Cisco ASR 9000 シリーズ アグリゲーション サービス ルータ機能トラブルシューティング モジュール

Cisco ASR 9000 Series Aggregation Services Router Feature Troubleshooting Module

OL-17503-01-J

【注意】シスコ製品をご使用になる前に、安全上の注意（www.cisco.com/jp/go/safety_warning/）をご確認ください。

本書は、米国シスコシステムズ発行ドキュメントの参考和訳です。米国サイト掲載ドキュメントとの差異が生じる場合があるため、正式な内容については米国サイトのドキュメントを参照ください。また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。

このモジュールでは、Cisco ASR 9000 シリーズ Aggregation Services Router のルーティング技術機能に関するトラブルシューティング方法について説明します。

マニュアルの変更履歴

表 1 に、初版以降このマニュアルに加えられた技術的な変更内容を示します。

表 1　　マニュアルの変更履歴

リビジョン	日付	変更点
OL-17503-01-J	2009 年 7 月	このマニュアルの初回リリース (Cisco IOS XR ソフトウェア リリース 3.7.3)

マニュアルの入手方法およびテクニカル サポート

マニュアルの入手方法、テクニカル サポート、その他の有用な情報について、次の URL で、毎月更新される『*What's New in Cisco Product Documentation*』を参照してください。シスコの新規および改訂版の技術マニュアルの一覧も示されています。

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

『*What's New in Cisco Product Documentation*』は RSS フィードとして購読できます。また、リーダー アプリケーションを使用してコンテンツがデスクトップに直接配信されるように設定することもできます。RSS フィードは無料のサービスです。シスコは現在、RSS バージョン 2.0 をサポートしています。

目次

このモジュールの構成は次のとおりです。

フォワーディング アプリケーション

アクセス コントロール リスト（ACL）

Access Control List（ACL; アクセス コントロール リスト）は、パケットのフィルタリング、分析または転送の対象となるトラフィック タイプの選択、または何らかの方法で影響を受けるトラフィック タイプの選択に使用されます。Access Control Entry（ACE; アクセス コントロール エントリ）は、ACL に含まれる個々の permit（許可）文または deny（拒否）文です。各 ACE には、アクション要素（「許可」または「拒否）と、送信元アドレス、宛先アドレス、プロトコル、プロトコル固有のパラメータなどの基準に基づくフィルタ要素が含まれます。このセクションの内容は次のとおりです。

show および debug コマンドの使用方法

手順概要

1. **show access-lists ipv4 [rp-access [hardware {ingress | egress}** {*sequence-number* | **location** *node-id* | **summary [rp-access] | maximum [detail] [usage {***pfilter location node-id***}]**
2. **debug feature-ea-dll {all | error | info | resmgr | vmr}**

詳細手順

	コマンドまたはアクション	目的
ステップ 1	**show access-lists ipv4 [rp-access [hardware {ingress | egress} {*sequence-number* | **location** *node-id* | **summary [rp-access]** | **maximum [detail] [usage {*pfilter location node-id*}]** **例：** `RP/0/RSP0/CPU0:router# show access-lists ipv4 dtho 10 ipv4 access-list dtho 10 permit ipv4 any any`	すべての IPv4 ACL の内容を表示します。結果をフィルタリングするには、次のパラメータとキーワードを使用します。 • **access-list-name**：IPv4 ACL 名。 • **hardware**：ingress は入力方向のインターフェイスを指定し、egress は出力方向のインターフェイスを指定します。 • **sequence-number**：ACL 番号（1 ～ 2147483646）。 • **location** *node-id*：ACL の*ラック/スロット/モジュール*表記。 • **summary**：現在のすべての IPv4 ACL の概要。 • **maximum**：設定可能な最大の IPv4 ACL および ACE。 • **detail**：Out-of-resource（OOR; リソース枯渇）の詳細。設定される ACL と ACE の数は、OOR によって制限されます。 • **usage**：特定の LC での ACL の使用状況を表示します。 • **pfilter**：LC のパケット フィルタリング。
ステップ 2	**debug feature_ea_dll {all | trace | error | info}**	エラー メッセージをさまざまなレベルで表示します。

ACL メッセージが表示されない

ステップ 1 ACL の ACE を表示します。

```
RP/0/RSP0/CPU0:router# show access-lists ipv4
```

ステップ 2 ACL の TCAM エントリを表示します。

```
RP/0/RSP0/CPU0:router# show access-lists ipv4 hardware {ingress | egress} detail ...
```

ステップ 3 ACL の TCAM エントリを表示します。

ステップ 4 ACL でログを設定します。

```
RP/0/RSP0/CPU0:router# ipv4 access-lists log-update threshold
```

回避策

fragment フラグを持つエントリが存在しない場合は、すべてのインターフェイスからアクセスリストを削除し、fragment フラグを持つエントリを再適用します。

（注） フラグメント パケットは、fragment キーワードのない拒否 ACE とは一致しません。フラグメント パケットを拒否する ACE には、fragment キーワードを明示的に追加してください。

フラグメント パケットが受け入れられる

ステップ 1 ACL の ACE を表示します。

```
RP/0/RSP0/CPU0:router# show access-list ipv4
```

ステップ 2 IPv4 カウンタ（フラグメントなど）を表示します。

```
RP/0/RSP0/CPU0:router# show ipv4 traffic
```

ステップ 3 ACL の TCAM エントリを表示します。

ステップ 4 `RP/0/RSP0/CPU0:router# show access-list ipv4 hardware {ingress | egress} location`

ステップ 5 ACE に **fragment** キーワードが含まれるようにします。

```
RP/0/RSP0/CPU0:router# deny ipv4 any any fragments
```

ステップ 6 デバイスで受信されたフラグメント パケット数をチェックします。

ステップ 7 TCAM エントリを表示します。

回避策

フラグメント パケットは、**fragment** キーワードのない拒否 ACE とは一致しません。**fragment** フラグを持つエントリが存在しない場合は、次の手順を実行します。

ステップ 1 すべてのインターフェイスから ACL を削除します。

ステップ 2 フラグメント パケットを拒否する ACE に **fragment** キーワードを明示的に追加します。

ステップ 3 この ACL をすべてのインターフェイスに再適用します。

出力カウンタが正しくない、または機能しない

ステップ 1 既知のルートを表示します。

```
RP/0/RSP0/CPU0:router# show route ipv4
```

ステップ 2 ARP テーブルのエントリを表示します。ネクストホップを探します。

```
RP/0/RSP0/CPU0:router# show arp
```

ステップ 3 ACL の TCAM エントリを表示します。

```
RP/0/RSP0/CPU0:router# show access-list ipv4 hardware
```

回避策

ステップ 1 ルートが欠落している場合、または ARP が不完全な場合は、**no shut** コマンドを使用して回復します。

ステップ 2 UIDB テーブルまたは TCAM エントリが正しくない場合は、すべてのインターフェイスから ACL を削除して再適用します。

ACL のインターフェイスへのバインドが拒否される

設定が適用されたときに発生したエラーを表示します。

```
RP/0/RSP0/CPU0:router# show configuration failed
```

回避策

エラーが TCAM スペースに関連する場合は、ACL から ACE を削除します。TCAM エントリの数は ACL あたり 64 個に制限されています。

単一の ACE で多数の TCAM を使用する

ステップ 1 ACL の ACE を表示します。

```
RP/0/RSP0/CPU0:router# show access-list ipv4
```

ステップ 2 ACE に含まれる範囲の数をチェックします。

ACL で可変の TCAM スペースを使用する

Pre-Internal Forwarding Information Base（Pre-IFIB; プレ内部転送情報ベース）ハードウェア統計情報エントリを表示します。

```
RP/0/RSP0/CPU0:router# show lpts pifib brief
```

L3 インターフェイスのイーサネット サービスが機能しない

イーサネット サービスは、L3 インターフェイスではサポートされていません。

イーサネット サービスに関する ACL ログが機能しない

イーサネット サービスのロギングはサポートされていません。

イーサネット サービス ACL のインターフェイスへのバインドが拒否される

ステップ 1 設定が適用されたときに発生したエラーを表示します。

```
RP/0/RSP0/CPU0:router# show configuration failed
```

ステップ 2 ACL の ACE を表示します。

```
RP/0/RSP0/CPU0:router# show access-list ipv4
```

ステップ 3 pfilter_ea のトレース ログを表示します。

回避策

ステップ 1 ACL のフィールドがサポートされていない場合は、そのフィールドを ACE から削除します。

ステップ 2 TCAM スペースが枯渇した場合は、その ACL の ACE を削除します。

ステップ 3 ACL に含まれる範囲を減らします。

ACL の変更時に TCAM が枯渇する

該当する ACL に対して設定された ACE を表示します。

```
RP/0/RSP0/CPU0:router# show access-list {ethernet-service/ipv4}
```

回避策

新しい ACL を適用する前に、古い ACL を削除します。

ACL を削除できない

ステップ 1 設定が適用されたときに発生したエラーを表示します。

```
RP/0/RSP0/CPU0:router# show configuration error
```

ステップ 2 該当する ACL を使用しているインターフェイスを表示します。

```
RP/0/RSP0/CPU0:router# show access-list {ethernet-services | ipv4} usage pfilter
```

ステップ 3 IPv4 トレース情報を表示します。

```
RP/0/RSP0/CPU0:router# show access-list ipv4 trace
```

ステップ 4 イーサネット サービスのトレースを表示します。

```
RP/0/RSP0/CPU0:router# show access-list ethernet-services trace
```

DF ビットがサポートされていない

現在のリリースでは、Do not Fragment（DF）ビットは一致基準としてサポートされていません。

最大 ACL 数の制限に達した

ACL ID の最大数は、NP あたり 2048 です。ACL の名前と方向に関する TCAM エントリは、インターフェイス間で共有されます。

ACL 内のサポートされていない組み合わせ

アクセスリストにサポートされていないフィールドの組み合わせが指定されている場合があります。アクセスリストに指定されている組み合わせが現在サポートされていることを確認してください。現在のリリースでサポートされている組み合わせは次のとおりです。

- VLAN OUT + L2 PROTO + MAC SA + MAC DA
- VLAN OUT + VLAN IN + MAC SA + MAC DA
- VLAN OUT + VLAN IN + L2 PROTO + MAC DA

統計情報カウンタがない

統計情報カウンタは、現在のリリースではサポートされていません。

TCAM のリソースが枯渇した

「TCAMs Out of Resources」メッセージは、使用可能な数を超える TCAM エントリをプロビジョニングしようとしたことを意味します。

Quality of Service（QoS）

システムでは、次の種類の QoS が提供されます。

- ジッタ、遅延、およびパケット損失を最小限に抑える、音声およびビデオ アプリケーションのマルチレベル プライオリティ スケジューリング
- トラフィック負荷の最も高い時間帯でもすべての階層レイヤ全体にわたって音声とビデオのサービス完全性を確保する、プライオリティ伝達
- Differentiated Service Code Point（DSCP; DiffServ コード ポイント）、MPLS EXP ビット、および IEEE 802.1p IP precedence ビットを使用した、マーキング、ポリシング、スケジューリング、入力 / 出力による分類

このセクションの内容は次のとおりです。

show および debug コマンドの使用方法

手順概要

1. **show run policy-map**
2. **show run classmap**
3. **show run interface**
4. **show policy-map interface** *type interface-name* **[output | input]**
5. **show qos interface** *type interface-name* **[output | input]**
6. **show qos-ea interface** *type interface-name* **[output | input]**
7. **show qos-ea km**
8. **debug qos-ea ?**

詳細手順

	コマンドまたはアクション	目的
ステップ 1	**show run policy-map** **例：** `RP/0/RSP0/CPU0:router# show run policy-map l1-all`	名前を指定してポリシーマップを表示します。
ステップ 2	**show run classmap** **例：** `RP/0/RSP0/CPU0:router# show run class-map c2`	名前を指定してクラスマップ設定を表示します。
ステップ 3	**show run interface** **例：** `RP/0/RSP0/CPU0:router# show run interface g0/2/0/0`	特定のポート/サブインターフェイスのサービスポリシー バインディングを表示します。

