルータです。この例では、6 つの明示パスと 1 つの動的パスがパス保護用に作成されています。プライマリ パスに対して最大 8 つのパス保護オプションを設定できます。

```
RP/0/RP0/cpu 0: router # configure
RP/0/RP0/cpu 0: router(config)# interface tunnel-te 0
RP/0/RP0/cpu 0: router(config-if)# destination 192.168.3.3
RP/0/RP0/cpu 0: router(config-if)# ipv4 unnumbered Loopback0
RP/0/RP0/cpu 0: router(config-if)# autoroute announce
RP/0/RP0/cpu 0: router(config-if)# path-protection
RP/0/RP0/cpu 0: router(config-if)# path-option 1 explicit name r1-r2-r3-00 protected-by
 2
RP/0/RP0/cpu 0: router(config-if)# path-option 2 explicit name r1-r2-r3-01 protected-by
```

```
 3
RP/0/RP0/cpu 0: router(config-if)# path-option 3 explicit name r1-r4-r3-01 protected-by
 4
RP/0/RP0/cpu 0: router(config-if)# path-option 4 explicit name r1-r3-00 protected-by 5
RP/0/RP0/cpu 0: router(config-if)# path-option 5 explicit name r1-r2-r4-r3-00 protected-by
 6
RP/0/RP0/cpu 0: router(config-if)# path-option 6 explicit name r1-r4-r2-r3-00 protected-by
 7
RP/0/RP0/cpu 0: router(config-if)# path-option 7 dynamic
RP/0/RP0/cpu 0: router(config-if)# exit
RP/0/RP0/cpu 0: router(config)# commit
```

確認

show mpls traffic-eng tunnels コマンドを使用して、MPLS-TE パス保護設定を確認します。

```
RP/0/RP0/cpu 0: router# show mpls traffic-eng tunnels 0
Fri Oct 13 16:24:39.379 UTC
Name: tunnel-te0  Destination: 192.168.92.125  Ifhandle:0x8007d34
  Signalled-Name: router
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected
    path option 1,  type explicit r1-r2-r3-00 (Basis for Setup, path weight 2)
      Protected-by PO index: 2
    path option 2,  type explicit r1-r2-r3-01 (Basis for Standby, path weight 2)
      Protected-by PO index: 3
    path option 3,  type explicit r1-r4-r3-01
      Protected-by PO index: 4
    path option 4,  type explicit r1-r3-00
      Protected-by PO index: 5
    path option 5,  type explicit r1-r2-r4-r3-00
      Protected-by PO index: 6
    path option 6,  type explicit r1-r4-r2-r3-00
      Protected-by PO index: 7
    path option 7,  type dynamic
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 0 kbps  CT0
    Creation Time: Fri Oct 13 15:05:28 2017 (01:19:11 ago)
  Config Parameters:
    Bandwidth:        0 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (global)
    Path Selection:
      Tiebreaker: Min-fill (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Delay-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute:  enabled  LockDown: disabled   Policy class: not set
    Forward class: 0 (not enabled)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    Loadshare:          0 equal loadshares
    Auto-bw: disabled
    Fast Reroute: Disabled, Protection Desired: None
    Path Protection: Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    Soft Preemption: Disabled
  History:
    Tunnel has been up for: 01:14:13 (since Fri Oct 13 15:10:26 UTC 2017)
    Current LSP:
      Uptime: 01:14:13 (since Fri Oct 13 15:10:26 UTC 2017)
    Reopt. LSP:
      Last Failure:
```

```
        LSP not signalled, identical to the [CURRENT] LSP
        Date/Time: Fri Oct 13 15:08:41 UTC 2017 [01:15:58 ago]
    Standby Reopt LSP:
      Last Failure:
        LSP not signalled, identical to the [STANDBY] LSP
        Date/Time: Fri Oct 13 15:08:41 UTC 2017 [01:15:58 ago]
        First Destination Failed: 192.3.3.3
    Prior LSP:
      ID: 8 Path Option: 1
      Removal Trigger: path protection switchover
    Standby LSP:
      Uptime: 01:13:56 (since Fri Oct 13 15:10:43 UTC 2017)
  Path info (OSPF 1 area 0):
  Node hop count: 2
  Hop0: 192.168.1.2
  Hop1: 192.168.3.1
  Hop2: 192.168.3.2
  Hop3: 192.168.3.3
  Standby LSP Path info (OSPF 1 area 0), Oper State: Up :
  Node hop count: 2
  Hop0: 192.168.2.2
  Hop1: 192.168.3.1
  Hop2: 192.168.3.2
  Hop3: 192.168.3.3
Displayed 1 (of 4001) heads, 0 (of 0) midpoints, 0 (of 0) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
```

MPLS-TE 機能の詳細

MPLS TE Fast Reroute リンクおよびノード保護

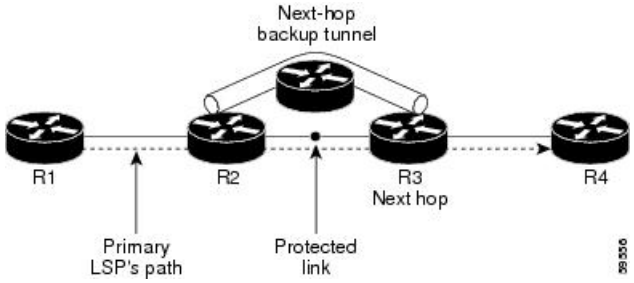
高速リルート（FRR）は、リンクおよびノードの障害から MPLS TE LSP を保護するためのメカニズムです。具体的には、障害ポイントの LSP をローカルに修復し、その LSP 上でのデータ フローを停止することなく、LSP のヘッドエンド ルータを新しく置き換えるエンドツーエンド LSP の確立を試行します。FRR は、保護対象 LSP を、障害が発生したリンクまたはノードをバイパスするバックアップ トンネル経由でリルートすることにより、LSP をローカルに修復します。

（注） FRR が 50 ミリ秒を超えると、トラフィック損失が発生する場合があります。

LSP のパスの単一リンクだけをバイパスするバックアップ トンネルが、リンク保護を提供します。パス上のリンクに障害が発生した場合、バックアップ トンネルは、LSP のトラフィックをネクスト ホップにリルートする（障害の発生したリンクをバイパスする）ことによって LSP を保護します。これらのトンネルは、障害ポイントの向こう側にある LSP のネクスト ホップで終端するため、ネクスト ホップ（NHOP）バックアップ トンネルと呼ばれます。

以下の図は、リンク保護を図示したものです。

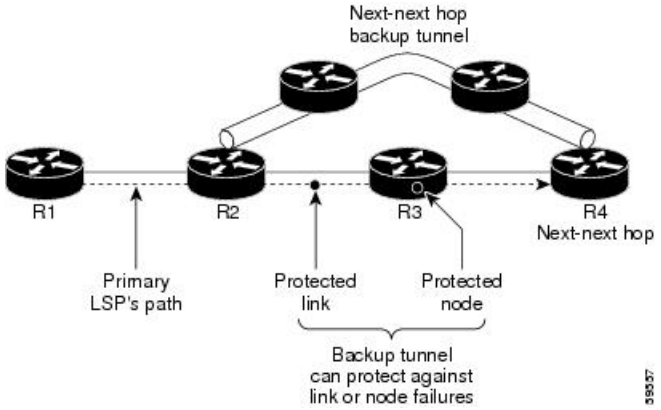
図 *7:* リンク保護



FRR により、LSP に対するノード保護が提供されます。LSP パス上のネクストホップ ノードをバイパスするバックアップ トンネルは、LSP パスのネクストホップ ノードの次のノードで終端して、ネクストホップ ノードをバイパスするため、ネクストネクストホップ（NNHOP）バックアップ トンネルと呼ばれます。LSP パス上のノードに障害が発生した場合は、NNHOP バックアップ トンネルが LSP を保護します。具体的には、障害のアップストリームにあるノードをイネーブルにして、障害の発生したノードの周囲の LSP とそのトラフィックをネクストネクストホップにリルートします。また、NNHOP バックアップ トンネルは、障害の発生したリンクおよびノードをバイパスするため、リンク障害からの保護も提供しています。

以下の図は、ノード保護を図示したものです。

図 *8:* ノード保護



MPLS-TE 転送隣接

MPLS TE転送隣接により、TE ラベルスイッチドパス（LSP）トンネルを、最短パス優先（SPF）アルゴリズムに基づいた内部ゲートウェイ プロトコル（IGP）ネットワーク内のリンクとして処理できます。Intermediate System-to-Intermediate System（IS-IS）と Open Shortest Path First（OSPF）の両方が IGP としてサポートされています。転送隣接は、ネットワーク内でのルータのロケーションに関係なく、ルータとルータの間に作成できます。ルータとルータは、間に何個かホップを入れて配置できます。

この結果、TE トンネルは IGP ネットワーク内にリンクとしてアドバタイズされ、トンネルのコストが関連付けられます。TE ドメインの外側にあるルータは、TE トンネルを参照し、その TE トンネルを使用して、ネットワーク内でトラフィックをルーティングするための最短パス

を計算します。TE トンネル インターフェイスは、他のリンクと同様に、IGP ネットワーク内にアドバタイズされます。これにより、ルータは、IGP 内のこれらのアドバタイズメントを使用して SPF を計算できるようになります。このことは、これらのアドバタイズメントがいずれかの TE トンネルのヘッドエンドでない場合も同様です。

自動帯域幅

自動帯域幅では、測定されたトラフィックに基づいて帯域幅予約を動的に調整できます。MPLS-TE 自動帯域幅は、すべてのヘッドエンドルータで個々のラベル スイッチド パス（LSP）上で設定されます。MPLS-TE の自動帯域幅は、トンネル インターフェイスのトラフィック レートを監視し、トンネルインターフェイスの帯域幅のサイズを変更して、トンネル内のトラフィックと厳密にそろえます。

MPLS-TE 自動帯域幅は、次の機能を実行できます。

- トンネル出力レートの定期的なポーリングをモニタします
- 一定の期間に測定された最大のレートを調整することで、トンネル帯域幅をサイズ変更します。

自動帯域幅用に設定された、トラフィック エンジニアリングを実行済みのすべてのトンネルで、設定可能なさまざまなパラメータに基づいて平均の出力レートがサンプリングされます。その後、特定の期間に通知された最大の平均出力レート、または設定されている最大帯域幅の値のいずれかに基づいて、トンネル帯域幅が自動的に再調整されます。

新しい帯域幅で LSP を再最適化すると、新しいパス要求が生成されます。新しい帯域幅が使用不可の場合、直前の適切な LSP が引き続き使用されます。この方法では、ネットワークでトラフィックの中断は発生しません。トンネルの最小または最大の帯域幅の値が設定されている場合、自動帯域幅によってシグナリングされる帯域幅は、これらの値の内に収まります。

トンネルでの出力レートは、MPLS-TE 自動帯域幅インターフェイス コンフィギュレーション モードで **application** コマンドを使用して設定された定期的な間隔で収集されます。適用期間タイマーが期限切れになったとき、および測定された帯域幅と現在の帯域幅の間の差分が調整しきい値を超えたときに、トンネルが再最適化されます。その後、帯域幅サンプルがクリアされ、次の間隔の新しい最大出力レートが記録されます。トンネルがシャットダウンされ、後で再度起動された場合、調整された帯域幅は失われ、トンネルは初期設定の帯域幅でアップ状態に戻ります。トンネルが復帰すると、適用期間がリセットされます。

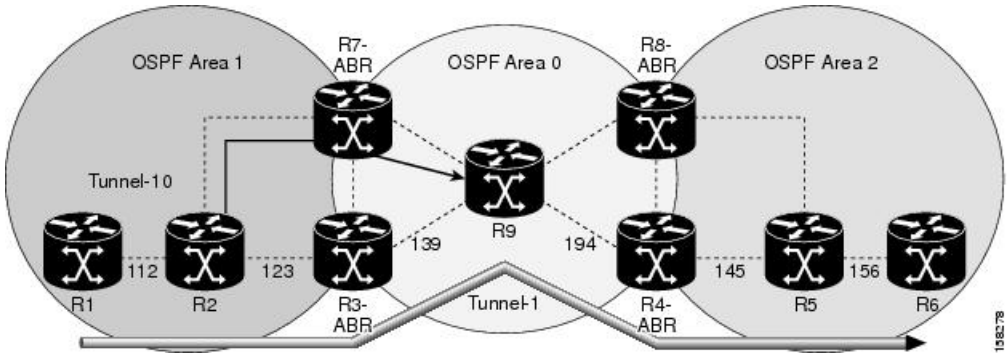
MPLS トラフィック エンジニアリングのエリア間トンネリング

MPLS-TE エリア間トンネリング機能を使用すると、複数の内部ゲートウェイ プロトコル（IGP）エリアとレベルにまたがる TE トンネルを確立できます。そのため、ヘッドエンドおよびテールエンド ルータが単一のエリアに存在するという要件がなくなります。

エリア間サポートでは、複数のエリアにまたがる TE LSP を設定できます。この場合、ヘッドエンドおよびテールエンド ラベル スイッチド ルータ（LSR）は異なる IGP エリア内にあります。（主に拡張性の理由から）複数の IGP エリア バックボーンを実行するお客様は、マルチエリアとエリア間 TE が必要です。これにより、フラッディングされる情報の量が制限され、SPF 期間が短くなり、特に複数のエリアで大きい WAN バックボーンが分割されているエリア内のリンクまたはノード障害の影響が少なくなります。

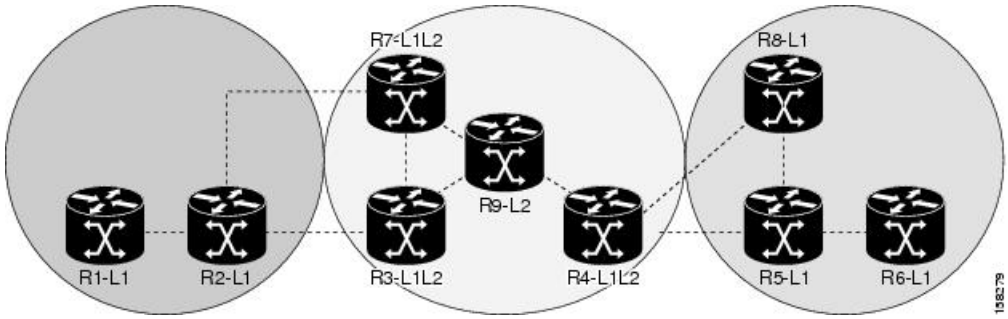
次の図は、OSPF を使用した典型的なエリア間 TE ネットワークを示しています。

図 *9*：エリア間（*OSPF*）*TE* ネットワークの図



次の図は、典型的なレベル間（IS-IS）TE ネットワークを示しています。

図 *10*：レベル間（*IS-IS*）*TE* ネットワーク構成図



「図 10：レベル間（IS-IS）TE ネットワーク構成図 （49 ページ）」に示されているように、R2、R3、R7、および R4 では、ルーティングと TE 情報に関する 2 つのデータベースが保持されます。たとえば、R3 では、R2 に関連する TE トポロジ情報はレベル 1 IS-IS LSP からフラッディングされ、R4、R9、および R7 に関連する TE トポロジ情報はレベル 2 IS-IS リンクステート PDU（LSP）（および独自の IS-IS LSP）からフラッディングされます。

ルーズ ホップ最適化を使用すると、複数のエリアにまたがるトンネルを再最適化でき、LSP のヘッドエンドの OSPF エリアおよび IS-IS レベル内にないホップを MPLS-TE LSP が通過させるときに発生する問題が解決されます。エリア内 MPLS-TE を使用すると、パス沿いにある ABR のルーズ ソース ルートを指定することで、エリア内トラフィック エンジニアリング（TE）ラベル スイッチド パス（LSP）を設定できます。その後、（ヘッドエンドに指定されている）ネクスト ホップ ABR に到達するために、次のエリア内で TE LSP 制約に従うパスを見つけるのは、（両方のエリアを全体的に把握する）ABR の責任になります。テールエンド LSR に到達するために、テールエンド エリアに接続されている最後の ABR によって同じ操作が実行されます。

ルーズ ホップ最適化を使用する場合は、次の考慮事項に注意する必要があります。

- （ABR のリンク アドレスとは対照的に）ABR ノードのルータ ID を指定する必要があります。

- サブエリアが含まれているマルチエリアをネットワーク内に配置する場合、ルーズホップの指定時にパスを見つけるために、TEのサブエリアでMPLS-TEをイネーブルにする必要があります。
- エリア間トンネルの到達可能な明示パスを指定する必要があります。

ポリシーベース トンネル選択

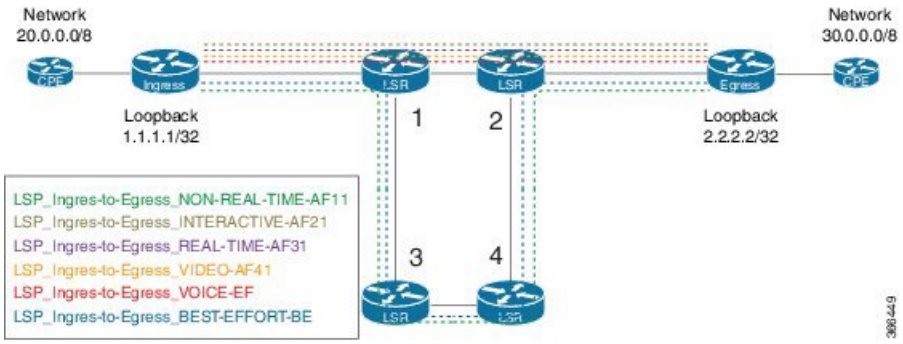
ポリシーベース トンネル選択（PBTS）は、さまざまな分類基準に基づいて特定の TE トンネルにトラフィックを転送できるメカニズムです。PBTS は、MPLS および MPLS/VPN ネットワーク経由で音声およびデータ トラフィックを伝送するインターネット サービス プロバイダー（ISP）や、最適化された音声サービスを提供するためにこのトラフィックをルーティングする必要がある ISP にとって役立ちます。

PBTS は、着信パケットの分類基準に基づいてトンネルを選択することで機能します。これらの基準は、パケットの IP プレシデンス、Differentiated Services Code Point（DSCP; DiffServ コードポイント）、またはタイプ オブ サービス（ToS）フィールドに基づいています。トラフィック転送の決定は、宛先ネットワークのみを考慮するのではなく、トラフィック クラスおよび宛先ネットワーク アドレスに基づいて行われます。

パスに設定されたデフォルトクラスは常にゼロ（0）です。所定の転送クラスに対してTEが存在しない場合は、デフォルトクラス（0）が試行されます。デフォルトクラスがない場合、パケットは設定された最も低い転送クラスのトンネルに対して試されます。PBTS は、1 つの TE トンネルに関連付けられている最大 7 までの EXP 値（exp 1 ～ 7）をサポートしています。

次の図に PBTS ネットワーク トポロジを示します。

図 *11* : ポリシーベース トンネル選択の実装



- トンネルは、LSR 1-2 および LSR 1-3-4-2 パスを介して入力ノードと出力ノード間に作成されます。
- 優先順位の高いトラフィックは、入力 -> LSR1 -> LSR2 -> 出力というパスをとります。
- 優先順位の低いトラフィックは、入力 -> LSR1 -> LSR3 -> LSR4 -> LSR2 -> 出力というパスをとります。

PBTS 機能の詳細

次の PBTS 機能がルータでサポートされます。

- PBR設定を使用してルールを作成することで、入力トラフィックをさまざまなクラスに分類する。
- IPv4 および IPv6 トラフィックの両方について DSCP/IP プレシデンスを使用してパケットを分類する。
- 分類後、目的の転送クラスを各タイプの入力トラフィックに設定する。
- トンネル設定を使用して、宛先に 1 つまたは複数の MPLS-TE トンネルを定義する。
- トンネル設定で MPLS-TE トンネルを特定の転送クラスに関連付ける。
- 設定された分類ルールを使用するサービスポリシーを適用して、入力インターフェイスで PBTS を有効にする。

次のリストは、PBTS のサポート情報を示しています。

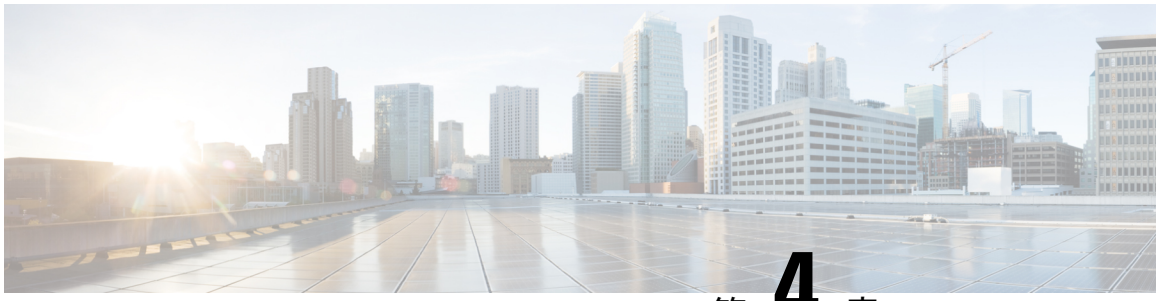
- PBTS は、Ipv4/Ipv6 着信トラフィックでのみサポートされます。
- 宛先プレフィックス 1 つあたり最大 8 つの転送クラスがサポートされます。
- 各転送クラス内に最大 64 の TE トンネルがサポートされます。
- 最大 64 の TE トンネルを所定の宛先に設定できます。
- 着信ラベル付きトラフィックはサポートされていません。
- L2VPN/L3VPN トラフィックによる PBTS はサポートされていません。

PBTS 転送クラス

クラスマップは様々なタイプのパケットに対して定義され、これらのクラスマップは転送クラスに関連付けられます。クラスマップは特定のタイプのトラフィックを分類するための一致基準を定義し、転送クラスはこれらのパケットが取るべき転送パスを定義します。

クラスマップがポリシーマップ内の転送クラスに関連付けられた後、クラスマップに一致するパケットはすべて、ポリシーマップで定義されたとおりに転送されます。パケットが各転送クラスに対して取るべき出力トラフィック エンジニアリング（TE）トンネル インターフェイスは、TE インターフェイスを明示的に（またはデフォルト値の場合は暗黙的に）転送グループに関連付けることによって指定されます。

TEインターフェイスが転送クラスに関連付けられている場合は、**auto-route** コマンドを使用してルーティングプロトコルモジュールにエクスポートできます。これにより、FIBデータベース内のルートがこれらのトンネルに関連付けられます。TE インターフェイスが転送クラスと明示的に関連付けられていない場合は、デフォルトクラス（0）に関連付けられます。すべての非TEインターフェイスは、ルーティングプロトコルによってフォワーディングプレーンにルーティングされます（転送クラスはデフォルトクラスに設定されます）。



第 4 章

MPLS-TE への RSVP の実装

リソース予約プロトコル（RSVP）は、システムによるネットワークからのリソース予約要求を可能にするシグナリング プロトコルです。RSVP は、他のシステムからのプロトコル メッセージを処理し、ローカル クライアントからのリソース要求を処理して、プロトコル メッセージを生成します。結果として、リソースは、ローカルおよびリモート クライアントの代わりにデータ フローに予約されます。RSVP は、これらのリソース予約を作成、保守および削除します。

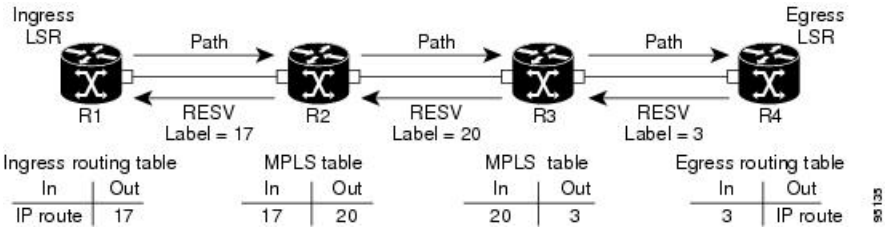
MPLS トラフィック エンジニアリング（MPLS-TE）はトポロジとネットワーク内で使用可能なリソースを学習し、帯域幅などのリソース要件とネットワーク リソースに基づいてトラフィック フローを特定のパスにマッピングします。MPLS TE では、ラベル スイッチド パス（LSP）の形でソースから宛先への単方向トンネルが構築され、その後トラフィックの転送で使用されます。MPLS-TE では、RSVP を使用して LSP をシグナリングします。

- RSVP を使用した MPLS LSP の設定 （53 ページ）
- MPLS-TE 用 RSVP の機能の概要 （54 ページ）
- MPLS-TE 用 RSVP の設定 （54 ページ）
- MPLS-TE 用 RSVP の機能の詳細 （61 ページ）

RSVP を使用した MPLS LSP の設定

次の図は、MPLS 環境で TE に使用できるルータ R4 を介してルータ R1 から LSP が RSVP によってどのように設定されるかを示しています。

図 *12 : RSVP* を使用した *MPLS LSP*



LSP の設定は LSP のヘッドノードが、テールノードにパスメッセージを送信すると開始されます。パスメッセージにより、各ノードへのパスに沿ってリソースが予約され、各ノードでセッ

ションに関連付けられたパスステートが作成されます。テール ノードがパス メッセージを受信すると、ラベル付きの予約（RESV）メッセージを直前のノードに戻します。各ルータの予約状態はソフト状態と見なされます。つまり、状態を維持するためには、各ホップで定期的なPATH メッセージと RESV メッセージを送信する必要があります。

予約メッセージが直前のノードに到着すると、予約されたリソースがロックされ、転送エントリが、テールエンドノードから送信されるMPLSラベルでプログラムされます。新しいMPLSラベルが割り当てられ、次のノードのアップストリームに送信されます。予約メッセージがヘッドノードに到着すると、ラベルがプログラムされ、MPLSデータがパスに送信されます。

MPLS-TE 用 RSVP の機能の概要

このセクションでは、MPLS-TE 用 RSVP のさまざまな機能の概要を示します。

RSVPは、MPLS-TEが設定されるインターフェイスで自動的にイネーブルにされます。帯域幅を持つ MPLS-TE LSP では、RSVP 帯域幅をインターフェイスで設定する必要があります。すべての MPLS-TE LSP がゼロ帯域幅の場合、RSVP を設定する必要はありません。

RSVP グレースフルリスタートは、高可用性を確保し、RSVP TE 対応ルータでネットワーク障害後にネイバーから RSVP 状態情報を回復できるようにします。

RSVP では、リフレッシュメッセージを定期的に送信することによって、LSP シグナリング時に設定されるパスと予約状態を更新する必要があります。リフレッシュメッセージは、RSVPネイバー間で状態を同期するため、および失われたRSVPメッセージから情報を回復するために使用されます。RSVP リフレッシュ削減機能では、メッセージが失われた場合に迅速に送信される信頼性の高いメッセージをサポートしています。サマリー リフレッシュ メッセージには、多数の状態を更新し、状態の更新に必要なメッセージの数を減らすための情報が含まれています。

RSVP メッセージを認証して、信頼できるネイバーのみが予約を設定できるようにすることができます。

MPLS-TE 用 RSVP の機能の詳細については、「MPLS-TE 用 RSVP の機能の詳細 （61 ページ）」を参照してください。

MPLS-TE 用 RSVP の設定

RSVP は、いくつかのルータでの調整が必要で、LSP を設定するため RSVP メッセージの交換を確立します。RSVPを設定するには、次の2つのRPMをインストールする必要があります。

- ncs540-mpls-2.0.0.0-r601.x86_64.rpm-6.0.1
- ncs540-mpls-te-rsvp-2.0.0.0-r601.x86_64.rpm-6.0.1

要件に応じて、RSVP では、次のトピックで説明するいくつかの基本設定が必要です。

RSVP メッセージ認証のグローバル設定

RSVP 認証機能により、RSVP ネットワークのネイバーは、安全なハッシュアルゴリズムを使用して、すべての RSVP シグナリングメッセージをデジタルで認証できます。この認証は、RSVP メッセージの RSVP インテグリティオブジェクトを使用して RSVP ホップごとに実行されます。インテグリティ オブジェクトには、キー ID、メッセージのシーケンス番号、およびキー付きメッセージ ダイジェストが含まれています。