	コマンドまたはアクション	目的
ステップ 4	**show policy-map interface g0/2/0/0 [output|input]** **例：** `RP/0/RSP0/CPU0:router# show policy-map inter g0/2/0/0`	すべての統計情報、キュー ID、およびクラス情報を表示します。
ステップ 5	**show qos interface** *type interface-name* **[output|input]** **例：** `RP/0/RSP0/CPU0:router# show qos int g0/2/0/0 out`	ハードウェア内の各クラスの設定をすべて表示します。
ステップ 6	**show qos-ea interface** *type interface-name* **[output|input]** **例：** `RP/0/RSP0/CPU0:router# show qos-ea int g0/2/0/0 out`	すべてのクラス情報構造を表示します。
ステップ 7	**show qos-ea km** **例：** `RP/0/RSP0/CPU0:router# show qos-ea km policy l2-all vmr interface g0/2/0/0 sw`	ポリシーマップ/インターフェイスのバインディングに関連付けられたキーマネージャ（TCAM キー マネージャ）関連のフィールドを表示します。
ステップ 8	**debug qos-ea ?** **例：** `RP/0/RSP0/CPU0:router# debug qos-ea ?`	—

サービスポリシー設定が拒否される

ステップ 1 サービスポリシー設定が拒否される、またはコミットできない場合は、**show configuration failed** コマンドを使用してエラー メッセージをチェックします。

ステップ 2 リソースの使用状況が設定値を超えている場合は、チェックポイントの回数を確認します。

`RP/0/RSP0/CPU0:router#` **show qos-ea ha chkpt all info location** *node-id*

ステップ 3 OOR をチェックします。

ステップ 4 使用されているリソースを確認します。

ステップ 5 概要情報を確認します。

`RP/0/RSP0/CPU0:router#` **show qos summary {queue | police | policy}**

パケットが不適切に分類される

ステップ 1 パケットが正しいインターフェイスに到着していることを確認します。

ステップ 2 パケットのフィールドが予想したとおりであることを確認します。

ステップ 3 パケット タイプを確認します。

ステップ 4 KM ポリシー情報が UIDB 設定と一致することを確認します。

```
RP/0/RSP0/CPU0:router# show qos-ea km policy policy info location filename

RP/0/RSP0/CPU0:router# show qos-ea km policy policy vmr interface filename hw detail
```

ステップ 5 各クラスの VMR エントリを確認します。

```
RP/0/RSP0/CPU0:router# show qos-ea km policy policy vmr interface filename hw detail
```

ステップ 6 パケットが実際にどのクラスと一致しているかを確認します。パケットのフィールドが異なるクラスに一致している場合は、NP マイクロコードによってこれをさらにデバッグする必要があります。

```
RP/0/RSP0/CPU0:router# show policy-map interface filename {output | input} [member
filename]
```

ステップ 7 入力時に L2 入力書き換えの前に QoS ルックアップが実行されているかどうか、および出力時に QoS ルックアップの前に L2 書き換えが実行されていることを確認します。

```
RP/0/RSP0/CPU0:router# show qos type interface {input | output} location node-id
```

ステップ 8
```
RP/0/RSP0/CPU0:router# show run interface type node-id
```

ステップ 9
```
RP/0/RSP0/CPU0:router# show run policy-map policy
```

ステップ 10
```
RP/0/RSP0/CPU0:router# show run class-map classmap
```

パケットが間違ったキューに格納される

ステップ 1
```
RP/0/RSP0/CPU0:router# show qos-ea type interface {input | output} location node-id
```

ステップ 2 パケットが正しく分類されていることを確認します。

ステップ 3 ハッシュ構造を確認します。

```
RP/0/RSP0/CPU0:router# show qos-ea type interface {input | output} location node-id
```

ステップ 4 クラスのハッシュ キーと、クラスのハッシュ結果が正しいキュー ID を持つことを確認します。

パケットが不適切にマークされる

ステップ 1 パケットが正しく分類されていることを確認します。

```
RP/0/RSP0/CPU0:router# show qos type interface {input | output} location node-id
```

ステップ 2 `RP/0/RSP0/CPU0:router# show qos-ea km policy policy vmr interface filename hw`

ステップ 3 `RP/0/RSP0/CPU0:router# show qos-ea km policy policy vmr interface filename sw`

ステップ 4 `RP/0/RSP0/CPU0:router# show policy-map interface filename {input | output} [member filename]`

ステップ 5 マーキング値を確認します。

```
RP/0/RSP0/CPU0:router# show qos type interface {input | output} location node-id
```

パケットが不適切にポリシングされる

ステップ 1 パケットが正しく分類されていることを確認します。

ステップ 2 ポリサーの CIR/CBS/PIR/PBS が、設定されたサービスポリシーに従って正しく設定されていることを確認します。また、トラフィックがどの程度のレートで到着しているかも確認し、ポリシング レートと照合します。

```
RP/0/RSP0/CPU0:router# show qos type interface {input | output} location node-id
```

ステップ 3 クラスのトークン バケットとポリシング ノード インデックスを取得します。

```
RP/0/RSP0/CPU0:router# show qos-ea type interface {input | output} location node-id
```

ステップ 4 `RP/0/RSP0/CPU0:router# show policy-map interface filename {input | output} [member filename]`

シェーピングが正しくない

ステップ 1 パケットが正しく分類されていることを確認します。

```
RP/0/RSP0/CPU0:router# show qos type interface {input | output} location node-id
```

ステップ 2 シェーパーの CIR/CBS/PIR/PBS が、設定されたサービスポリシーに従って正しく設定されていることを確認します。シェイプ プロファイル ID とエンティティ ハンドル情報（np、tm、レベル、インデックス、オフセット）を取得します。

```
RP/0/RSP0/CPU0:router# show qos-ea type interface {input | output} location node-id
```

ステップ 3 これらが正しく設定されている場合は、ハードウェア内のシェーパー プロファイルを確認します。

ステップ 4 `RP/0/RSP0/CPU0:router# show policy-map interface filename {input | output} [member filename]`

Weighted Random Early Detection（WRED）が正しくない

ステップ 1 パケットが正しく分類されていることを確認します。

ステップ 2 WRED カーブが正しく設定されていて、各カーブの最小および最大のしきい値が、設定されたサービスポリシーに従った適切な値であるかどうかを確認します。

```
RP/0/RSP0/CPU0:router# show qos type interface {input | output} location node-id
```

ステップ 3 WRED プロファイル ID とエンティティ ハンドル情報（np、tm、レベル、インデックス、オフセット）を取得します。

```
RP/0/RSP0/CPU0:router# show qos-ea type interface {input | output} location node-id
```

ステップ 4
```
RP/0/RSP0/CPU0:router# show policy-map interface filename {input | output} [member
filename]
```

帯域幅が保証されない

ステップ 1 パケットが正しく分類されていることを確認します。

ステップ 2 各クラスの重みが、クラス間の帯域幅比に従って正しく設定されているかどうかを確認します。

```
RP/0/RSP0/CPU0:router# show qos type interface {input | output} location node-id
```

ステップ 3
```
RP/0/RSP0/CPU0:router# show run policy-map policy
```

ステップ 4 正しく設定されている場合は、クラスの WFQ プロファイル ID とエンティティ ハンドル情報（np、tm、レベル、インデックス、オフセット）を取得します。

```
RP/0/RSP0/CPU0:router# show qos-ea type interface {input | output} location node-id
```

ステップ 5
```
RP/0/RSP0/CPU0:router# show policy-map interface filename {input | output} [member
filename]
```

帯域幅比が機能しない

ステップ 1 パケットが正しく分類されていることを確認します。

ステップ 2
```
RP/0/RSP0/CPU0:router# show run policy-map policy
```

ステップ 3 各クラスのコミット重みが、クラス間の帯域幅比に従って正しく設定されているかどうかを確認します。また、超過重みが、帯域幅の残りの比率の設定に従って設定されていることも確認します（超過重みの割り当てが超過重みの比率に収まっている必要があります）。

```
RP/0/RSP0/CPU0:router# show qos type interface {input | output} location node-id
```

ステップ 4
```
RP/0/RSP0/CPU0:router# show qos-ea type interface {input | output} location node-id
```

ステップ 5
```
RP/0/RSP0/CPU0:router# show policy-map interface filename {input | output} [member
filename]
```

ステップ 6 クラスの WFQ プロファイル ID とエンティティ ハンドル情報（np、tm、レベル、インデックス、オフセット）を取得します。

```
RP/0/RSP0/CPU0:router# show qos-ea type interface {input | output} location node-id
```

ステップ 7 コミット重みと超過重みが正しい場合は、次の手順を実行します。

a. 各クラスのキュー サイズをチェックします。

b. キュー サイズを増やします。

ポリシーマップまたはクラスマップを変更または削除できない

ステップ 1 ポリシーがインターフェイスに適用されていることを確認します。

```
RP/0/RSP0/CPU0:router# show running-config
```

ステップ 2 インターフェイス上のサービスポリシーを削除します。

ステップ 3 ポリシーマップを変更します。

クラスマップ ACL を変更または削除できない

- **show config failed**
- **show running-config**

ステップ 1 該当する ACL がクラスマップの match 文の一部であることを確認します。

ステップ 2 そのクラスマップが、インターフェイスに適用されているポリシーマップの一部であることを確認します。

ステップ 3 ポリシーマップがインターフェイスに適用されている場合、ACL の変更または削除は許可されません。

ステップ 4 このポリシーマップのサービスポリシー設定をすべて削除してから、ACL を変更します。

サービスポリシーを削除できない

ステップ 1

```
RP/0/RSP0/CPU0:router# show config failed
```

ステップ 2 qos_ma_ea プロセスを再起動します。

QoS EA が再起動した後、show policy-map interface が失敗する

- **show running-config**
- **show qos-ea ha chkpt all info location** *node-id*
- **show qos-ea ha chkpt if-qos all location** *node-id*

ステップ 1 QoS EA の状態が in_sync（state = 2）であるかどうかを確認します。

```
RP/0/RSP0/CPU0:router# show qos-ea ha state location node-id
```

ステップ 2 エラーがない場合は、次の手順を実行します。

a. RP/0/RSP0/CPU0:router# **debug generic**

b. 失敗するコマンドを実行して、デバッグを収集します。

QoS EA が再起動した後、service-policy config が失敗する

ステップ 1 QoS EA の状態が in_sync（state = 2）であるかどうかを確認します。

```
RP/0/RSP0/CPU0:router# show qos-ea ha state location node-id
```

ステップ 2 エラーがない場合は、次の手順を実行します。

a. RP/0/RSP0/CPU0:router# **debug generic**

b. 失敗するコマンドを実行して、デバッグを収集します。

show policy-map interface で出力エラーが発生する

バンドルの場合は、メンバ インターフェイスを指定します。バンドルインターフェイスのポリシー情報は、現在のリリースでは使用できません。

- **show policy-map interface {output | input} member**
- **show {qos | qos-ea} interface {output | input} location** *node-id*

サービスポリシーでバンドル メンバが設定されない

バンドルの場合は、メンバ インターフェイスを指定します。バンドルインターフェイスのポリシー情報は、現在のリリースでは使用できません。

- **show policy-map interface {output | input} member**
- **show {qos | qos-ea} interface {output | input} location** *node-id*

イーサネット/L2 機能

双方向フォワーディング検出（BFD）

Bidirectional Forwarding Detection（BFD; 双方向フォワーディング検出）は、すべてのメディア タイプ、カプセル化、トポロジ、およびルーティング プロトコルにおいてフォワーディング パスの障害検出を迅速化するために設計された検出プロトコルです。このセクションの内容は次のとおりです。

show および debug コマンドの使用方法

手順概要

1. **show bfd [ipv4 | all] [location** *node-id***]**
2. **show bfd client [detail]**
3. **show bfd [ipv4 | all] session [detail | interface** *ifname* **[destination** *address***]] [[agent] location** *node-id***]]**
4. **show bfd counters packet [interface** *ifname***] location** *node-id*
5. **show bfd trace {adjacency | error | fsm | packet} [filter {destination <address> | interface** *ifname***}] [location** *node-id***]**
6. **show tech-support routing bfd {terminal [page] | file** *send-to* **[background] [compressed | uncompressed]}**

詳細手順

	コマンドまたはアクション	目的
ステップ 1	**show bfd [ipv4 | all] [location** *node-id***]** **例：** `RP/0/RSP0/CPU0:router# show bfd location 0/4/CPU0`	RSP の全般的な BFD 情報（セッション数など）を表示します。**location** キーワードを指定すると、特定の LC の情報が表示されます。このキーワードを指定しない場合は、すべてのロケーションの情報が表示されます。
ステップ 2	**show bfd client [detail]** **例：** `RP/0/RSP0/CPU0:router# show bfd client detail`	BFD クライアントを表示します。**detail** キーワードを指定すると、詳細な情報が表示されます。