キーチェーン、信頼できる他の RSVP ネイバーとのセキュリティアソシエーションを RSVP が保持する期間（ライフタイム）、順序が正しくなくても受信できる RSVP 認証済みメッセージの最大数（ウィンドウサイズ）など、認証パラメータの値をグローバルに設定できます。これらのデフォルトは、各ネイバーまたはインターフェイスで継承されます。

設定例

この例では、ルータで認証パラメータをグローバルに設定しています。認証キーチェーン、ライフタイム、ウィンドウサイズを含む認証パラメータを設定しています。このタスクを実行する前に、有効なキーチェーンを設定する必要があります。

```
RP/0/RP0/cpu 0: router# configure
RP/0/RP0/cpu 0: router(config)# key chain mpls-keys
RP/0/RP0/cpu 0: router(config-mpls-keys)# exit
RP/0/RP0/cpu 0: router(config)# rsvp authentication
RP/0/RP0/cpu 0: router(config-rsvp-auth)# key-source key-chain mpls-keys
RP/0/RP0/cpu 0: router(config-rsvp-auth)# life-time 2000
RP/0/RP0/cpu 0: router(config-rsvp-auth)# window-size 33
```

確認

次のコマンドを使用して、認証パラメータの設定を確認します。

```
RP/0/RP0/cpu 0: router# show rsvp authentication detail

  RSVP Authentication Information:
    Source Address:         3.0.0.1
    Destination Address:    3.0.0.2
    Neighbour Address:      3.0.0.2
    Interface:              HundredGigabitEthernet 0/0/0/3
    Direction:              Send
    LifeTime:               2000 (sec)
    LifeTime left:          1305 (sec)
    KeyType:                Static Global KeyChain
    Key Source:             mpls-keys
    Key Status:             No error
    KeyID:                  1
    Digest:                 HMAC MD5 (16)
    window-size:            33
Challenge:              Not supported
    TX Sequence:            5023969459702858020 (0x45b8b99b00000124)
    Messages successfully authenticated:    245
    Messages failed authentication:         0
```

関連項目

- インターフェイスでの RSVP 認証の設定 （56 ページ）
- ネイバーでの RSVP 認証の設定 （57 ページ）

- MPLS-TE 用 RSVP の機能の詳細 （61 ページ）

インターフェイスでの RSVP 認証の設定

インターフェイスで、キーチェーン、ライフタイム、ウィンドウサイズを含むRSVP認証パラメータの値を個別に設定できます。インターフェイス固有の認証パラメータは、2 つの RSVP ネイバー間で特定のインターフェイスのセキュリティを確保するために使用します。

設定例

この例では、インターフェイスで、認証キーチェーン、セキュリティアソシエーションのライフタイム、およびウィンドウサイズを設定しています。有効なキーチェーンが、このタスクの一部として使用できるようにすでに設定されている必要があります。

```
RP/0/RP0/cpu 0: router# configure
RP/0/RP0/cpu 0: router(config)# rsvp interface HundredGigabitEthernet0/0/0/3
RP/0/RP0/cpu 0: router(config-rsvp-if)# authentication
RP/0/RP0/cpu 0: router(config-rsvp-if-auth)# key-source key-chain mpls-keys
RP/0/RP0/cpu 0: router(config-rsvp-if-auth)# life-time 2000
RP/0/RP0/cpu 0: router(config-rsvp-if-auth)# window-size 33
RP/0/RP0/cpu 0: router(config)# commit
```

確認

次のコマンドを使用して、認証パラメータの設定を確認します。

```
RP/0/RP0/cpu 0: router# show rsvp authentication detail

  RSVP Authentication Information:
    Source Address:          3.0.0.1
    Destination Address:     3.0.0.2
    Neighbour Address:       3.0.0.2
    Interface:               HundredGigabitEthernet 0/0/0/3
    Direction:               Send
    LifeTime:                2000 (sec)
    LifeTime left:           1305 (sec)
    KeyType:                 Static Global KeyChain
    Key Source:              mpls-keys
    Key Status:              No error
    KeyID:                   1
    Digest:                  HMAC MD5 (16)
    window-size:             33
    Challenge:               Not supported
    TX Sequence:             5023969459702858020 (0x45b8b99b00000124)
    Messages successfully authenticated:    245
    Messages failed authentication:         0
```

関連項目

- RSVP メッセージ認証のグローバル設定 （55 ページ）
- ネイバーでの RSVP 認証の設定 （57 ページ）
- MPLS-TE 用 RSVP の機能の詳細 （61 ページ）

ネイバーでの RSVP 認証の設定

ネイバーで、キーチェーン、ライフタイム、ウィンドウサイズを含むRSVP認証パラメータの値を個別に設定できます。

設定例

この例では、RSVPネイバーで、認証キーチェーン、セキュリティアソシエーションのライフタイム、およびウィンドウサイズを設定しています。有効なキーチェーンが、このタスクの一部として使用できるようにすでに設定されている必要があります。

```
RP/0/RP0/cpu 0: router# configure
RP/0/RP0/cpu 0: router(config)# rsvp neighbor 1.1.1.1 authentication
RP/0/RP0/cpu 0: router(config-rsvp-if-auth)# key-source key-chain mpls-keys
RP/0/RP0/cpu 0: router(config-rsvp-if-auth)# life-time 2000
RP/0/RP0/cpu 0: router(config-rsvp-if-auth)# window-size 33
RP/0/RP0/cpu 0: router(config)# commit
```

確認

次のコマンドを使用して、認証パラメータの設定を確認します。

```
RP/0/RP0/cpu 0: router# show rsvp authentication detail

  RSVP Authentication Information:
    Neighbour Address:      1.1.1.1
    Interface:              HundredGigabitEthernet 0/0/0/3
    Direction:              Send
    LifeTime:               2000 (sec)
    LifeTime left:          1205 (sec)
    KeyType:                Static Global KeyChain
    Key Source:             mpls-keys
    Key Status:             No error
    KeyID:                  1
    Digest:                 HMAC MD5 (16)
    window-size:            33
    Challenge:              Not supported
```

関連項目

- RSVP メッセージ認証のグローバル設定 （55 ページ）
- インターフェイスでの RSVP 認証の設定 （56 ページ）
- MPLS-TE 用 RSVP の機能の詳細 （61 ページ）

グレースフル リスタートの設定

RSVPグレースフルリスタートは、高可用性（HA）を確保するためのメカニズムを提供して、Cisco IOS XR ソフトウェアを実行するシステムで障害状態を検出および回復できるようにし、ノンストップ フォワーディング サービスを実現します。RSVP グレースフルリスタートは、RSVPhelloメッセージに基づいており、RSVPTE対応ルータでネットワーク障害後にネイバーから RSVP 状態情報を回復できるようにします。RSVP では、hello メッセージ内の Restart Cap オブジェクト（RSVPRESTART）を使用して、ノードの再起動機能をアドバタイズするために

再起動時間と回復時間を指定します。ネイバーノードは、再起動ノードのフォワーディングステートを回復するための Recover Label オブジェクトを送信することで、再起動ノードを支援します。

ノード ID アドレスベースの hello メッセージに基づく標準グレースフルリスタートを設定することも、インターフェイスアドレス ベースの hello メッセージに基づくインターフェイスベースのグレースフルリスタートを設定することもできます。

設定例

この例では、ネットワーク上のルータノードで RSVP-TE がすでに有効になっており、障害から回復できるようにルータノードでグレースフルリスタートが有効になっている必要があります。グレースフルリスタートを、有効なノード ID アドレスベースの hello メッセージに対してグローバルに設定し、インターフェイスアドレス ベースの hello メッセージをサポートするためにルータインターフェイスで設定します。

```
RP/0/RP0/cpu 0: router# configure
RP/0/RP0/cpu 0: router(config)# rsvp
RP/0/RP0/cpu 0: router(config-rsvp)# signalling graceful-restart
RP/0/RP0/cpu 0: router(config-rsvp)# interface HundredGigabitEthernet 0/0/0/3
RP/0/RP0/cpu 0: router(config-rsvp-if)# signalling graceful-restart
interface-based
RP/0/RP0/cpu 0: router(config)# commit
```

確認

次のコマンドを使用して、グレースフルリスタートが有効になっていることを確認します。

```
RP/0/RP0/cpu 0: router# show rsvp graceful-restart
 Graceful restart: enabled Number of global neighbors: 1
 Local MPLS router id: 192.168.55.55
 Restart time: 60 seconds  Recovery time: 120 seconds
 Recovery timer: Not running
 Hello interval: 5000 milliseconds Maximum Hello miss-count: 4

RP/0/RP0/cpu 0: router# show rsvp graceful-restart neighbors detail

 Neighbor: 192.168.77.77 Source: 192.168.55.55 (MPLS)
   Hello instance for application MPLS
     Hello State: UP        (for 00:20:52)
     Number of times communications with neighbor lost: 0
     Reason: N/A
Recovery State: DONE
   Number of Interface neighbors: 1
     address: 192.168.55.0
   Restart time: 120 seconds  Recovery time: 120 seconds
   Restart timer: Not running
   Recovery timer: Not running
   Hello interval: 5000 milliseconds  Maximum allowed missed Hello messages: 4
```

関連項目

- MPLS-TE 用 RSVP の機能の詳細 （61 ページ）

リフレッシュ削減の設定

RSVP リフレッシュ削減は、デフォルトで有効になっており、Resource Reservation Protocol（RSVP）シグナリングの信頼性を高めてネットワークのパフォーマンスとメッセージ配信の信頼性を向上します。リフレッシュ削減は、ネイバーでサポートされている場合に限り、ネイバーで使用されます。必要に応じて、インターフェイスでリフレッシュ削減を無効にすることもできます。

設定例

この例では、リフレッシュ削減機能で使用できるさまざまなパラメータを設定する方法を示します。

次のパラメータを設定して、そのデフォルト値を変更します。

- 更新間隔
- ノードで許容される、失われたリフレッシュメッセージの数
- 再送信時間
- 確認応答保持時間
- 確認応答メッセージのサイズ
- サマリー リフレッシュ メッセージのサイズ

```
RP/0/RP0/cpu 0: router# configure
RP/0/RP0/cpu 0: router(config)# rsvp
RP/0/RP0/cpu 0: router(config-rsvp)# interface HundredGigabitEthernet 0/0/0/3
RP/0/RP0/cpu 0: router(config-rsvp-if)# signalling refresh interval 40
RP/0/RP0/cpu 0: router(config-rsvp-if)# signalling refresh missed 6
RP/0/RP0/cpu 0: router(config-rsvp-if)# signalling refresh reduction reliable
retransmit-time 2000
RP/0/RP0/cpu 0: router(config-rsvp-if)# signalling refresh reduction reliable ack-hold-time
 1000
RP/0/RP0/cpu 0: router(config-rsvp-if)# signalling refresh reduction reliable ack-max-size
 1000
RP/0/RP0/cpu 0: router(config-rsvp-if)# signalling refresh reduction summary  max-size
1500
RP/0/RP0/cpu 0: router(config)# commit
```

ACL ベース プレフィックス フィルタリングの設定

拡張アクセス制御リスト（ACL）を設定して、RSVP ルータ アラート（RA）パケットで通常の処理を転送、ドロップ、または実行できます。各着信 RSVP RA パケットについて、RSVP では、IP ヘッダーを検査し、送信元 IP アドレスまたは宛先 IP アドレスと拡張 ACL で設定されたプレフィックスとの照合を行います。明示的な許可も明示的な拒否もない場合、デフォルトでは、ACL インフラストラクチャは暗黙的な拒否を返します。デフォルトでは、ACL 一致により暗黙的な（デフォルト）拒否が返された場合、RSVP によりパケットが処理されます。

設定例

この例では、RSVP RA パケットに対する ACL ベースのプレフィックス フィルタリングを設定します。RSVP で送信元アドレス 1.1.1.1 から RA パケットを受信した場合はそのパケットは転送され、IP アドレス 2.2.2.2 宛てのパケットはドロップされます。

```
RP/0/RP0/cpu 0: router# configure
RP/0/RP0/cpu 0: router(config)# ipv4 access-list rsvpac1
RP/0/RP0/cpu 0: router(config-ipv4-acl)# 10 permit ip host 1.1.1.1 any
RP/0/RP0/cpu 0: router(config-ipv4-acl)# 20 deny ip any host 2.2.2.2
RP/0/RP0/cpu 0: router(config)# rsvp
RP/0/RP0/cpu 0: router(config-rsvp)# signalling prefix-filtering access-list rsvp-acl
RP/0/RP0/cpu 0: router(config)# commit
```

確認

ACL ベース プレフィックス フィルタリングの設定の確認

```
RP/0/RP0/cpu 0: router# show rsvp counters prefix-filtering access-list rsvp-ac1

  ACL:rsvp-ac1            Forward          Local          Drop           Total
  Path                      0              0              0              0
  PathTear                  0              0              0              0
  ResvConfirm               0              0              0              0
  Total                     0              0              0
```

関連項目

- RSVP パケット ドロップの設定 （60 ページ）

RSVP パケット ドロップの設定

拡張アクセス制御リスト（ACL）を設定して、RSVP ルータ アラート（RA）パケットで通常の処理を転送、ドロップ、または実行できます。デフォルトでは、ACL との照合で暗黙的な拒否が返された場合でも、RSVP はその RA パケットを処理します。ACL との照合で暗黙的な拒否になった場合には RA パケットをドロップするように、RSVP を設定することもできます。

設定例

この例では、RSVP RA パケットに対する ACL ベースのプレフィックス フィルタリングを設定します。RSVP で送信元アドレス 1.1.1.1 から RA パケットを受信した場合はそのパケットは転送され、IP アドレス 2.2.2.2 宛てのパケットはドロップされます。ACL との照合で暗黙的な拒否になった場合には、RA パケットはドロップされます。

```
RP/0/RP0/cpu 0: router# configure
RP/0/RP0/cpu 0: router(config)# ipv4 access-list rsvpac1
RP/0/RP0/cpu 0: router(config-ipv4-acl)# 10 permit ip host 1.1.1.1 any
RP/0/RP0/cpu 0: router(config-ipv4-acl)# 20 deny ip any host 2.2.2.2
RP/0/RP0/cpu 0: router(config)# rsvp
RP/0/RP0/cpu 0: router(config-rsvp)# signalling prefix-filtering default-deny-action
RP/0/RP0/cpu 0: router(config)# commit
```

確認

次のコマンドを使用して、RSVP パケットドロップの設定を確認します。

```
RP/0/RP0/cpu 0: router# show rsvp counters prefix-filtering access-list rsvpac1

  ACL: rsvpac1            Forward          Local          Drop           Total
  Path                      4              1              0              5
  PathTear                  0              0              0              0
  ResvConfirm               0              0              0              0
  Total                     4              1              0              5
```

関連資料

- ACL ベース プレフィックス フィルタリングの設定 （59 ページ）

RSVP トラップの有効化

RSVP MIB を実装することで、SNMP を使用して、RSVP に属するオブジェクトにアクセスできます。また、新しいフローの作成または削除時にトリガーされる 2 つのトラップ（NewFlow と LostFlow）を指定できます。RSVP MIB は RSVP をオンにすると自動的に有効になりますが、RSVP トラップは別に有効にする必要があります。

設定例

この例は、フローが削除または作成された場合の RSVP MIB トラップを有効にする方法に加えて、両方のトラップを有効にする方法も示しています。

```
RP/0/RP0/cpu 0: router# configure
RP/0/RP0/cpu 0: router(config)# snmp-server traps rsvp lost-flow
RP/0/RP0/cpu 0: router(config)# snmp-server traps rsvp new-flow
RP/0/RP0/cpu 0: router(config)# snmp-server traps rsvp all
RP/0/RP0/cpu 0: router(config)# commit
```

MPLS-TE 用 RSVP の機能の詳細

RSVP グレースフルリスタートの動作

RSVP グレースフルリスタートは、RSVP hello メッセージに基づきます。hello メッセージは、ルータとそのネイバーノードの間で交換されます。各ネイバーノードは、hello 要求オブジェクトを含む hello メッセージを自律して発行できます。hello 拡張をサポートするレシーバは、hello 確認（ACK）オブジェクトを含む hello メッセージで応答します。送信側ノードが状態の回復をサポートしている場合、ノードの再起動機能を示す Restart Cap オブジェクトも hello メッセージで伝送されます。Restart Cap オブジェクトでは、再起動時間と回復時間が指定されています。再起動時間は、hello メッセージが失われてから RSVP hello セッションを再確立するまでの時間です。回復時間は、hello メッセージの再確立後に受信者が状態を再同期するまで送信側が待機する時間です。

グレースフル リスタートでは、hello メッセージは、64 の IP Time to Live（TTL）で送信されます。これは、hello メッセージの宛先が数ホップ離れることがあるためです。グレースフル リ

スタートがイネーブルで、RSVP ステートがネイバーと共有される場合、hello メッセージ（Restart Cap オブジェクトを含む）は RSVP ネイバーに送信されます。Restart Cap オブジェクトが RSVP ネイバーに送信される場合に、ネイバーが Restart Cap オブジェクトを含む hello メッセージで応答すると、そのネイバーはグレースフルリスタート可能と見なされます。ネイバーが hello メッセージに応答しない場合、または Restart Cap オブジェクトを含まない hello メッセージに応答した場合、RSVP は、そのネイバーへの hello の送信をバックオフします。hello Request メッセージが不明ネイバーから受信された場合、hello ACK は返されません。

RSVP 認証

ネットワーク管理者は、RSVP 要求を開始するシステムのセットを制御するセキュリティ ドメインを確立できる機能が必要です。RSVP 認証機能を使用すると、RSVP ネットワークのネイバーは、安全なハッシュを使用して、すべての RSVP シグナリング メッセージにデジタル署名できます。これにより、RSVP メッセージの受信側は、送信側の IP アドレスだけに頼ることなく、メッセージの送信側を確認できます。

署名は、RFC 2747 で定義されている RSVP メッセージの RSVP インテグリティ オブジェクトで RSVP ホップごとに実行されます。インテグリティ オブジェクトには、キー ID、メッセージのシーケンス番号、およびキー付きメッセージ ダイジェストが含まれています。この方式では、偽造やメッセージ改ざんに対する保護が提供されます。ただし、受信側で、受信した RSVP メッセージ内のデジタル署名を確認するためには、送信側で使用されたセキュリティ キーを取得する必要があります。ネットワーク管理者は、共有ネットワークの各 RSVP ネイバーで共有のキーを手動で設定します。送信側システムと受信側システムでは、共有する各認証キーのセキュリティ アソシエーションが維持されます。さまざまなセキュリティ アソシエーション パラメータの詳細については、次を参照してください。表 2 : セキュリティ アソシエーションのパラメータ （63 ページ）

キー、ウィンドウ サイズおよびライフタイムを含むすべての認証パラメータに対してグローバル デフォルトを設定できます。これらのデフォルトは、各ネイバーまたはインターフェイスで認証を設定するときに継承されます。ただし、これらのパラメータはネイバーまたはインターフェイスで個別で設定できますが、この場合はグローバル値（設定値またはデフォルト値）は継承されません。

インターフェイス モードおよびネイバー インターフェイス モードは、明示的に設定されていない限り、次のように、グローバル コンフィギュレーション モードからパラメータを継承します。

- ウィンドウ サイズは、1 に設定されます。
- 制限は 1800 に設定されます。
- key-source key-chain コマンドは、none またはディセーブルに設定されます。

次に、グローバル、インターフェイス、またはネイバー コンフィギュレーション モードの選択方法を示します。

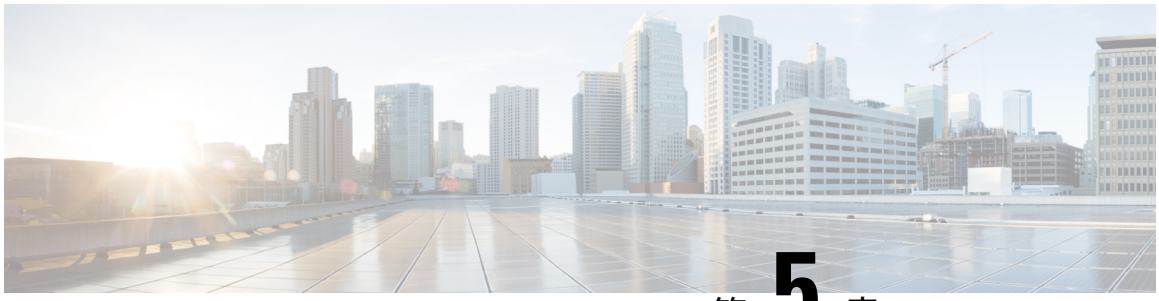
- グローバル コンフィギュレーション モードは、ルータが単一のセキュリティ ドメインに属する場合に最適です（たとえば、プロバイダー コア ルータのセットの一部などです）。単一の共有キー セットは、すべての RSVP メッセージの認証に使用されます。

- インターフェイスまたはネイバー コンフィギュレーション モードは、ルータが複数のセキュリティ ドメインに属する場合に最適です。たとえば、プロバイダー ルータが、プロバイダーエッジ（PE）に隣接する場合や、PEがエッジデバイスに隣接する場合です。異なるキーを使用できますが、共有はできません。

セキュリティ アソシエーション（SA）は、ピアとの安全な通信を維持するために必要な情報のコレクションです。次の表に、セキュリティアソシエーションを定義する主要パラメータを示します。

表 *2:* セキュリティ アソシエーションのパラメータ

セキュリティ アソシエーションのパラメータ	説明
src	送信元の IP アドレス。
dst	最終的な宛先の IP アドレス。
interface	セキュリティアソシエーションのインターフェイス。
direction	セキュリティ アソシエーションの送信または受信タイプ。
Lifetime	未使用のセキュリティアソシエーションデータの収集に使用される有効期限タイマーの値。
Sequence Number	送信または受信（direction のタイプ）された最後のシーケンス番号。
key-source	設定可能パラメータのキーのソース。
keyID	最後に使用されたキー番号（key-source から返されます）。
Window Size	順序どおりでなくても受信できる認証済みメッセージの最大数を指定します。
Window	受信または受け入れられた最後の *window size* 値のシーケンス番号を指定します。



第 5 章

MPLS OAM の実装

MPLS保守運用管理（OAM）は、MPLSネットワークの障害検出とトラブルシューティングに役立ちます。これを使用することで、サービスプロバイダーはラベルスイッチドパス（LSP）をモニタして MPLS フォワーディングの問題を迅速に隔離できます。このモジュールでは、MPLSネットワークの障害検出とトラブルシューティングに使用できる MPLS LSP の ping 機能と traceroute 機能について説明します。

- MPLS LSP ping （65 ページ）
- MPLS LSP traceroute （67 ページ）

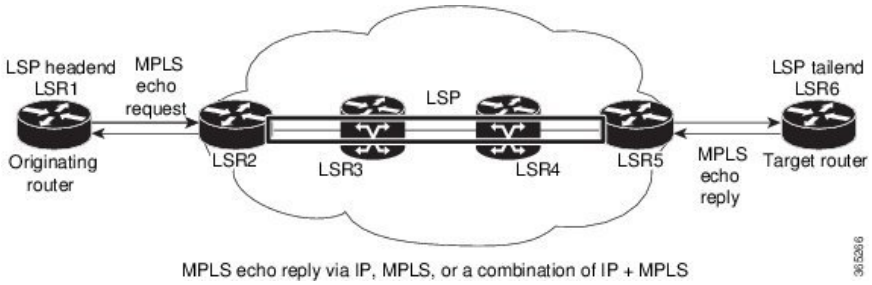
MPLS LSP ping

MPLS LSP ping 機能を使用して、LSP の入力 LSR と出力 LSR 間の接続を確認します。MPLS LSP ping は、Internet Control Message Protocol（ICMP）のエコー要求メッセージと応答メッセージと同様に、LSP の検証に MPLS エコーの要求メッセージと応答メッセージを使用します。ICMP のエコー要求メッセージと応答メッセージが IP ネットワークを検証するのに対し、MPLS エコー メッセージと応答メッセージは MPLS ネットワークを検証します。MPLS エコー要求パケットは、検証対象の LSP に関連付けられた適切なラベル スタックを使用してターゲット ルータに送信されます。ラベル スタックを使用すると、パケットは LSP 自体を介して転送されます。MPLS エコー要求パケットの宛先 IP アドレスは、ラベル スタックの選択に使用されるアドレスとは異なります。宛先 IP アドレスは 127.x.y.z/8 アドレスとして定義され、LSP が壊れている場合は IP パケットがそれ自体の宛先へ IP を切り替えないようにします。

MPLS エコー応答は、MPLS エコー要求に応じて送信されます。応答は IP パケットとして送信され、IP、MPLS、または両方のスイッチング タイプの組み合わせを使用して転送されます。MPLS エコー応答パケットの送信元アドレスは、エコー応答を生成するルータから取得されたアドレスです。宛先アドレスは、MPLS エコー要求パケットを送信したルータの送信元アドレスです。MPLS エコー応答の宛先ポートは、エコー要求の送信元ポートに設定されます。

次に、MPLS LSP ping のエコー要求とエコー応答のパスの図を示します。

図 *13 : MPLS LSP ping* のエコー要求と応答のパス



設定例

次に、MPLS LSP ping を使用して IPv4 LDP LSP の接続をテストする例を示します。宛先は Label Distribution Protocol（LDP）の IPv4 プレフィックスとして指定し、転送等価クラス（FEC）タイプは generic（汎用）と指定します。

```
RP/0/RP0/cpu 0: router# ping mpls ipv4 10.1.1.2/32 fec-type generic

Wed Nov 25 03:36:33.143 UTC
Sending 5, 100-byte MPLS Echos to 10.1.1.2/32,
      timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.

!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/3 ms
```

次に、MPLS LSP ping を使用して、宛先を MPLS のトラフィック エンジニアリング（TE）トンネルとして指定した場合に接続をテストする例を示します。

```
RP/0/RP0/cpu 0: router# ping mpls traffic-eng tunnel-te 4003 source 10.1.1.2

Tue Nov 24 20:39:39.179 PST

Sending 5, 100-byte MPLS Echos to tunnel-te4003,
      timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.

!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 3/3/4 ms
```

次に、**show mpls oam** コマンドを使用して MPLS OAM 情報を表示する例を示します。

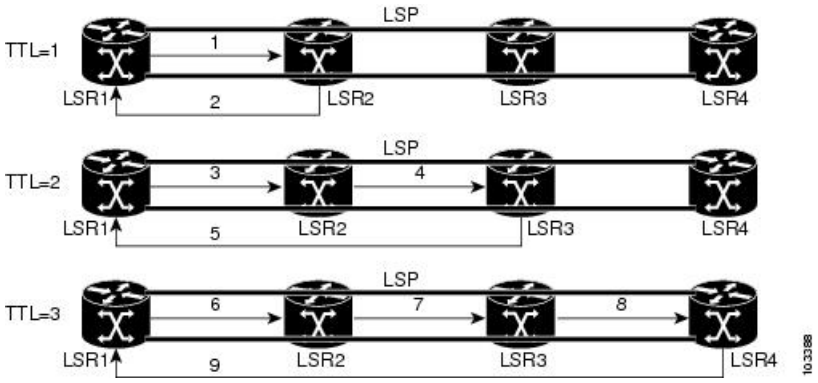
```
RP/0/RP0/cpu 0: router# show mpls oam counters packet

Wed Nov 25 03:38:07.397 UTC Global Packet Statistics:

                       Pkt                 Bytes
                       ---------------     ---------------
Receive Counts:
     Good Requests:                    0                   0
      Good Replies:                   10                 760
 Unknown Pkt Types:                    0                   0
   IP header error:                    0                   0
  UDP header error:                    0                   0
            Runts:                     0                   0
  Dropped (Q full):                    0                   0
     General error:                    0                   0
      Error, no IF:                    0                   0
  Error, no memory:                    0                   0

Transmit Counts:
              Good:                   10                 960
           Dropped:                    0                   0
```