	コマンドまたはアクション	目的
ステップ 3	**show bfd [ipv4 | all] session [detail] [interface** *ifname* **[[agent] location** *node-id***]** **例：** `RP/0/RSP0/CPU0:router# show bfd session interface Gig2/1/0/0 detail`	BFD セッション情報を表示します。結果をフィルタリングするには、次のパラメータとキーワードを使用します。 • **location**：このロケーションでホストされている BFD セッション。 • **interface**：指定したインターフェイス（ワイルドカードなし）での BFD セッション。 • **detail**：統計情報、状態遷移の回数などの詳細なセッション情報。 • **agent**：RSP を飛び越して LC から直接情報を取得するために使用されるシスコサポート キーワード。
ステップ 4	**show bfd counters packet [interface** *ifname***] location** *node-id* **例：** `RP/0/RSP0/CPU0:router# show bfd counters packet interface POS 0/3/0/0 location 0/3/cpu0`	パケット カウンタ情報を表示します。結果をフィルタリングするには、次のパラメータとキーワードを使用します。 • **location**：このロケーションでホストされている BFD セッションのパケット カウンタ。 • **interface**：指定したインターフェイス（ワイルドカードなし）での BFD セッションのパケット カウンタ。 • **invalid**：無効パケットのカウンタ情報。
ステップ 5	**show bfd trace {adjacency | error | fsm | packet} [interface** *ifname***] [location** *node-id***]** **例：** `RP/0/RSP0/CPU0:router# show bfd trace fsm location 0/4/CPU0`	RSP からトレース情報を表示します。結果をフィルタリングするには、次のパラメータとキーワードを使用します。 • **adjacency**：BFD が Adjacency Information Base（AIB）の Finite State Machine（FSM; 有限状態マシン）表示から隣接関係の更新を受信したときに生成されるトレース。 • **error**：エラーが検出されたときに生成されるトレース。 • **fsm**：セッションで状態変更が起こったときに生成されるトレース。 • **packet**：Tx パケットまたは Rx パケットで変更が起こったときに生成されるトレース。 • **location**：指定したインターフェイスでの BFD トレースのトレース。 **（注）** 結果を保存するためにトレースをログ ファイルに記録してください。
ステップ 6	**show tech-support routing bfd {terminal | file}** **例：** `RP/0/RSP0/CPU0:router# show tech-support routing bfd terminal`	BFD デバッグ情報を表示します。

BFD セッションがダウン状態になる

ステップ 1 IP の接続性を確認します。IP パケットの損失がないことを確認します。

```
RP/0/RSP0/CPU0:router# ping local-remote-address
```

ステップ 2 ルータとリモート デバイスに次のパラメータが設定されていることを確認します。

a. それぞれがサポートできる BFD セッションの数

b. ポリシング レートをサポートするためのタイマー

BFD セッションのフラップが起こる

次の手順に従って、各種 BFD パラメータをチェックします。次のセクションも参照してください。

- 「隣接ルータで BFD セッションがダウンしている」（P.19）
- 「BFD セッションが LC で作成されない」（P.19）

ステップ 1 IP の接続性を確認します。

```
RP/0/RSP0/CPU0:router# ping local-IP-address
```

ステップ 2 入力カウンタと出力カウンタを確認します。

```
RP/0/RSP0/CPU0:router# show interface
```

ステップ 3 セッションの詳細情報を表示します。

```
RP/0/RSP0/CPU0:router# show bfd session detail
```

ステップ 4 セッションのパケット カウンタを表示します。

```
RP/0/RSP0/CPU0:router# show bfd counter
```

ステップ 5 SPP カウンタを表示します。

```
RP/0/RSP0/CPU0:router# show spp node
```

ステップ 6 リソースの使用状況を表示します。

```
RP/0/RSP0/CPU0:router# monitor process
```

ステップ 7 IP の接続性を表示します。IP パケットの損失がないことを確認します。

```
RP/0/RSP0/CPU0:router# ping local remote address
```

次のメッセージが表示された場合、BFD フラップの原因はアプリケーション フラップです。

```
bfd_agent[104]: %BFD-6-SESSION_REMOVED : BFD session to neighbor 192.1.1.1 on interface
Gi0/5/0/0 has been remove
```

ステップ 8 SPP でパケットが失われていないことを確認します。

```
RP/0/RSP0/CPU0:router# show spp node location
```

ステップ 9 LC の CPU とメモリの使用状況をチェックします。

```
RP/0/RSP0/CPU0:router# monitor processes location
```

ステップ 10 ローカル インターフェイスのカウンタをチェックします。

```
RP/0/RSP0/CPU0:router# show interfaces type interface-name
```

ステップ 11 インターフェイスに適用されている QoS ポリシーをチェックします。

```
RP/0/RSP0/CPU0:router# show policy-map interface
```

ステップ 12 リモート側でステップ 1 ～ 11 を繰り返します。

ローカル エコーの障害が原因で BFD セッションのフラップが起こる

BFD セッションのフラップは、ルータがエコー障害を検出したためにローカルで起こる場合があります。

LC の CPU 使用率を調べます。
`RP/0/RSP0/CPU0:router# monitor process location`

LC CPU の SPP プロセスを調べて、BFD エコー パケットで生じた遅延を確認します。
`RP/0/RSP0/CPU0:router# show bfd trace performance reverse location`

BFD エコー パケットの損失を除外します。**show bfd counters packet location**

SPP プロセスの再起動が原因で BFD セッションのフラップが起こる

BFD 障害検出が 1 秒以内に設定されている場合、LC で SPP プロセスが再起動すると、BFD セッションのフラップが起こります。

隣接ルータで BFD セッションがダウンしている

隣接ルータは、BFD がダウンしていることを知らせるために、次のメッセージを送信します。

```
LC/0/6/CPU0:Aug 8 16:42:56.821: bfd_agent[104]: %L2-BFD-6-SESSION_STATE_DOWN: BFD session
to neighbor 192.1.1.1 on interface Gi0/5/0/0 has gone down. Reason: Nbor signalled down
```

BFD セッションが LC で作成されない

1 つの LC で許可される BFD セッションの数は 1024 です。1024 を超える BFD セッションを設定すると、ランダムな BFD セッションが作成されなくなる場合があります。

接続障害管理（CFM）

Connectivity Fault Management（CFM; 接続障害管理）は、リモート ネットワークの障害をネットワーク越しにエンドツーエンドで監視、検出し、診断します。これには、キープアライブと MAC ベースの ping およびトレースルートが使用されます。このセクションの内容は次のとおりです。

show および debug コマンドの使用方法

手順概要

1. **debug ethernet cfm platform**
2. **debug ethernet oam platform**
3. **show spp node**
4. **show spp client**

詳細手順

	コマンドまたはアクション	目的
ステップ 1	**debug ethernet cfm platform** **例：** `RP/0/RSP0/CPU0:router# debug ethernet cfm platform`	CFM に関する PD 固有のデバッグ情報を表示します。
ステップ 2	**debug ethernet oam platform** **例：** `RP/0/RSP0/CPU0:router# debug ethernet oam platform`	OAM に関する PD 固有のデバッグ情報を表示します。

	コマンドまたはアクション	目的
ステップ 3	**show spp node** **例：** RP/0/RSP0/CPU0:router# show spp node	SPP カウンタを表示します。
ステップ 4	**show spp client** **例：** RP/0/RSP0/CPU0:router# show spp client	SPP ドロップを表示します。

MEP に CCM がない

ステップ 1 設定された MEP と MIP を表示します。

```
RP/0/RSP0/CPU0:router# show ethernet cfm local main
```

ステップ 2 MEP ローカル ノードに関する情報と CCM 統計情報を表示します。

```
RP/0/RSP0/CPU0:router# show ethernet cfm location mep
RP/0/RSP0/CPU0:router# show ethernet cfm peer mep
```

ステップ 3 特定の LC CFM インスタンスによって示されるリモート MEP を表示します。CCM が受信されていない場合、ピアは表示されません。

ステップ 4 SPP によって認識された CFM SID 統計情報を表示します。

```
RP/0/RSP0/CPU0:router# show spp sid stats
```

ステップ 5 CFM PI によって認識されたパケットを表示します。すべてのオプションをイネーブルにします。パケットがドロップ、転送、または処理された場合、出力が表示されます。

```
RP/0/RSP0/CPU0:router# debug ethernet cfm packet pack ccm
```

ステップ 6 SPP ドロップを表示します。

```
RP/0/RSP0/CPU0:router# show spp client location 0/2/cpu0
```

上位レベルのパケットが下位レベルの MEP に送信される

設定されたしきい値を超える上位レベルの CFM パケットは、NP によって転送されます。show interface コマンドで表示されるパケット数の値は、通常は受信パケットに一致します。

- **show interface**：インターフェイスの統計情報を表示します。

上位レベルのパケットが下位レベルの MEP に送信されたときにデバッグ メッセージが生成されない

ステップ 1 設定された MEP と MIP を表示します。

```
RP/0/RSP0/CPU0:router# show ethernet cfm local main
```

ステップ 2 MEP ローカル ノードに関する情報と CCM 統計情報を表示します。

```
RP/0/RSP0/CPU0:router# show ethernet cfm location mep
```

ステップ 3 特定の LC CFM インスタンスによって示されるリモート MEP を表示します。CCM が受信されていない場合、ピアは表示されません。

```
RP/0/RSP0/CPU0:router# show ethernet cfm peer mep
```

ステップ 4 SPP によって認識された CFM SID 統計情報を表示します。

```
RP/0/RSP0/CPU0:router# show spp sid stats
```

ステップ 5 CFM PI によって認識されたパケットを表示します。すべてのオプションをイネーブルにします。パケットがドロップ、転送、または処理された場合、出力が表示されます。

```
RP/0/RSP0/CPU0:router# debug ethernet cfm packet pack ccm
```

ステップ 6 SPP ドロップを表示します。

```
RP/0/RSP0/CPU0:router# show spp client location 0/2/cpu0
```

MEP で CCM がディセーブルになっており、リモート/ピア MEP がその影響を受ける

ステップ 1 設定された MEP と MIP を表示します。

```
RP/0/RSP0/CPU0:router# show ethernet cfm local main
```

ステップ 2 MEP ローカル ノードに関する情報と CCM 統計情報を表示します。

```
RP/0/RSP0/CPU0:router# show ethernet cfm location mep
```

ステップ 3 特定の LC CFM インスタンスによって示されるリモート MEP を表示します。CCM が受信されていない場合、ピアは表示されません。

```
RP/0/RSP0/CPU0:router# show ethernet cfm peer mep
```

ステップ 4 SPP によって認識された CFM SID 統計情報を表示します。

```
RP/0/RSP0/CPU0:router# show spp sid stats
```

ステップ 5 CFM PI によって認識されたパケットを表示します。すべてのオプションをイネーブルにします。パケットがドロップ、転送、または処理された場合、出力が表示されます。

```
RP/0/RSP0/CPU0:router# debug ethernet cfm packet pack ccm
```

CFM の ping およびトレースルートの結果が「not found」になる

ステップ 1 すべての CFM グローバル設定を表示します。

```
RP/0/RSP0/CPU0:router# show run ethernet cfm
```

ステップ 2 ローカルの MEP とその CCM 統計情報を表示します。

```
RP/0/RSP0/CPU0:router# show ethernet cfm location main
```

ステップ 3 ピア MEP から受信された CFM CCM を表示します。

```
RP/0/RSP0/CPU0:router# show ethernet cfm peer meps
```

ステップ 4 CFM の接続性を確認します。

```
RP/0/RSP0/CPU0:router# ping ethernet cfm
```

ステップ 5 CFM のトレースを開始します。

```
RP/0/RSP0/CPU0:router# trace ethernet cfm
```

ドロップされた CFM PDU

ステップ 1 インターフェイスあたりの CFM PDU の統計情報を表示します。

```
RP/0/RSP0/CPU0:router# show ethernet cfm interfaces statistics
```

ステップ 2 MST ステータスを表示します。

```
RP/0/RSP0/CPU0:router# show spanning-tree mst mstp
```

ステップ 3 すべてのローカル MEP によって認識されたピア MEP を表示します。

```
RP/0/RSP0/CPU0:router# show ethernet cfm peer meps
```

ステップ 4 MEP または MIP が設定されたインターフェイス上での STP ステータスをチェックします。STP ブロック ポート上の MEP から発信された CFM PDU は転送されますが、MIP で転送される PDU は STP ポートの状態に従います。これは、STP がブロックされているポートで MIP が設定されている場合、CFM PDU はその MIP でドロップされることを意味します。

ステップ 5 STP 状態と CFM ピアの MEP ステータスを表示します。

```
RP/0/RSP0/CPU0:router# show spanning-tree mst mstp
```