MPLS LSP traceroute

MPLS LSP traceroute 機能を使用して、LSP の障害ポイントを隔離します。これはホップバイホップ エラーのローカリゼーションとパス トレースに使用されます。MPLS LSP traceroute 機能は、エコー要求を伝送するパケットの存続可能時間（TTL）値の期限切れに依存します。MPLSエコー要求メッセージが中継ノードを見つけるとTTL値をチェックし、期限が切れている場合はコントロールプレーンにパケットが渡されます。それ以外の場合は、メッセージが転送されます。エコー メッセージがコントロール プレーンに渡されると、要求メッセージの内容に基づいて応答メッセージが生成されます。

次の図に、LSR1 から LSR4 までの LSP の MPLS LSP traceroute の例を示します。

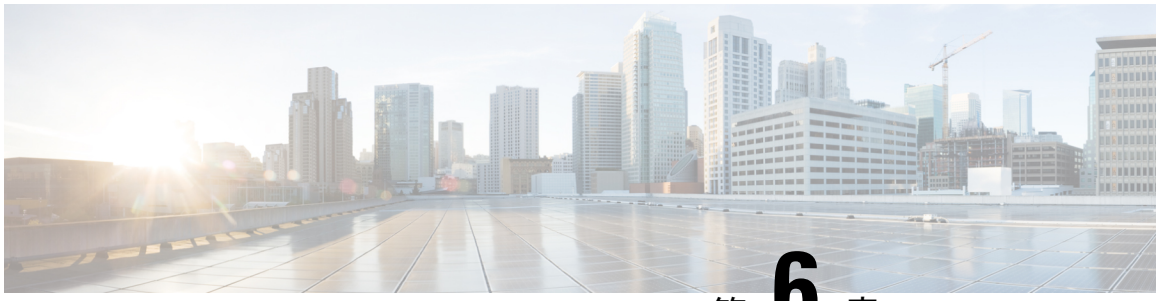
図 *14 : MPLS LSP traceroute*



設定例

次に、traceroute コマンドを使用して、generic として指定した転送等価クラスの宛先をトレースする例を示します。

```
RP/0/RP0/cpu 0: router# traceroute mpls ipv4 3.3.3.3/32 fec-type generic
Mon Nov 30 17:48:45.585 UTC

Tracing MPLS Label Switched Path to 3.3.3.3/32, timeout is 2 seconds

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.

  0 11.1.1.57 MRU 1500 [Labels: implicit-null Exp: 0]
! 1 11.1.1.58 7 ms23:19
```



第 6 章

グローバル重み付け SRLG 保護の設定

共有リスクリンクグループ（SRLG）は、共通のリソースを共有する一連のリンクであり、同じ障害リスクを共有します。内部ゲートウェイ プロトコル（IGP）における既存のループ フリー代替（LFA）の実装では、SRLG保護がサポートされています。ただし、既存の実装では、バックアップパスの計算中に直接接続されたリンクのみが考慮されます。したがって、直接接続されていないものの同じ SRLG を共有しているリンクが、バックアップ パスの計算中に追加された場合、SRLG 保護が失敗することがあります。グローバル重み付け SRLG 保護機能は、SRLG 値に重みを関連付けて、バックアップパスの計算時に SRLG 値の重みを使用することにより、SRLG のパス選択を向上させることができます。

グローバル重み付け SRLG 保護をサポートするには、エリア トポロジ内のすべてのリンクで SRLG に関する情報が必要です。ISIS を使用してリモート リンクの SRLG をフラッディングすることも、リモート リンクで SRLGS を手動で設定することもできます。

設定例：グローバル重み付け SRLG 保護

グローバル重み付け SRLG 保護機能では 3 種類の設定がサポートされています。

- グローバル重み付け SRLG 保護を使用したローカル SRLG
- リモート SRLG フラッディング
- リモート SRLG スタティック プロビジョニング

次に、グローバル重み付け SRLG 保護機能を使用してローカル SRLG を設定する例を示します。

```
RP/0/RP0/CPU0:router(config)# srlg
RP/0/RP0/CPU0:router(config-srlg)# interface TenGigE0/0/0/0
RP/0/RP0/CPU0:router(config-srlg-if)# name group1
RP/0/RP0/CPU0:router(config-srlg-if)# exit
RP/0/RP0/CPU0:router(config-srlg)# interface TenGigE0/0/0/1
RP/0/RP0/CPU0:router(config-srlg-if)# name group1
RP/0/RP0/CPU0:router(config-srlg)# name group value 100
RP/0/RP0/CPU0:router(config)# router isis 1
RP/0/RP0/CPU0:router(config-isis)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-isis-if-af)# fast-reroute per-prefix srlg-protection
weighted-global
RP/0/RP0/CPU0:router(config-isis-if-af)# fast-reroute per-prefix tiebreaker srlg-disjoint
 index 1
```

```
RP/0/RP0/CPU0:router(config-isis)# interface TenGigE0/0/0/0
RP/0/RP0/CPU0:router(config-isis-if)# point-to-point
RP/0/RP0/CPU0:router(config-isis-if)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-isis-if-af)# fast-reroute per-prefix
RP/0/RP0/CPU0:router(config-isis-if-af)# fast-reroute per-prefix ti-lfa
RP/0/RP0/CPU0:router(config-isis)# srlg
RP/0/RP0/CPU0:router(config-isis-srlg)# name group1
RP/0/RP0/CPU0:router(config-isis-srlg-name)# admin-weight 5000
```

次に、リモート SRLG フラッディングを使用してグローバル重み付け SRLG 保護機能を設定する例を示します。この設定には、ローカルおよびリモート ルータの設定が含まれています。ローカル ルータでは、**fast-reroute per-prefix srlg-protection weighted-global** コマンドを使用してグローバル重み付け SRLG 保護を有効にします。リモート ルータの設定では、**advertise application lfa link-attributes srlg** コマンドを使用して、SRLG 値のフラッディングを制御できます。また、リモート ルータで SRLG をグローバルに設定する必要もあります。

リモート SRLG フラッディングを使用したグローバル重み付け SRLG 保護のローカル ルータ設定は、次のとおりです。

```
RP/0/RP0/CPU0:router(config)# router isis 1
RP/0/RP0/CPU0:router(config-isis)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-isis-if-af)# fast-reroute per-prefix srlg-protection
weighted-global
RP/0/RP0/CPU0:router(config-isis-if-af)# fast-reroute per-prefix tiebreaker srlg-disjoint
 index 1
RP/0/RP0/CPU0:router(config-isis-if-af)# exit
RP/0/RP0/CPU0:router(config-isis)# interface TenGigE0/0/0/0
RP/0/RP0/CPU0:router(config-isis-if)# point-to-point
RP/0/RP0/CPU0:router(config-isis-if)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-isis-if-af)# fast-reroute per-prefix
RP/0/RP0/CPU0:router(config-isis-if-af)# fast-reroute per-prefix ti-lfa
RP/0/RP0/CPU0:router(config-isis-if-af)# exit
RP/0/RP0/CPU0:router(config-isis)# srlg
RP/0/RP0/CPU0:router(config-isis-srlg)# name group1
RP/0/RP0/CPU0:router(config-isis-srlg-name)# admin-weight 5000
```

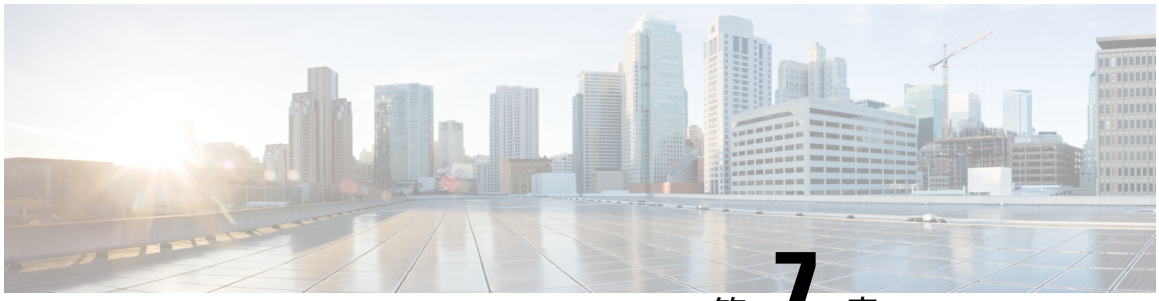
リモート SRLG フラッディングを使用したグローバル重み付け SRLG 保護のリモート ルータ設定は、次のとおりです。

```
RP/0/RP0/CPU0:router(config)# srlg
RP/0/RP0/CPU0:router(config-srlg)# interface TenGigE0/0/0/0
RP/0/RP0/CPU0:router(config-srlg-if)# name group1
RP/0/RP0/CPU0:router(config-srlg-if)# exit
RP/0/RP0/CPU0:router(config-srlg)# interface TenGigE0/0/0/1
RP/0/RP0/CPU0:router(config-srlg-if)# name group1
RP/0/RP0/CPU0:router(config-srlg)# name group value 100
RP/0/RP0/CPU0:router(config-srlg)# exit
RP/0/RP0/CPU0:router(config)# router isis 1
RP/0/RP0/CPU0:(config-isis)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-isis-af)# advertise application lfa link-attributes srlg
```

次に、リモート リンクの SRLG 値のスタティックプロビジョニングを使用したグローバル重み付け SRLG 保護機能の設定例を示します。これらの設定はローカル ルータで行う必要があります。

```
RP/0/RP0/CPU0:router(config)# srlg
RP/0/RP0/CPU0:router(config-srlg)# interface TenGigE0/0/0/0
RP/0/RP0/CPU0:router(config-srlg-if)# name group1
```

```
RP/0/RP0/CPU0:router(config-srlg-if)# exit
RP/0/RP0/CPU0:router(config-srlg)# interface TenGigE0/0/0/1
RP/0/RP0/CPU0:router(config-srlg-if)# name group1
RP/0/RP0/CPU0:router(config-srlg)# name group value 100
RP/0/RP0/CPU0:router(config-srlg)# exit
RP/0/RP0/CPU0:router(config)# router isis 1
RP/0/RP0/CPU0:router(config-isis)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-isis-if-af)# fast-reroute per-prefix srlg-protection
weighted-global
RP/0/RP0/CPU0:router(config-isis-if-af)# fast-reroute per-prefix tiebreaker srlg-disjoint
 index 1
RP/0/RP0/CPU0:router(config-isis)# interface TenGigE0/0/0/0
RP/0/RP0/CPU0:router(config-isis-if)# point-to-point
RP/0/RP0/CPU0:router(config-isis-if)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-isis-if-af)# fast-reroute per-prefix
RP/0/RP0/CPU0:router(config-isis-if-af)# fast-reroute per-prefix ti-lfa
RP/0/RP0/CPU0:router(config-isis)# srlg
RP/0/RP0/CPU0:router(config-isis-srlg)# name group1
RP/0/RP0/CPU0:router(config-isis-srlg-name)# admin-weight 5000
RP/0/RP0/CPU0:router(config-isis-srlg-name)# static ipv4 address 10.0.4.1 next-hop ipv4
 address 10.0.4.2
RP/0/RP0/CPU0:router(config-isis-srlg-name)# static ipv4 address 10.0.4.2 next-hop ipv4
 address 10.0.4.1
```



第 7 章

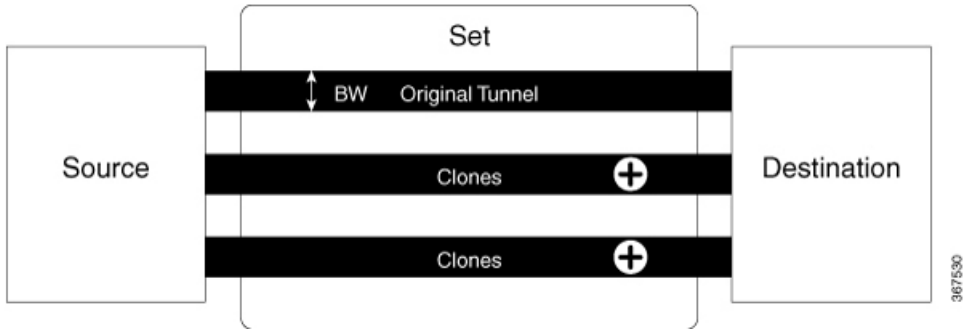
自動帯域幅バンドル TE++ の設定

MPLS-TEトンネルは、ラベル付きの接続を設定し、エンドポイント間の動的な帯域幅容量を提供するために使用されます。自動帯域幅機能は帯域幅容量の動的な要求に対応するもので、測定されたトラフィック負荷に基づいて MPLS-TE トンネルのサイズを動的に変更します。ただし、多くのお客様は、ロードバランシングと冗長性を目的として、エンドポイント間での複数の自動帯域幅トンネルを必要としています。2 つのエンドポイント間で集約帯域幅の要求が増加した場合は、自動帯域幅機能を設定してトンネルのサイズを変更するか、または新しいトンネルを作成して、2つのエンドポイント間のすべてのトンネルにわたって要求全体のロードバランシングを図ることができます。同様に、2 つのエンドポイント間で集約帯域幅の要求が減少した場合は、自動帯域幅機能を設定してトンネルのサイズを縮小するか、または新しいトンネルを削除して、エンドポイント間の残りのトンネルにわたってトラフィックのロードバランシングを図ることができます。自動帯域幅バンドル TE++ 機能は、自動帯域幅機能を拡張したものであり、リアルタイムでのトラフィックのニーズに基づいて、宛先への MPLS-TE トンネルの数を自動的に増減することができます。

帯域幅の要求の増加に対する応答として自動的に作成されるトンネルのことを「クローン」と呼びます。クローントンネルは、設定済みのメインのトンネルのプロパティを継承します。ただし、ユーザが設定した負荷間隔を継承することはできません。元のトンネルとそのクローンをまとめて「セット」と呼びます。元のトンネルに対して作成できるクローンの数について、上限と下限を指定できます。