ステップ 6 パケットのデバッグを有効にして、転送されたパケットが MIP でドロップされることを確認します。

```
RP/0/RSP0/CPU0:router# debug ethernet cfm pack rec dropped int gig
```

CFM の ping でシーケンス エラーが表示される

ステップ 1 CFM の接続性を表示します。

```
RP/0/RSP0/CPU0:router# ping ethernet cfm
```

Dynamic Host Configuration Protocol（DHCP）スヌーピング

DHCP スヌーピングは DHCP のセキュリティ機能で、信頼されない DHCP メッセージをフィルタリングし、DHCP スヌーピングのバインディング テーブルを構築、維持することによってセキュリティを提供します。信頼されないメッセージとは、ネットワークまたはファイアウォールの外部から受信され、ネットワーク内でトラフィック攻撃を引き起こす可能性のあるメッセージのことです。ここでは、次のコマンドについて説明します。

- 「show コマンド」（P.24）
- 「trace コマンド」（P.25）
- 「syslog コマンド」（P.25）
- 「tech-support コマンド」（P.25）
- 「アクション コマンド」（P.25）
- 「L2VPN コマンド」（P.26）
- 「L2Snoop コマンド」（P.26）
- 「L2Snoop コマンド」（P.26）
- 「インターフェイス コントローラ コマンド」（P.26）

show コマンド

DHCP アプリケーションは RSP で実行されます。DHCP アプリケーションには、アプリケーションの設定状態、DHCP クライアントの状態、および DHCP パケットの統計情報を表示する EXEC モード CLI show コマンドがいくつかあります。

- **show dhcp ipv4 snoop binding**：DHCP クライアントの状態を表形式で表示します。
- **show dhcp ipv4 snoop binding mac-address** *macaddress*：指定した MAC アドレスを持つ DHCP クライアントの詳細な状態を表示します。
- **show dhcp ipv4 snoop binding summary**：DHCP クライアントの総数を表示します。
- **show dhcp ipv4 snoop profile**：DHCP スヌープ プロファイルの一覧を表示します。
- **show dhcp ipv4 snoop profile name** *name*：特定の DHCP スヌープ プロファイルの詳細を表示します。
- **show dhcp ipv4 snoop statistics**：DHCP スヌープの Rx パケット、Tx パケット、およびドロップ パケットの総計をブリッジ ドメインごとに表示します。
- **show dhcp ipv4 snoop statistics bridge-domain** *name*：特定のブリッジ ドメインにおける DHCP スヌープの Rx パケット、Tx パケット、およびドロップ パケットの詳細をメッセージ タイプごとに表示します。

trace コマンド

DHCP アプリケーションには 1200 以上のトレース ログがあります。トレース ログには、DHCP アプリケーションで発生した重要なイベントが記録されます。特定の DHCP クライアントに関連付けられたトレース ログには、そのクライアントの MAC アドレスが含まれます。

- **show dhcp ipv4 trace errors**：エラー トレースを表示します。
- **show dhcp ipv4 trace events**：イベント トレースを表示します。
- **show dhcp ipv4 trace packets**：パケット処理トレースを表示します。
- **show dhcp ipv4 trace snoop errors**：DHCP スヌープ機能のエラー トレースを表示します。
- **show dhcp ipv4 trace snoop events**：DHCP スヌープ機能のイベント トレースを表示します。
- **show dhcp ipv4 trace snoop internal**：DHCP スヌープ機能の内部デバッグ トレースを表示します。

syslog コマンド

DHCP アプリケーションには 1600 以上の syslog ログがあります。これらのログには、DHCP アプリケーションで発生したイベントが記録されます。

- **debug dhcp ipv4 errors**：エラー ログを表示します。
- **debug dhcp ipv4 events**：イベント ログを表示します。
- **debug dhcp ipv4 packets**：パケット処理ログを表示します。
- **debug dhcp ipv4 snoop errors**：DHCP スヌープ機能のエラー ログを表示します。
- **debug dhcp ipv4 snoop events**：DHCP スヌープ機能のイベント ログを表示します。
- **debug dhcp ipv4 snoop internal**：DHCP スヌープ機能の内部デバッグ ログを表示します。

tech-support コマンド

DHCP アプリケーションには、DHCP CLI コマンドのグループを呼び出す tech-support コマンドが 4 つ用意されています。tech-support コマンドは、デバッグの際に DHCP アプリケーションの情報を得るために使用します。

- **show tech-support dhcp ipv4 snoop file** *filename*
- **show tech-support dhcp ipv4 snoop bridge-domain-name** *bridgedomainname* **file** *filename*：指定したブリッジ ドメインの情報を表示します。
- **show tech-support dhcp ipv4 snoop profile-name** *profilename* **file** *filename*：指定したプロファイルの情報を表示します。

アクション コマンド

次の CLI コマンドは、DHCP スヌープのバインディング状態をクリアする場合に使用します。

- **clear dhcp ipv4 snoop binding**：すべての DHCP スヌープ クライアントのバインディングをクリアします。
- **clear dhcp ipv4 snoop binding bridge-domain** *bridgedomainname*：指定したブリッジ ドメイン内のすべての DHCP スヌープ クライアントのバインディングをクリアします。
- **clear dhcp ipv4 snoop binding mac-address** *macaddress*：指定した MAC アドレスを持つ DHCP スヌープ クライアントのバインディングをクリアします。

L2VPN コマンド

L2VPN AC で DHCP スヌープをイネーブルにするには、DHCP スヌープ プロファイルをブリッジ ドメインまたは AC に関連付けます。DHCP スヌープの trusted 属性は、DHCP スヌープ プロファイル内の trusted 属性の値に従って AC に設定されます。L2VPN CLI コマンドを使用すると、L2VPN ブリッジ ドメインおよび AC の DHCP スヌープ属性の状態が表示されます。

- **show l2vpn bridge-domain bd-name** *bridgename* **detail**:指定したブリッジ ドメインの L2VPN DHCP スヌープ設定を表示します。
- **show l2vpn forwarding interface** *interface* **detail location** *location*:特定のインターフェイスの L2VPN DHCP スヌープ設定を表示します。

L2Snoop コマンド

L2Snoop は、NETIO と RSP 上の DHCP スヌープ アプリケーションとの間で DHCP スヌープ パケットを送受信します。

show l2snoop statistics pcb all:RSP 上の DHCP スヌープ アプリケーションとの間で送受信された L2SNOOP DHCP パケットの Rx/Tx 統計情報を表示します。

インターフェイス コントローラ コマンド

インターフェイス コントローラは、回線とネットワーク プロセッサとの間で DHCP スヌープ パケットを送受信します。

show controllers *interface* **stats**:回線との間で送受信された DHCP パケットを含むインターフェイス コントローラの統計情報を表示します。

イーサネット フィルタリング（L2PT）

イーサネット フィルタリングは、既存の 2 つのトンネリング プロトコル、シスコの Layer 2 Forwarding（L2F; レイヤ 2 フォワーディング）と Microsoft の Point-to-Point Tunneling Protocol（PPTP）の優れた機能を集約したものです。このセクションの内容は次のとおりです。

パケットがフィルタに基づいて不適切にドロップまたは転送される

ドロップまたは転送されるはずの宛先 MAC を持つパケットが適切にフィルタリングされません。

ステップ 1 `RP/0/RSP0/CPU0:router# show version`

ステップ 2 L2 ブリッジ/xconnect がアップしていることを確認します。

`RP/0/RSP0/CPU0:router# show running-configuration`

アップしていない場合は、VPLS/EoMPLS デバッグ ガイドを参照してください。

回避策

ステップ 1 フィルタリング設定をいったん削除し、実行コンフィギュレーションに再度追加してみます。

ステップ 2 ラインカードを再起動します。

イーサネット運用管理および保守（EOAM）管理性

このセクションの内容は次のとおりです。

検出運用ステータスがローカル/リモートで「Reject」になっている

次のコマンドを使用して、**require remote** パラメータが一方のポートでイネーブルであり、他方ではディセーブルであることを確認します。

- **show ethernet oam discovery**：ネイバー検出ステータスを表示します。
- **show ethernet oam configuration**：設定を表示します。

require remote loopback support パラメータが一方のポートでイネーブルであり、**remote loopback** が他方のポートでディセーブルであることを確認します。

リンク モニタ イベントが設定どおりにトリガーされない

次のコマンドを使用して、リンク モニタリング設定を確認し、関連するエラーを表示します。

- **show ether-ctrl trace configuration**：イーサネット コントローラのリンク モニタリング設定を表示し、ウィンドウ サイズとしきい値が正しいことを確認します。

CCM が MEP で受信されない

次のコマンドを使用して、MEP 設定を確認します。

- **show ethernet cfm local meps**：ローカルの MEP とその CCM 統計情報を表示します。
- **show ethernet cfm peer mep**：CCM が受信されていない場合、ピアは表示されません。
- **show spp sid stats**：SPP SID 統計情報をチェックし、CFM トラフィックが注入およびパントされていることを確認します。
- **show spp client**：RSP から SPP ドロップを探します。
- **debug ethernet cfm packets**:すべてのパケット タイプについてデバッグをイネーブルにします。

MEP で CCM がディセーブルになっている

CFM 継続性チェック メッセージは単方向です。ある MEP で CCM がディセーブルになっている場合、ピア MEP では CCM は受信されず、送信元 MEP のエントリがタイムアウトします。

```
RP/0/RSP0/CPU0:router# show ethernet cfm peer meps
```

「not found」エラーが受信される

次のコマンドを使用して、CFM 設定の確認、統計情報の表示、および接続性のチェックを行います。

- **show run ethernet cfm**：すべての CFM グローバル設定を表示します。
- **show ethernet cfm local meps**：ローカルの MEP とその CCM 統計情報を表示します。
- **show ethernet cfm peer meps**：ピア MEP から受信された CFM CCM を表示します。
- **show ethernet cfm ccm-learning-database**：CCM データベースのエントリを表示します。
- **ping ethernet cfm**：CFM の ping を実行します。

ドロップされたパケット

次のコマンドを使用して、ドロップされたパケットに関する情報を表示します。

- **show ethernet cfm interfaces statistics**：CFM PDU のインターフェイスごとのドロップ統計情報を表示します。
- **show spanning-tree mst**：STP ステータスをチェックします。
- **show ethernet cfm peer meps**:MEP または MIP が設定されたインターフェイス上での CFM ピア MEP ステータスをチェックします。

IGMP スヌーピング（L2 マルチキャスト）

show コマンドの使用

手順概要

1. 正しいトポロジと設定を確認します。
   a. **show l2vpn bridge-domain summary**
   b. **show igmp snooping bridge-domain**
   c. **show l2vpn bridge-domain bd-name** *bd-name*
   d. **show l2vpn bridge-domain bd-name** *bd-name* **detail**
   e. **show igmp snooping bridge-domain** *bd-name* **detail**
   f. **show igmp snooping port bridge-domain** *bd-name*
2. IGMP スヌーピングの制御トラフィックが送受信されていることを確認します。
   a. **show igmp snooping summary statistics**
   b. **show igmp snooping bridge-domain** *bd-name* **detail statistics**
   c. **show igmp snooping port [***if-type* if-name**] detail statistics**

3. IGMP スヌーピングのグループ状態が想定したとおりに作成されていることを確認します。
   a. **show igmp snooping group**
   b. **show igmp snooping group bridge-domain** *bd-name*
   c. **show igmp snooping port** *if-type if-name* **group [detail]**
4. フォワーディング状態が IGMP スヌーピング状態と一致していることを確認します。
   a. **show l2vpn forwarding bridge-domain [***bd-name***] mroute ipv4 location** *lc-name*
   b. **show l2vpn forwarding bridge-domain [***bd-name***] mroute ipv4 hardware [ingress | egress] location** *lc-name*

詳細手順

	コマンドまたはアクション	目的
ステップ 1	a. **show l2vpn bridge-domain summary** b. **show igmp snooping bridge-domain** c. **show l2vpn bridge-domain bd-name** *bd-name* d. **show l2vpn bridge-domain bd-name** *bd-name* **detail** e. **show igmp snooping bridge-domain** *bd-name* **detail** f. **show igmp snooping port bridge-domain** *bd-name*	正しいトポロジと設定を確認します。 a. すべての L2VPN ブリッジ ドメインを一覧表示します。 b. すべてのブリッジ ドメイン内の IGMP スヌーピング状態を表示します。 c. 指定したブリッジ ドメインの情報（インターフェイスと VFI の一覧など）を表示します。 d. 指定したブリッジ ドメインの詳細な情報（IGMP スヌーピング プロファイルなど）を表示します。 e. 指定したブリッジ ドメイン内の詳細な IGMP スヌーピング情報を表示します。 f. 指定したブリッジ ドメイン内の、IGMP スヌーピングの観点から見たポートとポート状態を表示します。
ステップ 2	a. **show igmp snooping summary statistics** b. **show igmp snooping bridge-domain** *bd-name* **detail statistics** c. **show igmp snooping port [***if-type* if-name**] detail statistics**	IGMP スヌーピングの制御トラフィックが送受信されていることを確認します。 a. グローバルなトラフィック統計情報を表示します。 b. ブリッジ ドメイン レベルでのトラフィックを表示します。 c. ポート レベルでのトラフィックを表示します。