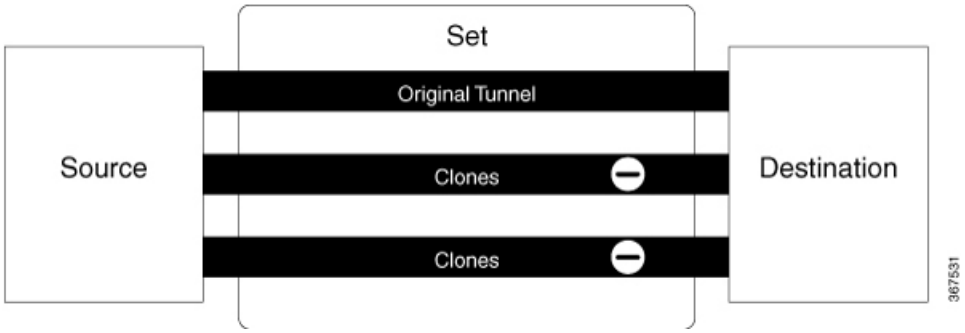
帯域幅の増加が必要な場合に新しいトンネルのクローンを作成するプロセスのことを「スプリット」と呼びます。セット内のいずれかのトンネルのサイズが、設定したスプリット帯域幅を上回ると、スプリット割が開始され、クローン トンネルが作成されます。

次の図は、スプリット帯域幅を超えた場合のクローン トンネルの作成について説明しています。



帯域幅の要求が減少したときにクローン トンネルを削除するプロセスのことを「マージ」と呼びます。帯域幅が、セット内のいずれかのトンネルで設定されたマージ帯域幅を下回ると、クローン トンネルが削除されます。

次の図は、帯域幅がマージ帯域幅を下回った場合のクローン トンネルの削除と元のトンネルとのマージについて説明しています。



セット内のトンネル間で集約帯域幅の要求を均等にロードシェアリングするには、複数の方法があります。これは、集約帯域幅の要件を満たすペアを選択するためのアルゴリズムが必要であることを意味します。アルゴリズムを導くガイドとして名目帯域幅を設定し、トンネルの平均帯域幅を決定することができます。名目帯域幅を設定しない場合、TE は、スプリットおよびマージ帯域幅の平均を名目帯域幅として使用します。

制約事項および使用上の注意事項

自動帯域幅バンドル TE++ 機能には、次の使用上のガイドラインが適用されます。

- この機能は名前付きトンネルでのみサポートされており、tunnel-te インターフェイスではサポートされていません。
- クローン数の下限値の範囲は 0 ～ 63 で、クローン数の下限のデフォルト値は 0 に設定されています。
- クローン数の上限値の範囲は 1 ～ 63 で、クローン数の上限のデフォルト値は 63 に設定されています。

設定例

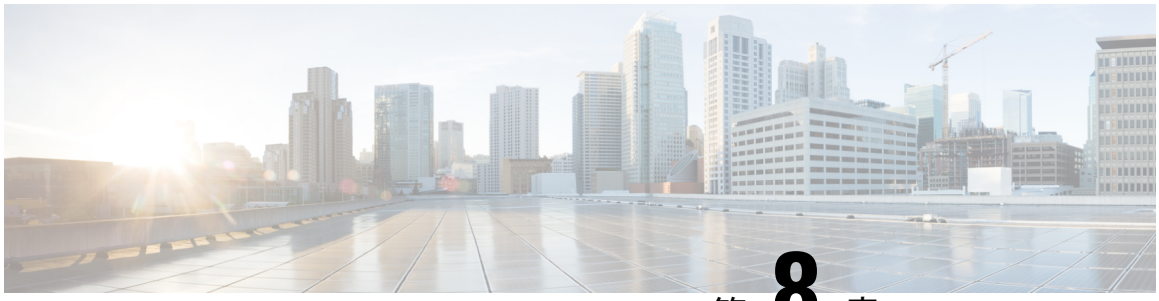
次に、名前付き MPLS-TE トラフィック トンネルにおいて自動帯域幅バンドル TE++ 機能を設定する例を示します。この機能を動作させるには、次の値を設定する必要があります。

- min-clones：元のトンネルが作成できるクローン トンネルの最小数を指定します。
- max-clones：元のトンネルが作成できるクローン トンネルの最大数を指定します。
- nominal-bandwidth：要求全体を満たすトンネル数を計算するための平均帯域幅を指定します。
- split-bandwidth：元のトンネルを分割するための帯域幅の値を指定します。トンネル帯域幅が、設定したスプリット帯域幅を超えると、クローン トンネルが作成されます。

- merge-bandwidth：元のトンネルとクローンをマージするための帯域幅を指定します。帯域幅が、設定したマージ帯域幅を下回ると、クローン トンネルが削除されます。

この例では、クローン数の下限を 2 に設定し、クローン数の上限を 4 に設定しています。スプリットおよびマージの帯域幅サイズを、200 および 100 kbps と設定しています。

```
RP/0/RP0/CPU0:router(config)# mpls traffic-eng
RP/0/RP0/CPU0:router(config-mpls-te)# named-tunnels
RP/0/RP0/CPU0:router(config-te-named-tunnels)# tunnel-te xyz
RP/0/RP0/CPU0:router(config-te-tun-name)# auto-bw
RP/0/RP0/CPU0:router(config-mpls-te-tun-autobw)# auto-capacity
RP/0/RP0/CPU0:router(config-te-tun-autocapacity)# min-clones 2
RP/0/RP0/CPU0:router(config-te-tun-autocapacity)# max-clones 4
RP/0/RP0/CPU0:router(config-te-tun-autocapacity)# nominal-bandwidth 5
RP/0/RP0/CPU0:router(config-te-tun-autocapacity)# split-bandwidth 200
RP/0/RP0/CPU0:router(config-te-tun-autocapacity)# merge-bandwidth 100
```



第 8 章

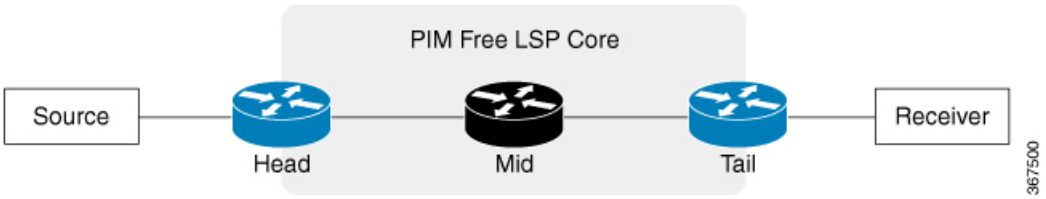
Point to Multipoint Traffic Engineering（ポイントツーマルチポイント トラフィック エンジニアリング）の設定

サービス プロバイダー ネットワークでマルチキャスト トラフィックを伝送するには、PIM などのマルチキャスト プロトコルを展開し、サービス プロバイダー コアで転送パスを設定する必要があります。ただし、MPLS バックボーン ネットワークの場合、サービス プロバイダーは ip トンネリングではなくラベル カプセル化を使用できます。この方式は、サービス プロバイダー コアの制御トラフィックのオーバーヘッドを減らすのに役立ちます。また、MPLS トラフィック エンジニアリングおよび保護機能を活用できます。

ラベルのカプセル化は、ポイントツーマルチポイント（P2MP）ラベルスイッチドパス（LSP）か、マルチポイントツーマルチポイント（MP2MP）LSP のどちらかになります。マルチキャスト LSP を作成する場合は、RSVP-TE プロトコル拡張を使用できます。RSVP-TE プロトコルは、MPLS ネットワーク全体で P2MP LSP をシグナリングするように拡張されています。P2MP-TE 機能により、P2MP-TE トンネルを使用した、PIM フリー サービス プロバイダー コア経由でのマルチキャスト トラフィックの伝送が可能になります。

次の図は、この機能で使用されるトポロジについて説明しています。

図 *15 : PIM* フリー *LSP* コア



この図では次の用語を使用しています。

- ヘッド：TE トンネルが設定されているルータ。
- テール：TE トンネルの終端となるルータ。
- ミッド：TE トンネルが通過するルータ。

マルチキャスト VPN（mVPN）プロファイルは、グローバル コンテキストに対して、または VRF ごとに設定されます。マルチキャスト ストリームをどこに転送する必要があるかに応じて、異なる mVPN プロファイルを適用できます。

P2MP-TE 機能では、次の mVPN プロファイルがサポートされています。

- mVPN プロファイル 8（グローバル コンテキスト用）
- mVPN プロファイル 10（L3VPN コンテキスト用）

制約事項および使用上の注意事項

この機能には次の制約事項と注意事項が適用されます。

- Source-Specific Multicast（SSM）トラフィックのみがサポートされます。
- プロファイル 8 で は、IPv4 と IPv6 の両方がサポートされます。
- プロファイル 10 では、IPv4 のみがサポートされます。
- P2MP-TE トンネルの Fast Reroute（FRR）はサポートされません。
- BVI インターフェイスはサポートされません。

設定例：P2MP-TE プロファイル 8

この例は、プロファイル 8 の P2MP-TE 設定を示しています。P2MP トンネルで、ヘッド、ミッド、テールの各ルータを設定する必要があります。

ヘッド ルータの設定は次のようになります。この設定には、IGP、MPLS-TE トンネル、およびマルチキャストの設定が含まれます。この機能を設定する際には、LDP と RSVP も設定する必要があります。

```
RP/0/RP0/CPU0:router(config)# router ospf 1
RP/0/RP0/CPU0:router(config-router)# area 0
RP/0/RP0/CPU0:router(config-ospf-ar)# mpls traffic-eng
RP/0/RP0/CPU0:router(config-ospf-ar-mpls-te)# exit
RP/0/RP0/CPU0:router(config-ospf-ar)# interface Loopback0
RP/0/RP0/CPU0:router(config-ospf-ar-if)# exit
RP/0/RP0/CPU0:router(config-ospf-ar)# interface TenGigE0/0/0/0
RP/0/RP0/CPU0:router(config-ospf-ar-if)# cost 1
RP/0/RP0/CPU0:router(config-ospf-ar-if)# network point-to-point
RP/0/RP0/CPU0:router(config-ospf-ar-if)# exit
RP/0/RP0/CPU0:router(config-ospf-ar)# interface TenGigE0/0/0/2
RP/0/RP0/CPU0:router(config-ospf-ar-if)# cost 1
RP/0/RP0/CPU0:router(config-ospf-ar-if)# network point-to-point
RP/0/RP0/CPU0:router(config-ospf-ar-if)# exit
RP/0/RP0/CPU0:router(config-ospf-ar)# exit
RP/0/RP0/CPU0:router(config-ospf)# mpls traffic-eng router-id loopback 0
RP/0/RP0/CPU0:router(config)# interface tunnel-mte 2
RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0
RP/0/RP0/CPU0:router(config-if)# destination 10.2.2.2
RP/0/RP0/CPU0:router(config-if-p2mp-dest)# path-option 1 dynamic
RP/0/RP0/CPU0:router(config-if-p2mp-dest)# exit
RP/0/RP0/CPU0:router(config)# multicast-routing
RP/0/RP0/CPU0:router(config-mcast)# address-family ipv4
RP/0/RP0/CPU0:router(config-mcast-default-ipv4)# interface Loopback0
```

```
RP/0/RP0/CPU0:router(config-mcast-default-ipv4-if)# enable
RP/0/RP0/CPU0:router(config-mcast-default-ipv4-if)# exit
RP/0/RP0/CPU0:router(config-mcast-default-ipv4)# interface tunnel-mte 2
RP/0/RP0/CPU0:router(config-mcast-default-ipv4-if)# enable
RP/0/RP0/CPU0:router(config-mcast-default-ipv4-if)# exit
RP/0/RP0/CPU0:router(config-mcast-default-ipv4)# mdt source Loopback0
RP/0/RP0/CPU0:router(config-mcast-default-ipv4)# interface all enable
RP/0/RP0/CPU0:router(config-mcast-default-ipv4)# accounting per-prefix
RP/0/RP0/CPU0:router(config)# router igmp
RP/0/RP0/CPU0:router(config-igmp)# interface tunnel-mte 2
RP/0/RP0/CPU0:router(config-igmp-if)# static-group 232.0.0.2 10.0.0.100
RP/0/RP0/CPU0:router(config-igmp)# interface TenGigE0/0/0/0
RP/0/RP0/CPU0:router(config-igmp-if)# version 3
RP/0/RP0/CPU0:router(config)# router pim
RP/0/RP0/CPU0:router(config-pim)# address-family ipv4
RP/0/RP0/CPU0:router(config-pim-default-ipv4)# interface tunnel-mte 2
RP/0/RP0/CPU0:router(config-pim-default-ipv4-if)# enable
RP/0/RP0/CPU0:router(config-pim-default-ipv4-if)# exit
```

ヘッド ルータの実行コンフィギュレーションは次のようになります。

```
interface Loopback0
 ipv4 address 10.1.1.1 255.255.255.255
!
interface TenGigE0/0/0/0
ipv4 address 10.0.0.1 255.255.255.0
!
interface TenGigE0/0/0/2
ipv4 address 10.2.0.1 255.255.255.0
!
router ospf 1
 area 0
  mpls traffic-eng
  !
  interface Loopback0
  interface TenGigE0/0/0/0
   cost 1
   network point-to-point
  interface TenGigE0/0/0/2
   cost 1
   network point-to-point
  !
 mpls traffic-eng router-id Loopback0
!
rsvp
 interface TenGigE0/0/0/2
  bandwidth percentage 100
 !
!
mpls traffic-eng
 interface TenGigE0/0/0/2
!
mpls ldp
 discovery
  targeted-hello interval 10
 !
 router-id 10.1.1.1
 address-family ipv4
  discovery targeted-hello accept
 !
 interface TenGigE0/0/0/2
 !
!
!
```

```
interface tunnel-mte2
 ipv4 unnumbered Loopback0
 destination 10.2.2.2
  path-option 1 dynamic
 !
!
!
multicast-routing
 address-family ipv4
  interface Loopback0
   enable
  !
  interface tunnel-mte2
   enable
  !
  mdt source Loopback0
  interface all enable
  accounting per-prefix
  !
 !
!
router igmp
  interface tunnel-mte2
   static-group 232.0.0.2 10.0.0.100
  !
  interface TenGigE0/0/0/0
   version 3
  !
!
router pim
  address-family ipv4
   interface tunnel-mte2
    enable
   !
  !
!
```