	コマンドまたはアクション	目的
ステップ 3	**a. show igmp snooping group** **b. show igmp snooping group bridge-domain** *bd-name* **c. show igmp snooping port** *if-type if-name* **group [detail]**	IGMP スヌーピングのグループ状態が想定したとおりに作成されていることを確認します。 **a.** すべてのブリッジ ドメイン内のグループ状態を表示します。 **b.** 指定したブリッジ ドメイン内のグループ状態を表示します。 **c.** 指定したインターフェイス上のグループ状態を表示します。
ステップ 4	**a. show l2vpn forwarding bridge-domain [***bd-name***] mroute ipv4 location** *lc-name* **b. show l2vpn forwarding bridge-domain [***bd-name***] mroute ipv4 hardware [ingress | egress] location** *lc-name*	フォワーディング状態が IGMP スヌーピング状態と一致していることを確認します。 **a.** 指定したラインカード上の L2FIB でのフォワーディング状態を表示します。 **b.** 指定したラインカード上のハードウェアにインストールされたフォワーディング状態を表示します。

デバッグ コマンド

IGMP スヌーピングのデバッグに役立つコマンドを次に示します。

- **debug l2snoop {call | error | events | init | packet}**
  - **call** : L2snoop 関数呼び出しに関連するデバッグ
  - **error** : L2snoop エラーのデバッグ
  - **events** : L2snoop イベントのデバッグ
  - **init** : L2snoop 初期化のデバッグ
  - **packet** : L2snoop パケット送受信のデコード

トレース コマンド

その他にデバッグに役立つコマンドとして、トレース コマンドがあります。

- **show igmp snooping trace**
  - **all** : IGMP スヌープ トレースをすべて表示します。
  - **error** : IGMP スヌーピング トレースのエラーを表示します。
  - **file** : 特定のファイル。
  - **hexdump** : トレースを 16 進数で表示します。
  - **last** : 最後の <n> 個のエントリを表示します。
  - **location** : カードの位置。
  - **packet-error** : IGMP スヌーピング トレースのパケット エラーを表示します。
  - **reverse** : 最新のトレースから順に表示します。
  - **stats** : 統計情報を表示します。
  - **tailf** : 新しいトレースが追加されるたびにその内容を表示します。
  - **unique** : 重複したエントリを集約してカウントとともに表示します。

- **verbose**：内部デバッグ情報を表示します。
- **wrapping**：ラッピング エントリを表示します。

リンク バンドル

リンク バンドルは、1 つに束ねて単一のリンクとするポートのグループです。リンク バンドルには次のような利点があります。

- 複数の LC にまたがる複数のリンクによって単一のインターフェイスを構成できます。そのため、単一のリンクで障害が発生しても接続性は失われません。
- バンドルされたインターフェイスでは、バンドルの使用可能なすべてのメンバにわたってトラフィックが転送されるため、帯域幅の可用性が向上します。そのため、バンドル内のリンクの 1 つで障害が発生した場合、トラフィックを別のリンクに移動できます。帯域幅を増減する際、パケット フローが中断することはありません。

このセクションの内容は次のとおりです。

バンドルが起動しない

ステップ 1 バンドルがアップしていることを確認します。

```
RP/0/RSP0/CPU0:router# show interface bundle-ether bundle-id
```

ステップ 2 メンバ ポートが「shutdown」でないことを確認します。ポートの MAC アドレス（BIA）が有効であることを確認します。

```
RP/0/RSP0/CPU0:router# show interface
```

ステップ 3 LACP を実行している場合は、LACP パケットを相応に送受信できることを確認します。LACP パケットを相応に送受信できない場合は、インターフェイス カウンタをチェックして、どの段階のパケットがドロップされているかを特定します。

```
RP/0/RSP0/CPU0:router# show lacp counters
```

ステップ 4 LACP 統計情報を表示します。

```
RP/0/RSP0/CPU0:router# show lacp
```

ステップ 5 リンクの相手側がアップしていることを確認します（バンドルおよびメンバ）。

```
RP/0/RSP0/CPU0:router# show bundle
```

バンドル メンバにトラフィックが分散しない

ステップ 1 メンバがアップしていることを確認します。

```
RP/0/RSP0/CPU0:router# show interface node-id
```

ステップ 2 リモート側がアップしていることを確認します。

ステップ 3 LACP パラメータが両側で同じであることを確認します。

a. LACP がイネーブルの場合は、LACP の状態をチェックします。

```
RP/0/RSP0/CPU0:router# show lacp bundle
```

b. バンドル メンバが同じ特性を持つことを確認します。

```
RP/0/RSP0/CPU0:router# show running interface interface-name
```

c. バンドル メンバが異なる特性を持っている場合は、すべてのバンドル メンバの特性を同じにします。

d. LACP パケットが送受信されていることを確認します。

```
RP/0/RSP0/CPU0:router# debug bundlemgr local packets port node-id
```

回避策

LACP によるバンドルが起動しない場合は、バンドルの一方の側をパッシブ モードにし、もう一方の側をアクティブ モードにします。少なくとも一方の側がアクティブである必要があります。

バンドルでバックプレーンからの MAC アドレスが使用されない

ステップ 1 バックプレーン MAC がプログラムされていることを確認します。

```
RP/0/RSP0/CPU0:router(admin)# show diag chassis eeprom-info
```

このコマンドは管理モードで実行する必要があります。

ステップ 2

```
RP/0/RSP0/CPU0:router# show controllers backplane bpe-trace
```

L3 データ トラフィックが流れない

通常インターフェイス（サブインターフェイスなし）

ステップ 1 Address Resolution Protocol（ARP; アドレス解決プロトコル）を表示します。

```
RP/0/RSP0/CPU0:router# show arp
```

ステップ 2 LAG テーブルがハードウェアに適切にプログラムされていることを確認します。

```
RP/0/RSP0/CPU0:router# show interface bundle-ether bundle-id
```

ステップ 3 実行コンフィギュレーション情報を確認します。

```
RP/0/RSP0/CPU0:router# show running-config
```

ステップ 4 Cisco Express Forwarding（CEF; シスコ エクスプレス フォワーディング）によって転送されたパケットに関する情報を表示します。

```
RP/0/RSP0/CPU0:router# show cef
```

ステップ 5 `RP/0/RSP0/CPU0:router# show cef hardware ingress location node-id`

ステップ 6 `RP/0/RSP0/CPU0:router# show cef hardware egress location node-id`

サブインターフェイス

ステップ 1 L3 IPv4 トラフィックのトラブルシューティングを行います。

ステップ 2 VLAN 着信トラフィックが受信インターフェイス上の VLAN トラフィックと一致することを確認します。

バンドル上での ping が失敗する

ステップ 1 ARP を表示します。

```
RP/0/RSP0/CPU0:router# show arp
```

ステップ 2 特定の LC または RSP での ARP 情報を表示します。

```
RP/0/RSP0/CPU0:router# show arp location node-id
```

ステップ 3 `RP/0/RSP0/CPU0:router# show cef hardware detail location node-id ingress`

ステップ 4 `RP/0/RSP0/CPU0:router# show interface`

ステップ 5 ハッシュ計算機能を使用して、テストするバンドル メンバ（インターフェイス）を特定します。

ステップ 6 該当インターフェイスをバンドルから削除します。

ステップ 7 該当インターフェイスに IP アドレスを割り当てます。

ステップ 8 該当インターフェイスに対して ping を実行します。

ステップ 9 ルータと ping 先のノードとの間で ARP が解決されていることを確認します。

ステップ 10 相手側の ARP テーブル内の MAC アドレスがルータ上の MAC アドレスに対応していることを確認します。

ステップ 11 バンドルの MAC アドレスが有効であることを確認します。

ステップ 12 ルーティングおよびハードウェア ルーティング テーブルにネクストホップへのエントリがあることを確認します。

ステップ 13 インターフェイス カウンタをチェックし、ping パケットが送信され、バンドルのルータ メンバ ポートで受信されているかどうかを確認します。

ステップ 14 ucode カウンタをチェックし、パケットがバンドルの受信メンバまたは送信メンバ上でドロップされている箇所を確認します。

回避策

別のポートで試してみます。

L3 パケットがバンドル上で同期しない

ステップ 1 RP/0/RSP0/CPU0:router# **show interface**

ステップ 2 そのプロトコルのデバッグを有効にするか、プロトコル カウンタをチェックして、プロトコル パケットが送受信されているかどうかを確認します。

ステップ 3 プロトコル パケットが送受信されていない場合は、インターフェイス カウンタをチェックして、インターフェイスでパケットの入出力が示されているかどうかを確認します。

ステップ 4 インターフェイス レベルではパケットの受信と送信が示されているにもかかわらず、プロトコルにパケットが到達していない場合は、ucode カウンタをチェックして、ドロップが起こっているかどうかを確認します。

L2 トラフィックが流れない

VPLS

ステップ 1 AC がアップしていることを確認します。

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain
```

ステップ 2 ブリッジ ドメインがアップしていることを確認します。

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain
```

ステップ 3 MTU の不一致を探します。

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain detail
```

VPWS

ステップ 1 設定されている相互接続に関する簡潔な情報を表示します。

```
RP/0/RSP0/CPU0:router# show l2vpn xconnect summary
```

ステップ 2
```
RP/0/RSP0/CPU0:router# show l2vpn xconnect state
```

ステップ 3
```
RP/0/RSP0/CPU0:router# show controllers bundle bundle-ether bundle-id location node-id
```

バンドル上でのロード バランシング

L2 トラフィックがロード バランシングされない

ステップ 1
```
RP/0/RSP0/CPU0:router# show running-config
```

ステップ 2
```
RP/0/RSP0/CPU0:router# show bundle
```

ステップ 3
```
RP/0/RSP0/CPU0:router# show interface bundle-ether bundle-id
```

ステップ 4
```
RP/0/RSP0/CPU0:router# show arm router-ids
```

ステップ 5
```
RP/0/RSP0/CPU0:router# show controllers bundle bundle-ether bundle-id location node-id
```

L3 トラフィックがロード バランシングされない

ステップ 1
```
RP/0/RSP0/CPU0:router# show arm router-id
```

ステップ 2
```
RP/0/RSP0/CPU0:router# show bundle
```

ステップ 3
```
RP/0/RSP0/CPU0:router# show interface bundle-ether bundle-id
```

ステップ 4
```
RP/0/RSP0/CPU0:router# show controllers bundle bundle-ether bundle-id location node-id
```

ステップ 5
```
RP/0/RSP0/CPU0:router# show running-config
```

ステップ 6
```
RP/0/RSP0/CPU0:router# show arm router-ids
```

ステップ 7 トラフィックを送出するのが望ましいメンバを特定します。

```
RP/0/RSP0/CPU0:router# bundle-hash bundle-ether bundle-id
```

ステップ 8 バンドルの各メンバを表示し、実際にトラフィックを送出しているメンバを確認します。