ミッドルータには、MPLS-TE、RSVP、および OSPF などの IGP の設定のみ必要です。ミッドルータの実行コンフィギュレーションは次のようになります。

```
interface Loopback0
 ipv4 address 10.5.5.5 255.255.255.255
interface TenGigE0/0/0/2
 ipv4 address 10.10.0.5 255.255.255.0
interface TenGigE0/0/0/3
 ipv4 address 10.13.0.5 255.255.255.0
router ospf 1
 area 0
  mpls traffic-eng
  interface Loopback0
  interface TenGigE0/0/0/2
   cost 1
   network point-to-point
  interface TenGigE0/0/0/3
   cost 1
   network point-to-point
 mpls traffic-eng router-id Loopback0
rsvp
 interface TenGigE0/0/0/2
  bandwidth percentage 100
 interface TenGigE0/0/0/3
  bandwidth percentage 100
mpls traffic-eng
```

```
 interface TenGigE0/0/0/2
 interface TenGigE0/0/0/3
mpls ldp
 discovery
  targeted-hello interval 10
 router-id 10.5.5.5
 address-family ipv4
  discovery targeted-hello accept
 interface TenGigE0/0/0/2
 interface TenGigE0/0/0/3
 !
!
```

テール ルータの設定は次のようになります。この設定には、IGP、MPLS-TE トンネル、およびマルチキャストの設定が含まれます。ヘッド ルータと同様に、この機能を設定する際には RSVP と LDP も設定する必要があります。

```
RP/0/RP0/CPU0:router(config)# router ospf 1
RP/0/RP0/CPU0:router(config-router)# area 0
RP/0/RP0/CPU0:router(config-ospf-ar)# mpls traffic-eng
RP/0/RP0/CPU0:router(config-ospf-ar-mpls-te)# exit
RP/0/RP0/CPU0:router(config-ospf-ar)# interface Loopback0
RP/0/RP0/CPU0:router(config-ospf-ar-if)# exit
RP/0/RP0/CPU0:router(config-ospf-ar)# interface TenGigE0/0/0/3
RP/0/RP0/CPU0:router(config-ospf-ar-if)# cost 1
RP/0/RP0/CPU0:router(config-ospf-ar-if)# network point-to-point
RP/0/RP0/CPU0:router(config-ospf-ar-if)# exit
RP/0/RP0/CPU0:router(config-ospf-ar)# exit
RP/0/RP0/CPU0:router(config-ospf)# mpls traffic-eng router-id loopback 0
RP/0/RP0/CPU0:router(config)# interface tunnel-mte 2
RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0
RP/0/RP0/CPU0:router(config-if)# destination 10.2.2.2
RP/0/RP0/CPU0:router(config-if-p2mp-dest)# path-option 1 dynamic
RP/0/RP0/CPU0:router(config-if-p2mp-dest)# exit
RP/0/RP0/CPU0:router(config)# multicast-routing
RP/0/RP0/CPU0:router(config-mcast)# address-family ipv4
RP/0/RP0/CPU0:router(config-mcast-default-ipv4)# interface Loopback0
RP/0/RP0/CPU0:router(config-mcast-default-ipv4-if)# enable
RP/0/RP0/CPU0:router(config-mcast-default-ipv4-if)# exit
RP/0/RP0/CPU0:router(config-mcast-default-ipv4)# mdt source Loopback0
RP/0/RP0/CPU0:router(config-mcast-default-ipv4)# core-tree-protocol rsvp-te
RP/0/RP0/CPU0:router(config-mcast-default-ipv4)# static-rpf 10.0.0.100 32 mpls 1.1.1.1
RP/0/RP0/CPU0:router(config-mcast-default-ipv4)# rate-per-route
RP/0/RP0/CPU0:router(config-mcast-default-ipv4)# interface all enable
RP/0/RP0/CPU0:router(config-mcast-default-ipv4)# accounting per-prefix
RP/0/RP0/CPU0:router(config)# router igmp
RP/0/RP0/CPU0:router(config-igmp# interface TenGigE0/0/0/3
RP/0/RP0/CPU0:router(config-igmp-if)# version 3
RP/0/RP0/CPU0:router(config)# router pim
RP/0/RP0/CPU0:router(config-pim)# address-family ipv4
RP/0/RP0/CPU0:router(config-pim-default-ipv4)# interface TenGigE0/0/0/3
RP/0/RP0/CPU0:router(config-pim-default-ipv4-if)# enable
RP/0/RP0/CPU0:router(config-pim-default-ipv4-if)# exit
```

テール ルータの実行コンフィギュレーションは次のようになります。

```
!
interface Loopback0
 ipv4 address 10.2.2.2 255.255.255.255
!
interface TenGigE0/0/0/3
 ipv4 address 10.3.0.2 255.255.255.0
```

```
!
interface TenGigE0/0/0/6
 ipv4 address 10.6.0.2 255.255.255.0
!
router ospf 1
 area 0
  mpls traffic-eng
  interface Loopback0
  !
  interface TenGigE0/0/0/3
   cost 1
   network point-to-point
  !
!
mpls traffic-eng router-id Loopback0
!
rsvp
 interface TenGigE0/0/0/3
  bandwidth percentage 100
 !
!
mpls traffic-eng
 interface TenGigE0/0/0/3
!
mpls ldp
 discovery
  targeted-hello interval 10
 !
 router-id 10.2.2.2
 address-family ipv4
  discovery targeted-hello accept
 !
 interface TenGigE0/0/0/3
 !
!
multicast-routing
 address-family ipv4
interface Loopback0
   enable
  !
  mdt source Loopback0
  core-tree-protocol rsvp-te
  static-rpf 10.0.0.100 32 mpls 10.1.1.1
  rate-per-route
  interface all enable
  accounting per-prefix
  !
 !
!
router igmp
 !
 interface TenGigE0/0/0/6
   version 3
 !
!
router pim
 !
 address-family ipv4
   interface TenGigE0/0/0/6
     enable
   !
  !
!
```

設定例：P2MP-TE プロファイル 10

この例は、プロファイル 10 の P2MP-TE 設定を示しています。ヘッド、ミッド、テールの各ルータを設定する必要があります。

ヘッドルータの設定は次のようになります。この設定には、IGP、L3VPN、およびマルチキャストの設定が含まれます。この機能を設定する際には、MPLS-TE、LDP、およびRSVP も設定する必要があります。

```
RP/0/RP0/CPU0:router(config)# router ospf 1
RP/0/RP0/CPU0:router(config-router)# area 0
RP/0/RP0/CPU0:router(config-ospf-ar)# mpls traffic-eng
RP/0/RP0/CPU0:router(config-ospf-ar-mpls-te)# exit
RP/0/RP0/CPU0:router(config-ospf-ar)# interface Loopback0
RP/0/RP0/CPU0:router(config-ospf-ar-if)# exit
RP/0/RP0/CPU0:router(config-ospf-ar)# interface TenGigE0/0/0/2
RP/0/RP0/CPU0:router(config-ospf-ar-if)# cost 1
RP/0/RP0/CPU0:router(config-ospf-ar-if)# network point-to-point
RP/0/RP0/CPU0:router(config-ospf-ar-if)# exit
RP/0/RP0/CPU0:router(config-ospf-ar)# exit
RP/0/RP0/CPU0:router(config-ospf)# mpls traffic-eng router-id loopback 0
RP/0/RP0/CPU0:router(config-ospf)# exit
RP/0/RP0/CPU0:router(config)# vrf vpn_2
RP/0/RP0/CPU0:router(config-vrf)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-vrf-af)# import route-target 100:2
RP/0/RP0/CPU0:router(config-vrf-af)# export route-target 120:2
RP/0/RP0/CPU0:router(config)# interface TengigE0/0/0/0
RP/0/RP0/CPU0:router(config-if)# vrf vpn_2
RP/0/RP0/CPU0:router(config-if-vrf)# ipv4 address 10.0.0.1 255.255.255.0
RP/0/RP0/CPU0:router(config)# route-policy pass-all
RP/0/RP0/CPU0:router(config)# pass
RP/0/RP0/CPU0:router(config)#router bgp 1
RP/0/RP0/CPU0:router(config-bgp)# bgp router-id 10.1.1.1
RP/0/RP0/CPU0:router(config-bgp)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-bgp-af)# address-family vpnv4 unicast
RP/0/RP0/CPU0:router(config-bgp-af)# address-family ipv4 mvpn
RP/0/RP0/CPU0:router(config-bgp)# neighbor 10.2.2.2
RP/0/RP0/CPU0:router(config-bgp-nbr)# remote-as 1
RP/0/RP0/CPU0:router(config-bgp-nbr)# update-source Loopback0
RP/0/RP0/CPU0:router(config-bgp-nbr)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-bgp-nbr-af)# route-policy pass-all in
RP/0/RP0/CPU0:router(config-bgp-nbr-af)# route-policy pass-all out
RP/0/RP0/CPU0:router(config-bgp-nbr)# address-family vpnv4 unicast
RP/0/RP0/CPU0:router(config-bgp-nbr-af)# route-policy pass-all in
RP/0/RP0/CPU0:router(config-bgp-nbr-af)# route-policy pass-all out
RP/0/RP0/CPU0:router(config-bgp-nbr)# address-family ipv4 mvpn
RP/0/RP0/CPU0:router(config-bgp)# vrf vpn_2
RP/0/RP0/CPU0:router(config-bgp-vrf)#rd 100:2
RP/0/RP0/CPU0:router(config-bgp-vrf)#address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-bgp-vrf-af)#label mode per-prefix
RP/0/RP0/CPU0:router(config-bgp-vrf-af)#redistribute connected
RP/0/RP0/CPU0:router(config-bgp-vrf-af)#exit
RP/0/RP0/CPU0:router(config-bgp-vrf)# address-family ipv4 mvpn
RP/0/RP0/CPU0:router(config)# multicast-routing
RP/0/RP0/CPU0:router(config-mcast)# address-family ipv4
RP/0/RP0/CPU0:router(config-mcast-default-ipv4)# interface Loopback0
RP/0/RP0/CPU0:router(config-mcast-default-ipv4-if)# enable
RP/0/RP0/CPU0:router(config-mcast-default-ipv4-if)# exit
RP/0/RP0/CPU0:router(config-mcast-default-ipv4)# mdt source Loopback0
RP/0/RP0/CPU0:router(config-mcast)# vrf vpn_2
RP/0/RP0/CPU0:router(config-mcast-vpn_2)# address-family ipv4
RP/0/RP0/CPU0:router(config-mcast-vpn_2-ipv4)# mdt source loopback0
```

```
RP/0/RP0/CPU0:router(config-mcast-vpn_2-ipv4)# rate-per-route
RP/0/RP0/CPU0:router(config-mcast-vpn_2-ipv4)# interface all enable
RP/0/RP0/CPU0:router(config-mcast-vpn_2-ipv4)# bgp auto-discovery p2mp-te
RP/0/RP0/CPU0:router(config-mcast-vpn_2-ipv4-bgp-ad)# mdt static p2mp-te tunnel-mte2
RP/0/RP0/CPU0:router(config)# router igmp
RP/0/RP0/CPU0:router(config-igmp)# vrf vpn_2
RP/0/RP0/CPU0:router(config-igmp-vpn_2)# interface tunnel-mte2
RP/0/RP0/CPU0:router(config-igmp-vpn_2-if)# static-group 239.0.0.1 100.0.0.100
RP/0/RP0/CPU0:router(config-igmp-vpn_2-if)# exit
RP/0/RP0/CPU0:router(config-igmp-vpn_2)# interface TenGigE0/0/0/0
RP/0/RP0/CPU0:router(config-igmp-vpn_2-if)# version 3
RP/0/RP0/CPU0:router(config-igmp-vpn_2-if)# exit
RP/0/RP0/CPU0:router(config-igmp-vpn_2)#
RP/0/RP0/CPU0:router(config)# router pim
RP/0/RP0/CPU0:router(config-pim)# vrf vpn_2
RP/0/RP0/CPU0:router(config-pim-vpn_2)# address-family ipv4
RP/0/RP0/CPU0:router(config-pim-vpn_2-ipv4)# interface tunnel-mte2
RP/0/RP0/CPU0:router(config-pim-vpn_2-ipv4-if)# enable
RP/0/RP0/CPU0:router(config-pim-vpn_2-ipv4-if)# exit
RP/0/RP0/CPU0:router(config-pim-vpn_2-ipv4)# interface TenGigE0/0/0/0
RP/0/RP0/CPU0:router(config-pim-vpn_2-ipv4-if)# enable
```

ヘッド ルータの実行コンフィギュレーションは次のようになります。

```
!
interface Loopback0
 ipv4 address 10.1.1.1 255.255.255.255
!
interface TenGigE0/0/0/2
ipv4 address 10.2.0.1 255.255.255.0
!
router ospf 1
 area 0
  mpls traffic-eng
  !
  interface Loopback0
  !
  interface TenGigE0/0/0/2
   cost 1
   network point-to-point
  !
 mpls traffic-eng router-id Loopback0
!
rsvp
 interface TenGigE0/0/0/2
  bandwidth percentage 100
 !
!
mpls traffic-eng
 interface TenGigE0/0/0/2
!
mpls ldp
 discovery
  targeted-hello interval 10
 !
 router-id 10.1.1.1
 address-family ipv4
  discovery targeted-hello accept
 !
 interface TenGigE0/0/0/2
 !
!
vrf vpn_2
 address-family ipv4 unicast
```

```
  import route-target
   100:2
  export route-target
   100:2

interface TenGigE0/0/0/0
 vrf vpn_2
 ipv4 address 10.0.0.1 255.255.255.0

route-policy pass-all
  pass
end-policy

router bgp 1
 bgp router-id 10.1.1.1
 address-family ipv4 unicast
 address-family vpnv4 unicast
 address-family ipv4 mvpn
 neighbor 10.2.2.2
  remote-as 1
  update-source Loopback0
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  address-family vpnv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  address-family ipv4 mvpn
 vrf vpn_2
  rd 100:2
  address-family ipv4 unicast
   label mode per-prefix
   redistribute connected
  address-family ipv4 mvpn
hostname head
!
multicast-routing
 address-family ipv4
  interface Loopback0
   enable
  !
  mdt source Loopback0
  !
  vrf vpn_2
   address-family ipv4
    mdt source Loopback0
    rate-per-route
    interface all enable
    bgp auto-discovery p2mp-te
    !
    mdt static p2mp-te tunnel-mte2
   !
 !
!
router igmp
 vrf vpn_2
  interface tunnel-mte2
   static-group 239.0.0.1 100.0.0.100
  !
  interface TenGigE0/0/0/0
   version 3
  !
!
router pim
```

```
 vrf vpn_2
  address-family ipv4
   interface tunnel-mte2
    enable
   !
   interface TenGigE0/0/0/0
    enable
   !
  !
 !
```