```
RP/0/RSP0/CPU0:router# show interface
```

バンドルの統計情報

現在のリリースでは、バンドル インターフェイスに対する **show interface accounting** コマンドで L2 統計情報はサポートされていません。

多重スパニング ツリー（MST）

Multiple Spanning Tree（MST; 多重スパニング ツリー）は、シスコ独自の Multiple Instances Spanning Tree Protocol（MISTP）の実装を参考にして IEEE で策定された標準規格です。このセクションの内容は次のとおりです。

show および debug コマンドの使用方法

手順概要

1. **show spanning-tree mst**
2. **show spanning-tree ring-termination**
3. **show l2vpn bridge-domain [bd-name** *bridge-domain name* **| brief | detail | group** *bridge-domain group name* **| interface {***type interface-id***} | neighbor** *IP address* **[pw-id** *value***] | summary]**
4. **debug spanning-tree mst controller**
5. **debug spanning-tree mst io**
6. **debug spanning-tree mst packet**
7. **debug spanning-tree mst protocol-state**

詳細手順

	コマンドまたはアクション	目的
ステップ 1	**show spanning-tree mst**	MST ステータスを表示します。結果をフィルタリングするには、次のパラメータとキーワードを使用します。 • **name**：プロトコル インスタンス名。 • **blocked-ports**：ブロック ポートに関する MST 情報。 • **bpdu interface** *interface-name*：MST の Bridge Protocol Data Unit（BPDU; ブリッジ プロトコル データ ユニット）。 • **brief**：MST 情報の概要。 • **configuration**：MST 関連の設定情報。 • **errors**：MST 設定エラー。 • **instance-id**：MST インスタンス。 • **interface** *interface-name*：インターフェイスごとの MST 情報。 • **internal**：MST 内部情報。
ステップ 2	**show spanning-tree ring-termination**	Spanning Tree Protocol（STP; スパニング ツリー プロトコル）がイネーブルであることを確認します。結果をフィルタリングするには、次のパラメータとキーワードを使用します。 • **name**：プロトコル インスタンス名。 • **bpdu interface** *interface-name*：MST の BPDU。 • **interface** *interface-name*：インターフェイスごとの MST 情報。
ステップ 3	**show l2vpn bridge-domain [bd-name** *bridge-domain name* | **brief** | **detail** | **group** *bridge-domain group name* | **hardware** | **interface {***type interface-path-id***}** | **neighbor** *IP address* **[***pw-id value***]** | **summary]** **例：** `RP/0/RSP0/CPU0:router# show l2vpn bridge-domain`	特定のブリッジ ドメインのブリッジ ポートに関する情報（接続回線や疑似回線など）を表示します。結果をフィルタリングするには、次のパラメータとキーワードを使用します。 • **bd-name**：ブリッジ ドメイン名。 • **brief**：簡潔な情報。 • **detail**：詳細な情報。 • **group**：ブリッジ ドメイン グループ名。 • **hardware**：ハードウェア情報。 • **interface**：インターフェイスによるフィルタ。 • **neighbor**：ネイバーによるフィルタ。 • **summary**：ブリッジ ドメインの概要情報。

	コマンドまたはアクション	目的
ステップ 4	`debug spanning-tree mst controller`	• **all**：すべてのモジュールをデバッグします。 • **comms**：通信モジュールをデバッグします。 • **config**：設定モジュールをデバッグします。 • **database**：データベース モジュールをデバッグします。 • **edm**：EDM モジュールをデバッグします。 • **engine**：エンジン モジュールをデバッグします。 • **l2vpn**：L2VPN モジュールをデバッグします。 • **pfi**：PFI モジュールをデバッグします。 • **rl2gp**：RL2GP モジュールをデバッグします。
ステップ 5	`debug spanning-tree mst io`	• **all**：すべてのモジュールをデバッグします。 • **comms**：通信モジュールをデバッグします。 • **config**：設定モジュールをデバッグします。 • **database**：データベース モジュールをデバッグします。 • **edm**：EDM モジュールをデバッグします。 • **engine**：エンジン モジュールをデバッグします。 • **main**：初期化/クリーンアップをデバッグします。 • **packet-io**：パケット入出力をデバッグします。 • **pfi**：PFI モジュールをデバッグします。 • **rl2gp**：RL2GP モジュールをデバッグします。
ステップ 6	`debug spanning-tree mst packet`	• **brief**：簡潔な出力を表示します。 • **full**：完全な出力を表示します。 • **raw**：生パケット出力を表示します。
ステップ 7	`debug spanning-tree mst protocol-state`	特定の MSTI インターフェイスのデバッグを表示します。

MSTP の構成が不適切または一貫していない

スパニング ツリーが適切に構成されない場合、その原因の多くは設定ミスか、BPDU の損失です。これは通常、複数のノードが自身をルートとして提示する現象として現れますが、結果的にどのノードがルートであるかについて不一致が見られる場合もあります。

MSTP の構成が不適切または一貫していない：設定ミス

各ノードで次が一致していることを確認します。

- 設定名
- ブリッジのリビジョン
- プロバイダーブリッジ モード
- インスタンスと VLAN のマッピング

```
RP/0/RSP0/CPU0:router# show run spanning-tree mst protocol instance name
```

MSTP の構成が不適切または一貫していない：BPDU の損失

ノード A が ノード B に BPDU を送信しているかどうかを確認します。それらのノードを接続しているインターフェイスごとに次のコマンドを数回実行します。

```
RP/0/RSP0/CPU0:router# show spanning-tree mst protocol instance name internal io interfaces interface-name
```

定期的に BPDU を送信するのは指定ポートだけです。非指定ポートは、トポロジの変更時または起動時にアップデートを送信します。送受信された BPDU が適宜上位に送られていることを確認します。

MSTP は適切に構成されているが、トラフィックのフラッディングが発生する

BPDU の断続的な損失は、show コマンドでスパニングツリーが不適切に見えないにもかかわらず、トポロジ変更通知が送出されることを意味する場合があります。これらの通知は MAC のフラッシュを引き起こし、その結果、MAC アドレスが再学習されるまで強制的にトラフィックのフラッディングが起こります。

トポロジ変更通知を探します。次のコマンドを実行して、TC 1 を探します。

（注） このオプションは冗長です。

```
RP/0/RSP0/CPU0:router# debug spanning-tree mst packet full {received | sent}
```

同じコマンドを、出力形式を「brief」にして両方のノードで実行し、失われた BPDU をチェックします。タイムスタンプをチェックし、間隔が 6 秒以上開いている箇所を探します。間隔が 6 秒以上開くと、トポロジ変更が起こります。

```
RP/0/RSP0/CPU0:router# debug spanning-tree mst packet brief {received | sent}
```

MSTP で間違ったポート状態が示される

STP は、ポート状態を変更しようとするときに L2VPN を使用します。「Sent Update」の値をチェックします。この値が Yes の場合、STP は L2VPN からのアップデートを待っています。

```
RP/0/RSP0/CPU0:router# show spanning-tree mst name internal l2vpn
```

パケット フォワーディングが MSTP 状態と一致していない

ステップ 1 冗長リンクをシャットダウンし、MSTP 設定を削除して、基本的なブリッジ処理が動作することを確認します。

```
RP/0/RSP0/CPU0:router# show spanning-tree mst name
RP/0/RSP0/CPU0:router# show interface interface-name
```

ステップ 2 MSTP によって計算された各ポートの状態をチェックし、それを MSTP によって制御されているポートおよび EFP 上でのパケット送受信数と比較します。通常のデータ パケットは、フォワーディング（FWD）状態のポートだけで送受信されます。BPDU は、MSTP によって制御されているすべてのポートで送受信されます。

ステップ 3 BPDU が流れていて、ルート ブリッジの選択が正しいことを確認します。これらの関連するシナリオを最初にチェックします。

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain [detail]
```

このコマンドは、ブリッジ ドメインのメンバのステータスを表示します。関連するブリッジ ドメインのメンバがアップしていることを確認します。

ステップ 4 ハードウェアにプログラムされたフォワーディング状態をチェックします。

RL2GP アクセス ネットワークが RL2GP ノードをルートとして認識しない

RL2GP を設定する方法には次の 2 通りがあります。

- あたかも両方のノードが別々であるかのようにアドバタイズする。この方法では、各ノードが一意のブリッジ ID を持ち、設定が互いを補完する必要があります。
- あたかも各ノードが同じノード上の異なるポートであるかのようにアドバタイズする。この方法では、ポート ID 以外の設定は同一です。

RL2GP のコマンドは、基本インターフェイスではなく、タグ付けされていない EFP を対象とする必要があります。

ステップ 1

```
RP/0/RSP0/CPU0:router# show running-config spanning-tree ring-termination name
```

ステップ 2 両方のノードをデバッグし、出力を含めます。

```
RP/0/RSP0/CPU0:router# debug spanning-tree mst packet full sent interface interface-name
```

トラフィックが RL2GP ノードを通じてスイッチングされない

ステップ 1 l2vpn と UIDB のデータを収集し、データパスが正しいことを確認します。

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain [detail]
```

ステップ 2 フォワーディング状態がハードウェアにプログラムされたとおりに設定されていることを確認します。

Virtual Private Local Area Network Service（VPLS）

Virtual Private Local Area Network Service（VPLS）を使用すると、サイトを地理的に分離し、各サイトを疑似回線で接続することによってイーサネット ブロードキャスト ドメインを共有できます。このセクションの内容は次のとおりです。

show および debug コマンドの使用方法

手順概要

1. **show l2vpn bridge-domain [bd-name** *bridge-domain name* **| brief | detail | group** *bridge-domain group name* **| interface** {*type interface-id*} **| neighbor** *IP address* **[pw-id** *value*] **| summary]**
2. **show l2vpn forwarding bridge-domain [** *bridge-domain-name*] **{detail | hardware {egress | ingress}} {location** *node-id*}

詳細手順

	コマンドまたはアクション	目的
ステップ 1	**show l2vpn bridge-domain [bd-name** *bridge-domain name* | **brief** | **detail** | **group** *bridge-domain group name* | **interface** *type* | **neighbor** *IP address* **[pw-id value]** | **summary]** **例：** `RP/0/RSP0/CPU0:router# show l2vpn bridge-domain bd-name d1`	ブリッジドメインのブリッジポートを表示します（AC および PW）。結果をフィルタリングするには、次のパラメータとキーワードを使用します。 • **bd-name** *bridge-domain name*：（任意）ブリッジ ID に従ってブリッジを表示します。bridge-domain name 引数では、ブリッジ ドメインの名前を指定します。 • **brief**：（任意）ブリッジの簡潔な情報を表示します。 • **detail**：（任意）Layer 2 VPN（L2VPN; レイヤ 2 VPN）の出力を表示し、MAC 回収機能がイネーブルかどうか、および疑似回線から送受信された MAC 回収メッセージの数を示します。 • **group** *bridge-domain group name*：（任意）ブリッジドメイン グループ名でのフィルタ情報を表示します。bridge-domain group name 引数では、ブリッジ ドメイン グループの名前を指定します。 • **interface**：（任意）ブリッジ ドメイン上のインターフェイスでのフィルタ情報を表示します。 • *type*：インターフェイス タイプ。 • *interface-id*：物理インターフェイスまたは仮想インターフェイスを識別します。 • **neighbor** *IP address*：（任意）ネイバーのフィルタに一致する疑似回線を含むブリッジ ドメインだけを表示します。IP address 引数では、ネイバーの IP アドレスを設定します。 • **pw-id** *value*：（任意）疑似回線 ID でのフィルタ情報を表示します。範囲は 1 ～ 4294967295 です。 • **summary**：（任意）ブリッジ ドメインの概要情報を表示します。
ステップ 2	**show l2vpn forwarding bridge-domain** **例：** `RP/0/RSP0/CPU0:router# show l2vpn forwarding bridge-domain ABC mac-address interface Gi0/1/2/1.2 detail hardware location 0/4/CPU0 Bridge`	フォワーディング ブリッジ ドメインの情報を表示します。結果をフィルタリングするには、次のパラメータとキーワードを使用します。 • *bridge-domain-name*：（任意）ブリッジ ドメインの名前。 • **detail**：接続回線と疑似回線に関する詳細情報をすべて表示します。 • **hardware**：ハードウェア ロケーション エントリを表示します。 • **egress**：出力 PSE から情報を読み取ります。 • **ingress**：入力 PSE から情報を読み取ります。 • **location** *node-id*：指定したロケーションのブリッジドメイン情報を表示します。

AC がダウンしている

ステップ 1 `RP/0/RSP0/CPU0:router# show interface`

ステップ 2 `RP/0/RSP0/CPU0:router# show l2vpn bridge interface detail`

ステップ 3 AC インターフェイスで l2transport が設定されていることを確認します。

ステップ 4 AC インターフェイスがアップしていることを確認します。

ステップ 5 MTU が一致していることを確認します。

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain interface type interface-name detail
```

疑似回線がダウンしている

ステップ 1 OSPF ネイバー情報を表示します。

```
RP/0/RSP0/CPU0:router# show ospf neighbor
```

ステップ 2 MPLS LDP ネイバー情報を表示します。

```
RP/0/RSP0/CPU0:router# show mpls ldp neighbor neighbor
```

ステップ 3 ブリッジの疑似回線の状態を表示します。

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain neighbor
```

ステップ 4 両方の PE で疑似回線が適切に設定されていることを確認します。

ステップ 5 MPLS パッケージがインストールされていることを確認します。

ステップ 6 コア インターフェイスがアップしていることを確認します。

ステップ 7 ルーティング プロトコルが OSPF であることを確認します。

ステップ 8 PE ピアとの LDP セッションが確立されていることを確認します。

ステップ 9 MTU が一致していることを確認します。

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain detail
```

VPLS がフラッディング トラフィックを転送しない

ステップ 1 OSPF ネイバー情報を表示します。

```
RP/0/RSP0/CPU0:router# show ospf neighbor
```

ステップ 2 MPLS LDP ネイバー情報を表示します。

```
RP/0/RSP0/CPU0:router# show mpls ldp neighbor neighbor
```

ステップ 3 両方の PE で疑似回線が適切に設定されていることを確認します。

ステップ 4 MPLS パッケージがインストールされていることを確認します。

ステップ 5 コア インターフェイスがアップしていることを確認します。

ステップ 6 ルーティング プロトコルが OSPF であることを確認します。

ステップ 7 PE ピアとの LDP セッションが確立されていることを確認します。

ステップ 8 MTU が一致していることを確認します。

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain detail
```

VPLS が AC から疑似回線にフラッディング トラフィックを転送しない

ステップ 1 セグメントの入力 UIDB および XID を表示します。

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding interface hardware ingress detail location
```

ステップ 2 疑似回線のハードウェア情報を表示します。

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding neighbor 192.12.12.5 pw-id 100 hardware
egress location node-id0
```

ステップ 3 MPLS リーフ情報を表示します。

```
RP/0/RSP0/CPU0:router# show mpls forwarding labels hardware egress detail location
```

ステップ 4 ブロードキャスト、マルチキャスト、および未知のユニキャストに関するブリッジ情報を表示します。

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain bd-name 1 det
```

ステップ 5 MAC 制限を超えていないことを確認します。

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding bridge-domain 1:1 detail location
```

ステップ 6 PI イベント トレースを表示します。

```
RP/0/RSP0/CPU0:router# show l2vpn trace location
```

ステップ 7 疑似回線と AC がアップしていることを確認します。

ステップ 8 両方の AC に対してハードウェアがプログラムされていることを確認します。

ステップ 9 `RP/0/RSP0/CPU0:router# show l2vpn forwarding interface GigabitEtherne0/5/0/2 hardware ingress detail location` *node-id*

ステップ 10 疑似回線に対してハードウェアがプログラムされていることを確認します。

VPLS が疑似回線から AC にフラッディング トラフィックを転送しない

ステップ 1 セグメントの入力 UIDB および XID を表示します。

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding interface hardware ingress detail location
```

ステップ 2 MPLS リーフ情報を表示します。

```
RP/0/RSP0/CPU0:router# show mpls forwarding labels hardware egress detail location
```

ステップ 3 ブロードキャスト、マルチキャスト、および未知のユニキャストに関するブリッジ情報を表示します。

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain bd-name 1 det
```

ステップ 4 MAC 制限を超えていないことを確認します。

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding bridge-domain 1:1 detail location
```

ステップ 5

ステップ 6 PI イベント トレースを表示します。

```
RP/0/RSP0/CPU0:router# show l2vpn trace location
```

ステップ 7 疑似回線と AC がアップしていることを確認します。

ステップ 8 両方の AC に対してハードウェアがプログラムされていることを確認します。

ステップ 9 `RP/0/RSP0/CPU0:router# show l2vpn forwarding interface GigabitEtherne0/5/0/2 hardware ingress detail location` *node-id*

ステップ 10 疑似回線に対してハードウェアがプログラムされていることを確認します。

VPLS が AC から AC にユニキャスト トラフィックを転送しない

ステップ 1 ブリッジ ドメインの状態を表示します。

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain bd-name node-id detail
```

ステップ 2 MAC 情報を表示します。

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding bridge-domain mac-address location
```

ステップ 3 両方の AC に対してハードウェアがプログラムされていることを確認します。

ステップ 4 LC の宛先インターフェイスに対して宛先 MAC エントリがプログラムされていることを確認します。

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding bridge-domain mac-address location node-id0
```

VPLS が AC から疑似回線にユニキャスト トラフィックを転送しない

ステップ 1 ブリッジ ドメインの状態を表示します。

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain bd-name node-id detail
```

ステップ 2 MAC 情報を表示します。

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding bridge-domain mac-address location
```

ステップ 3 AC と疑似回線の両方に対してハードウェアがプログラムされていることを確認します。

ステップ 4 LC の宛先インターフェイスに対して宛先 MAC エントリがプログラムされていることを確認します。

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding bridge-domain mac-address location node-id0
```

VPLS が疑似回線から AC にフラッディング トラフィックを転送しない

ステップ 1 ブリッジ ドメインの状態を表示します。

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain bd-name node-id detail
```

ステップ 2 AC と疑似回線の両方に対してハードウェアがプログラムされていることを確認します。

疑似回線はアップしているのに ping が失敗する

ステップ 1 ブリッジ ドメインの状態を表示します。

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain bd-name node-id detail
```

ステップ 2 両方の CE が同じサブネット上にあることを確認します。

ステップ 3 MTU が一致していることを確認します。

ステップ 4 エンドツーエンドのカプセル化が一致していることを確認します。

トラフィックが失われる

ステップ 1 ブリッジ ドメインの状態を表示します。

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain bd-name node-id detail
```

ステップ 2 セグメント カウンタを表示し、交換されたパケットおよびバイト数が増えているかどうかを確認します。

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding interface GigabitEthernet node-id detail
location node-id
```

ステップ 3 帯域幅レートが CE 間で一致していることを確認します。

疑似回線のフラップが原因でトラフィックが失われる

ステップ 1 ブリッジ ドメインの状態を表示します。

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain bd-name node-id detail
```

ステップ 2 セグメント カウンタを表示し、交換されたパケットおよびバイト数が増えているかどうかを確認します。

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding interface GigabitEthernet node-id detail
location node-id
```

ステップ 3 PI イベント トレースを表示します。

```
RP/0/RSP0/CPU0:router# show l2vpn trace location
```

RSP フェールオーバー中にトラフィックが失われる

ステップ 1 xconnect の状態を表示します。

```
RP/0/RSP0/CPU0:router# show l2vpn bridge detail
```

ステップ 2 セグメントのカウンタを表示します。

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding interface GigabitEthernet node-id detail
location node-id
```

ステップ 3 ブリッジ ドメインの状態を表示します。

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain bd-name detail
```

ステップ 4 入力 UIDB を表示します。

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding interface interface hardware ingress detail
location node-id
```

ステップ 5 MPLS パス内のすべてのルータをチェックし、次が設定されていることを確認します。

a. MPLS LDP グレースフル リスタート

b. OSPF NSF

ステップ 6 セグメント カウンタを表示し、交換されたパケットおよびバイト数が増えているかどうかを確認します。

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding interface GigabitEthernet node-id detail
location node-id
```

優先パスが機能しない

ステップ 1 ブリッジ ドメインの状態を表示します。

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain bd-name detail
```

ステップ 2 入力 UIDB を表示します。

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding interface interface hardware ingress detail
location node-id
```

Virtual Private Wire Services（VPWS）

Virtual Private Wire Services（VPWS）は、基盤となる MPLS トンネルを使用して地理的に離れたサイト間の回線セットをエミュレートすることにより、これらのサイトを接続します。このセクションの内容は次のとおりです。

show および debug コマンドの使用方法

手順概要

1. **show l2vpn xconnect [detail | group | interface | neighbor | state | summary | type | state unresolved]**
2. **show l2vpn forwarding {detail | hardware | interface | location | message | resource | summary | unresolved} location** *node-id*
3. **show mpls forwarding [detail | {label** *label number***} | interface** *interface-id* **| labels value | location | prefix [***network/mask* | *length***] | summary | tunnels** *tunnel-id***]**

詳細手順

	コマンドまたはアクション	目的
ステップ 1	**show l2vpn xconnect [detail** | **group** | **interface** | **neighbor** | **state** | **summary** | **type** | **state unresolved]** **例：** `RP/0/RSP0/CPU0:router# show l2vpn xconnect`	設定されている相互接続に関する簡潔な情報を表示します。結果をフィルタリングするには、次のパラメータとキーワードを使用します。 • **detail**：詳細な情報。 • **group**：指定したグループ内のすべての相互接続。 • **interface**：インターフェイスとサブインターフェイス。 • **neighbor**：ネイバー。 • **state**：xconnect の状態のタイプ（アップ、ダウン）。 • **summary**：AC マネージャ データベースからの AC 情報。 • **type**：xconnect のタイプ（ac-pw、ローカル スイッチング）。 • **state unresolved**：未解決の相互接続。
ステップ 2	**show l2vpn forwarding {detail** | **hardware** | **interface** | **location** | **message** | **resource** | **summary** | **unresolved} location** *node-id* **例：** `RP/0/RSP0/CPU0:router# show l2vpn forwarding location 0/2/cpu0`	対応する AC サブインターフェイスを表示します。
ステップ 3	**show mpls forwarding [detail** | **{label** *label number***}** | **interface** *interface-id* | **labels value** | **location** | **prefix** *[network/mask* | **length]** | **summary** | **tunnels** *tunnel-id***]**	MPLS Label Forwarding Information Base（LFIB; ラベル転送情報ベース）エントリとローカル ラベルの範囲を表示します。

AC がダウンしている

ステップ 1 インターフェイスの状態を表示します。

```
RP/0/RSP0/CPU0:router# show interface
```

ステップ 2 xconnect の状態を表示します。

```
RP/0/RSP0/CPU0:router# show l2vpn xconnect detail
```

ステップ 3 AC インターフェイスで l2transport が設定されていることを確認します。

ステップ 4 AC インターフェイスがアップしていることを確認します。

ステップ 5 MTU が一致していることを確認します。

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain interface type interface-name detail
```

疑似回線がダウンしている

ステップ 1 OSPF ネイバー情報を表示します。

```
RP/0/RSP0/CPU0:router# show ospf neighbor
```

ステップ 2 MPLS LDP ネイバー情報を表示します。

```
RP/0/RSP0/CPU0:router# show mpls ldp neighbor neighbor
```

ステップ 3 ブリッジの疑似回線の状態を表示します。

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain neighbor
```

ステップ 4 両方の PE で疑似回線が適切に設定されていることを確認します。

ステップ 5 MPLS パッケージがインストールされていることを確認します。

ステップ 6 コア インターフェイスがアップしていることを確認します。

ステップ 7 ルーティング プロトコルが OSPF であることを確認します。

ステップ 8 PE ピアとの LDP セッションが確立されていることを確認します。

ステップ 9 MTU が一致していることを確認します。

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain detail
```

VPWS が AC から疑似回線にトラフィックを転送しない

ステップ 1 疑似回線のハードウェア情報を表示します。

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding neighbor 192.12.12.5 pw-id 100 hardware egress location node-id0
```

ステップ 2 ブロードキャスト、マルチキャスト、および未知のユニキャストに関するブリッジ情報を表示します。

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain bd-name 1 det
```

ステップ 3 MAC 制限を超えていないことを確認します。

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding bridge-domain 1:1 detail location
```

ステップ 4 疑似回線と AC がアップしていることを確認します。

ステップ 5 両方の AC に対してハードウェアがプログラムされていることを確認します。

ステップ 6 `RP/0/RSP0/CPU0:router# show l2vpn forwarding interface GigabitEtherne0/5/0/2 hardware ingress detail location` *node-id*

ステップ 7 疑似回線に対してハードウェアがプログラムされていることを確認します。

VPWS が疑似回線から AC にトラフィックを転送しない

ステップ 1 入力 UIDB を表示します。

`RP/0/RSP0/CPU0:router# show l2vpn forwarding interface` *interface* `hardware ingress detail location` *node-id*

ステップ 2 疑似回線と AC がアップしていることを確認します。

ステップ 3 この疑似回線に関連付けられたローカル ラベルを探します。

```
RP/0/RSP0/CPU0:router# show l2vpn xconnect detail
```

ステップ 4 AC に関連付けられた XID を探します。

```
RP/0/RSP0/CPU0:router# show l2vpn xconnect detail
```

ステップ 5 `RP/0/RSP0/CPU0:router# show l2vpn forwarding interface GigabitEthernet 0/4/0/2 hardware ingress detail location 0/4/CPU0`

疑似回線はアップしているのに ping が失敗する

ステップ 1 ブリッジ ドメインの状態を表示します。

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain bd-name node-id detail
```

ステップ 2 両方の CE が同じサブネット上にあることを確認します。

ステップ 3 MTU が一致していることを確認します。

ステップ 4 エンドツーエンドのカプセル化が一致していることを確認します。

トラフィックが失われる

ステップ 1 ブリッジ ドメインの状態を表示します。

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain bd-name node-id detail
```