ミッド ルータには、MPLS-TE、RSVP、および IGP の設定のみ必要です。ミッド ルータの実行コンフィギュレーションは次のようになります。

```
interface Loopback0
 ipv4 address 10.5.5.5 255.255.255.255

interface TenGigE0/0/0/2
 ipv4 address 10.0.0.5 255.255.255.0

interface TenGigE0/0/0/3
 ipv4 address 10.3.0.5 255.255.255.0

router ospf 1
 area 0
  mpls traffic-eng
  interface Loopback0
  interface TenGigE0/0/0/2
   cost 1
   network point-to-point
  interface TenGigE0/0/0/3
   cost 1
   network point-to-point
mpls traffic-eng router-id Loopback0

rsvp
 interface TenGigE0/0/0/2
  bandwidth percentage 100
 interface TenGigE0/0/0/3
  bandwidth percentage 100

mpls traffic-eng
 interface TenGigE0/0/0/2
 interface TenGigE0/0/0/3

mpls ldp
 discovery
  targeted-hello interval 10
 router-id 10.5.5.5
 address-family ipv4
  discovery targeted-hello accept
 interface TenGigE0/0/0/2
 interface TenGigE0/0/0/3
 !
!
```

テール ルータの設定は次のようになります。この設定には、L3VPN、マルチキャスト、および IGP の設定が含まれます。ヘッド ルータと同様に、この機能を設定する前に MPLS-TE と RSVP も設定する必要があります。

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# router ospf 1
```

```
RP/0/RP0/CPU0:router(config-router)# area 0
RP/0/RP0/CPU0:router(config-ospf-ar)# mpls traffic-eng
RP/0/RP0/CPU0:router(config-ospf-ar-mpls-te)# exit
RP/0/RP0/CPU0:router(config-ospf-ar)# interface Loopback0
RP/0/RP0/CPU0:router(config-ospf-ar-if)# exit
RP/0/RP0/CPU0:router(config-ospf-ar)# interface TenGigE0/0/0/3
RP/0/RP0/CPU0:router(config-ospf-ar-if)# cost 1
RP/0/RP0/CPU0:router(config-ospf-ar-if)# network point-to-point
RP/0/RP0/CPU0:router(config-ospf-ar-if)# exit
RP/0/RP0/CPU0:router(config-ospf-ar)# exit
RP/0/RP0/CPU0:router(config-ospf)# mpls traffic-eng router-id loopback 0
RP/0/RP0/CPU0:router(config-ospf)# exit
RP/0/RP0/CPU0:router(config)# vrf vpn_2
RP/0/RP0/CPU0:router(config-vrf)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-vrf-af)# import route-target 100:2
RP/0/RP0/CPU0:router(config-vrf-af)# export route-target 120:2
RP/0/RP0/CPU0:router(config)# interface TengigE0/0/0/6
RP/0/RP0/CPU0:router(config-if)# vrf vpn_2
RP/0/RP0/CPU0:router(config-if-vrf)# ipv4 address 10.0.0.1 255.255.255.0
RP/0/RP0/CPU0:router(config)# route-policy pass-all
RP/0/RP0/CPU0:router(config)# pass
RP/0/RP0/CPU0:router(config)# end-policy
RP/0/RP0/CPU0:router(config)# router bgp 1
RP/0/RP0/CPU0:router(config-bgp)# bgp router-id 10.2.2.2
RP/0/RP0/CPU0:router(config-bgp)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-bgp-af)# address-family vpnv4 unicast
RP/0/RP0/CPU0:router(config-bgp-af)# address-family ipv4 mvpn
RP/0/RP0/CPU0:router(config-bgp)# neighbor 10.1.1.1
RP/0/RP0/CPU0:router(config-bgp-nbr)# remote-as 1
RP/0/RP0/CPU0:router(config-bgp-nbr)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-bgp-nbr-af)# route-policy pass-all in
RP/0/RP0/CPU0:router(config-bgp-nbr-af)# route-policy pass-all out
RP/0/RP0/CPU0:router(config-bgp-nbr)# address-family vpnv4 unicast
RP/0/RP0/CPU0:router(config-bgp-nbr-af)# route-policy pass-all in
RP/0/RP0/CPU0:router(config-bgp-nbr-af)# route-policy pass-all out
RP/0/RP0/CPU0:router(config-bgp-nbr)# address-family ipv4 mvpn
RP/0/RP0/CPU0:router(config)# multicast-routing
RP/0/RP0/CPU0:router(config-mcast)# address-family ipv4
RP/0/RP0/CPU0:router(config-mcast-default-ipv4)# interface Loopback0
RP/0/RP0/CPU0:router(config-mcast-default-ipv4-if)# enable
RP/0/RP0/CPU0:router(config-mcast-default-ipv4-if)# exit
RP/0/RP0/CPU0:router(config-mcast-default-ipv4)# mdt source Loopback0
RP/0/RP0/CPU0:router(config-mcast)# vrf vpn_2
RP/0/RP0/CPU0:router(config-mcast-vpn_2)# address-family ipv4
RP/0/RP0/CPU0:router(config-mcast-vpn_2-ipv4)# mdt source loopback0
RP/0/RP0/CPU0:router(config-mcast-vpn_2-ipv4)# core-tree-protocol rsvp-te
RP/0/RP0/CPU0:router(config-mcast-vpn_2-ipv4)# rate-per-route
RP/0/RP0/CPU0:router(config-mcast-vpn_2-ipv4)# interface all enable
RP/0/RP0/CPU0:router(config-mcast-vpn_2-ipv4)# bgp auto-discovery p2mp-te
RP/0/RP0/CPU0:router(config)# router igmp
RP/0/RP0/CPU0:router(config-igmp)# vrf vpn_2
RP/0/RP0/CPU0:router(config-igmp-vpn_2)# interface TenGigE0/0/0/6
RP/0/RP0/CPU0:router(config-igmp-vpn_2-if)# version 3
RP/0/RP0/CPU0:router(config-igmp-vpn_2-if)# exit
RP/0/RP0/CPU0:router(config)# router pim
RP/0/RP0/CPU0:router(config-pim)# vrf vpn_2
RP/0/RP0/CPU0:router(config-pim-vpn_2)# address-family ipv4
RP/0/RP0/CPU0:router(config-pim-vpn_2-ipv4)# interface TenGigE0/0/0/6
RP/0/RP0/CPU0:router(config-pim-vpn_2-ipv4-if)# enable
RP/0/RP0/CPU0:router(config)#router bgp 1
RP/0/RP0/CPU0:router(config-bgp)# bgp router-id 192.168.1.2
RP/0/RP0/CPU0:router(config-bgp)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-bgp-af)# address-family vpnv4 unicast
RP/0/RP0/CPU0:router(config-bgp-af)# address-family ipv4 mvpn
```

```
RP/0/RP0/CPU0:router(config-bgp)# neighbor 192.168.1.1
RP/0/RP0/CPU0:router(config-bgp-nbr)# remote-as 2002
RP/0/RP0/CPU0:router(config-bgp-nbr)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-bgp-nbr-af)# route-policy pass-all in
RP/0/RP0/CPU0:router(config-bgp-nbr-af)# route-policy pass-all out
RP/0/RP0/CPU0:router(config-bgp-nbr)# address-family vpnv4 unicast
RP/0/RP0/CPU0:router(config-bgp-nbr-af)# route-policy pass-all in
RP/0/RP0/CPU0:router(config-bgp-nbr-af)# route-policy pass-all out
RP/0/RP0/CPU0:router(config-bgp-nbr)# address-family ipv4 mvpn
RP/0/RP0/CPU0:router(config-bgp)# vrf vpn_2
RP/0/RP0/CPU0:router(config-bgp-vrf)#rd 100:2
RP/0/RP0/CPU0:router(config-bgp-vrf)#address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-bgp-vrf-af)#label mode per-prefix
RP/0/RP0/CPU0:router(config-bgp-vrf-af)#redistribute connected
RP/0/RP0/CPU0:router(config-bgp-vrf-af)#exit
RP/0/RP0/CPU0:router(config-bgp-vrf)# address-family ipv4 mvpn
```

テール ルータの実行コンフィギュレーションは次のようになります。

```
interface Loopback0
 ipv4 address 10.2.2.2 255.255.255.255
!
interface TenGigE0/0/0/3
 ipv4 address 10.3.0.2 255.255.255.0
!
router ospf 1
 area 0
  mpls traffic-eng
  interface Loopback0
  !
  interface TenGigE0/0/0/3
   cost 1
   network point-to-point
  !
!
mpls traffic-eng router-id Loopback0
!
rsvp
 interface TenGigE0/0/0/3
  bandwidth percentage 100
 !
!
mpls traffic-eng
 interface TenGigE0/0/0/3
!
mpls ldp
 discovery
  targeted-hello interval 10
 !
 router-id 10.2.2.2
 address-family ipv4
  discovery targeted-hello accept
 !
 interface TenGigE0/0/0/3
 !
! vrf vpn_2
 address-family ipv4 unicast
  import route-target
   100:2
  export route-target
   100:2

interface TenGigE0/0/0/6
 vrf vpn_2
```

```
 ipv4 address 10.6.0.2 255.255.255.0

route-policy pass-all
  pass
end-policy

router bgp 1
 bgp router-id 10.2.2.2
 address-family ipv4 unicast
 address-family vpnv4 unicast
 address-family ipv4 mvpn
 neighbor 10.1.1.1
  remote-as 1
  update-source Loopback0
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  address-family vpnv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  address-family ipv4 mvpn
 vrf vpn_2
  rd 100:2
  address-family ipv4 unicast
   label mode per-prefix
   redistribute connected
  address-family ipv4 mvpn
!
multicast-routing
 address-family ipv4
  interface Loopback0
   enable
   !
   mdt source Loopback0
   !
  vrf vpn_2
   address-family ipv4
    mdt source Loopback0
    core-tree-protocol rsvp-te
    rate-per-route
    interface all enable
    bgp auto-discovery p2mp-te
   !
  !
router igmp
 vrf vpn_2
  interface TenGigE0/0/0/6
   version 3
 !
!
router pim
 vrf vpn_2
 address-family ipv4
   interface TenGigE0/0/0/6
     enable
   !
  !
!
```

検証：P2MP-TE

次に、**show mrib vrf vpn_2 route** コマンドを使用して、ヘッド ルータでのマルチキャスト制御の状態が正しいかどうかを確認する例を示します。

```
RP/0/RP0/CPU0:router# show mrib vrf vpn_2 route

(10.0.0.100,232.0.0.1) RPF nbr: 10.0.0.100 Flags: RPF
  Up: 00:00:38
  Incoming Interface List
    TenGigE0/0/0/0 Flags: A, Up: 00:00:38
  Outgoing Interface List
    Tunnel-mte2 Flags: F NS LI LVIF, Up: 00:00:38
```

また、テール ルータでのマルチキャスト制御の状態を確認することもできます。

```
RP/0/RP0/CPU0:router# show mrib vrf vpn_2 route

(10.0.0.100,232.0.0.1) RPF nbr: 10.1.1.1 Flags: RPF
  Up: 00:03:55
  Outgoing Interface List
    TenGigE0/0/0/6 Flags: F NS LI, Up: 00:03:55
```

次に、**show mpls traffic-eng tunnels p2mp** コマンドを使用して、ヘッド ルータで TE トンネルが確立されているかどうかを確認する例を示します。

```
RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels p2mp 2

 Name: tunnel-mte2
   Signalled-Name: head_mt2
   Status:
     Admin: up  Oper: up (Up for 00:09:37)
     Config Parameters:
      Bandwidth: 0 kbps (CT0) Priority: 7 7 Affinity: 0x0/0xffff
      Interface Bandwidth: 0 kbps
      Metric Type: TE (global)
      Fast Reroute: Not Enabled, Protection Desired: None
      Record Route: Not Enabled
      Reoptimization after affinity failure: Enabled
      Destination summary: (1 up, 0 down, 0 disabled) Affinity: 0x0/0xffff
      Auto-bw: disabled
      Destination: 10.2.2.2
        State: Up for 00:09:37
        Path options:
          path-option 1 dynamic     [active]
    Current LSP:
      lsp-id: 10002 p2mp-id: 2 tun-id: 2 src: 10.1.1.1 extid: 10.1.1.1
      LSP up for: 00:09:37 (since Fri May 25 22:32:03 UTC 2018)
      Reroute Pending: No
      Inuse Bandwidth: 0 kbps (CT0)
      Number of S2Ls: 1 connected, 0 signaling proceeding, 0 down      S2L Sub LSP:
Destination 2.2.2.2 Signaling Status: connected
        S2L up for: 00:09:37 (since Fri May 25 22:32:03 UTC 2018)
        Sub Group ID: 1 Sub Group Originator ID: 10.1.1.1
        Path option path-option 1 dynamic    (path weight 2)
        Path info (OSPF 1 area 0)
          10.0.0.5
          10.0.0.2
          10.2.2.2
    Reoptimized LSP (Install Timer Remaining 0 Seconds):
      None
    Cleaned LSP (Cleanup Timer Remaining 0 Seconds):
      None
Displayed 1 (of 101) heads, 0 (of 0) midpoints, 0 (of 0) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
```

次に、**show mpls forwarding p2mp** コマンドを使用して、ヘッド ルータでのラベルの割り当てを確認する例を示します。

```
RP/0/RP0/CPU0:router# show mpls forwarding p2mp

Local  Outgoing    Prefix            Outgoing     Next Hop        Bytes
Label  Label       or ID             Interface                    Switched
------ ----------- ----------------- ------------ --------------- ------------
64106  64008       P2MP TE: 2        TenGigE0/0/0/2    10.0.0.5        0
```