ステップ 2 セグメント カウンタを表示し、交換されたパケットおよびバイト数が増えているかどうかを確認します。

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding interface GigabitEthernet node-id detail
location node-id
```

ステップ 3 帯域幅レートが CE 間で一致していることを確認します。

RSP フェールオーバー中にトラフィックが失われる

RSP フェールオーバーが実行されるときにトラフィックが失われることがあります。これは、プレフィクスの学習に使用される IGP がダウンすることに起因する可能性があります。次の説明は、IGP として OSPF を使用していることを前提とします。

- **show process failover**：フェールオーバー中のプロセスの詳細を表示します。
- **debug ospf ha**：OSPF HA 関連のデバッグをイネーブルにします。
- **debug ospf** *instance* **nsf**：フェールオーバーの前に実行し、デバッグ ログを収集します。
- **show process failover**：フェールオーバーの後に実行します。

ステップ 1 すぐにチェックすべきこととして、ネクストホップ ルータでも（このルータと同様に）フェールオーバー メカニズムが作動したかどうかが挙げられます。ネクストホップ ルータでもフェールオーバー メカニズムが作動した場合は、OSPF がダウンしている可能性があります。

ステップ 2 ネクストホップ ルータでフェールオーバー メカニズムが作動していない場合は、OSPF で「nsf cisco」が設定されていることを確認します。「nsf cisco」が設定されている場合は、フェールオーバー中にネクストホップに到達可能かどうかを確認します。到達不能な場合は、リンクのダウンやネゴシエーションの問題など、到達可能性に関する問題がある可能性があります。

優先パスが機能しない

ステップ 1 ブリッジ ドメインの状態を表示します。

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain bd-name detail
```

ステップ 2 入力 UIDB を表示します。

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding interface interface hardware ingress detail
location node-id
```

L3 機能

インターネット プロトコル（IP）

このセクションの内容は次のとおりです。

show および debug コマンドの使用方法

手順概要

1. **show cef location** *node-id*
2. **show cef ipv4** {*prefix/mask*} **location** *node-id*
3. **show bgp summary**
4. **show bgp [{ipv4 | all} {unicast | multicast | all}] dampened-paths**
5. **show bgp [{ipv4 | all} {unicast | multicast | all}] flap-statistics [regexp** *regular-expression* | *filter-list access-list* | *cidr-only* | {*ip-address* [{*mask* | */prefix-length*} [*longer-prefixes*]] **[detail]}]**
6. **show arp [vrf** *vrf-name*] [*ip-address* **[location** *node-id*] | *hardware-address* **[location** *node-id]* | **traffic [location** *node-id* | *interface-name*]

7. **show interface accounting**
8. **show cef ipv4** *{prefix/mask}* **hardware {ingress | egress} location** *node-id*
9. **show cef platform trace common [all | errors | events | info] [location** *node-id***]**

詳細手順

	コマンドまたはアクション	目的
ステップ 1	**show cef location** *node-id* **例：** `RP/0/RSP0/CPU0:router# show cef location 0/2/CPU0`	Line Card（LC; ラインカード）上での Cisco Express Forwarding（CEF; シスコ エクスプレス フォワーディング）の IPv4 ルートをすべて表示します。 **（注）** これは、ルートの数が少ない場合にだけ使用します。
ステップ 2	**show cef ipv4 {***prefix/mask***} location** *node-id* **例：** `RP/0/RSP0/CPU0:router# show cef ipv4 192.1.1.1/32 location 0/2/CPU0`	LC 上でのプレフィクスのルートを表示します。
ステップ 3	**show bgp summary**	Border Gateway Protocol（BGP; ボーダー ゲートウェイ プロトコル）ネイバーを、各アクティブ アドレス ファミリの着信および発信ポリシーなしで表示します。 **（注）** これは、ルートの数が多い場合に使用します。
ステップ 4	**show bgp [{ipv4 | all} {unicast | multicast | all}] dampened-paths** **例：** `RP/0/RSP0/CPU0:router# show bgp dampened-paths`	ダンプニングがイネーブルになっているルートを表示します。
ステップ 5	**show bgp flap-statistics [***ip-address***[/***mask***]]** **例：** `RP/0/RSP0/CPU0:router# show bgp flap-statistics`	BGP のフラップ統計情報を表示します。 **（注）** これは、ダンプニングがイネーブルになっているルートに対して使用します。 引数またはキーワードを指定しない場合は、アドレス ファミリのすべてのルートが表示されます。 IP アドレスを入力する際、マスクまたはプレフィクス長を指定しなかった場合は、最長一致プレフィクスが表示されます。
ステップ 6	**show arp [vrf** *vrf-name***] [***ip-address* **[location** *node-id***] | traffic [location** *node-id***]** **例：** `RP/0/RSP0/CPU0:router# show arp`	Address Resolution Protocol（ARP; アドレス解決プロトコル）レコードを表示します。 バンドルおよび VLAN-on-Bundle インターフェイスでは、**location** *node-id* を入力します。これは、どのキャッシュ エントリを表示するかをシステムに知らせます。

	コマンドまたはアクション	目的
ステップ 7	**show interface accounting** [*location*] **例：** RP/0/RSP0/CPU0:router# show interface accounting location 0/4/CPU0	インターフェイス上でのパケット アカウンティングをプロトコルごとに表示します。
ステップ 8	**show cef ipv4** {*prefix/mask*} **hardware** {**ingress** | **egress**} **location** *node-id* **例：** RP/0/RSP0/CPU0:router# show cef ipv4 38.1.1.2/32 hardware egress location 0/4/CPU0	LC のハードウェアに設定された IPv4 プレフィクス/ルートを表示します。 この情報は、宛先の IP またはプレフィクス アクションが COMPLETE、PUNT、DROP のいずれであるかを確認するのに役立ちます。
ステップ 9	**show cef platform trace common** [**all** | **errors** | **events** | **info**] [**location** *node-id*] **例：** RP/0/RSP0/CPU0:router# show cef platform trace common all errors location 0/4/CPU0	一般的な Dynamic Link Library（DLL; ダイナミック リンク ライブラリ）コードのトレースを表示します。

トラフィックが失われる

show interface accounting コマンドを使用するか、宛先での Rx パケットを調べて、パケットの損失を確認します。

- **show cef {ipv4}** *(destination-ip)/(mask)* **hardware egress detail location** *node-id*：プレフィクス *(destination-ip)/(mask)* に関連するハードウェア データ構造を表示します。
- **show arp location** *node-id*：特定の LC または RSP での ARP 情報を表示します。
- **show cef trace errors all reverse location** *node-id*：記録された PI コード ltrace エラーを表示します。
- **show cef platform trace {ipv4} errors all reverse location** *node-id*：プロトコル IPv4 でのプラットフォーム コード ltrace エラーを表示します。

パケットが入力から出力に正常に転送されたことを示す ucode のカウンタは次のとおりです。

- PARSE_ENET_RECEIVE_CNT：入力インターフェイスに到着したパケットを表示します。
- MODIFY_FABRIC_TRANSMIT_CNT：出力 LC に送信されたパケットを表示します。
- PARSE_FABRIC_RECEIVE_CNT：出力 LC 上のファブリックから受信されたパケットを表示します。
- MODIFY_ENET_TRANSMIT_CNT：出力インターフェイスの外部に送信されたパケットを表示します。

ドロップのないトラフィックでは、これらのカウンタの値はすべて一致します。いずれかのカウンタが一致しない場合は、状況に応じて調査を続けます。たとえば、PARSE_FABRIC_RECEIVE_CNT が MODIFY_FABRIC_TRANSMIT_CNT よりも少ない場合は、ファブリックの内部でパケットの損失が起こっています。ファブリックでパケットがドロップされる原因について、さらにトラブルシューティングを進める必要があります。

ソフトウェアで交換されるパケット

ステップ 1 宛先 IP アドレスのハードウェア チェーンが次のどちらかを示していることを確認します。

- COMPLETE の隣接関係：有効な発信パスが存在します。
- PUNT の隣接関係：ハードウェアはパケットの送出方法を知りません。単にソフトウェアで交換されるようにパケットをパント（迂回）するだけです。送信隣接関係が PUNT の場合、これは ARP がまだ解決されていないことに起因する可能性があります。

ステップ 2 宛先 IP 用の ARP エントリが存在するかどうかを表示するには、**show arp location** コマンドを使用します。

```
RP/0/RSP0/CPU0:router# show arp location node-id
```

a. ARP エントリが存在しない場合、または不完全な場合は、スタティック ARP エントリを追加します。Tx 隣接関係が「COMPLETE」を示していることを確認します。

```
RP/0/RSP0/CPU0:router# show cef {ipv4} 192.1.1.1/32 hardware egress detail location
0/4/CPU0
```

b. その場合、問題は ARP エントリが更新されないことにあります。トラブルシューティングの対象を ARP エントリが追加されない原因の追求に向けます（これには、**show arp**、**show arp idb**、**show adjacency gig** *node-id* **detail location** *node-id*、**show arp trace** などのステップが含まれます）。

c. Tx 隣接関係がまだ「PUNT」を示している場合は、ARP データベースにエントリは追加されるものの、fib_mgr が隣接関係を「COMPLETE」としてマークできないことを意味します。

d. これは fib_mgr、ARP、または AIB の問題です。スタティック ARP エントリをいったん削除し、AIB および CEF のデバッグを有効にしてスタティック ARP エントリを再設定します。デバッグ メッセージから、ARP が AIB 内部にエントリを追加しているかどうか、および AIB が fib_mgr に通知しているかどうかがわかります。

ステップ 3 パケットがファブリック内でドロップされる場合があります。これを確認するには、ファブリックのカウンタを表示します。

回避策

ステップ 1 発信インターフェイスに対して **shut** コマンド（後に **commit** を続ける）と **no shut** コマンド（後に **commit** を続ける）を使用します。

ステップ 2 宛先 IP 用のスタティック ARP エントリを追加します。

トレースルートが失敗する

traceroute は、宛先への接続性を確認するときに使用します。宛先へのトレースルートが失敗する場合は、次のコマンドを使用します。

- **show cef {ipv4}** {*destination_ip*}/*(mask*} **hardware egress detail location** *node-id*：プレフィクスに関連するハードウェア データ構造を表示します。
- **show interface location** {*outgoing_interface*} **accounting**：発信インターフェイスからの入力または出力パケットを表示します。

ステップ 1 宛先 IP アドレスの送信隣接関係が適切であるかどうかをチェックします。「Tx Adjacency」の状態を確認します（「COMPLETE」である必要があります）。

```
RP/0/RSP0/CPU0:router# show cef {ipv4} prefix hardware egress detail location node-id
```

ステップ 2 送信隣接関係が COMPLETE でない場合は、問題があります。送信隣接関係が「PUNT」を示している場合は、おそらく宛先 IP に対応する MAC アドレスが学習されていません。「static arp」エントリを追加して、送信隣接関係が「COMPLETE」に移行するかどうかを確認します。宛先 IP アドレスが OSPF などのルーティング プロトコルによってアドバタイズされる場合、送信隣接関係が「PUNT」になることはありません。

送信隣接関係が「DROP」の場合は、ルートを DROP に明示的に向ける宛先 IP アドレスへのスタティック ルートが存在することを意味します。

送信隣接関係が「COMPLETE」の場合は、設定されているハードウェア チェーンに問題はありません。カウンタを確認する必要があります。

ステップ 3 出力パケットが、送信されたトレースルート パケットと一致しているかどうかを確認します。

```
RP/0/RSP0/CPU0:router# show interface location outgoing_interface accounting
```

回避策

ステップ 1 発信インターフェイスに対して **shut** コマンド（後に **commit** を続ける）と **no shut** コマンド（後に **commit** を続ける）を使用します。

ステップ 2 宛先 IP 用のスタティック ARP エントリを追加します。

ルートの追加が失敗する

Out Of Resource（OOR; リソース枯渇）の間は、既存のルートが削除されない限り、ルータは追加のルートを受け入れません。

- **show cef resource location** *node-id* : 各種データ構造の PI 状態を表示します。理想的な状態は GREEN です。状態が YELLOW または RED の場合は、OOR と呼ばれる状態に入っていることを意味します。
- **show cef platform trace {ipv4} error reverse location** *node-id* :プロトコル IPv4 のプラットフォーム ltrace エラーを表示します。
- **show cef platform trace common error reverse location** *node-id* : すべてのプロトコルのプラットフォーム ltrace 一般エラーを表示します。

ステップ 1 正確にどのリソースが枯渇しているかを特定します。「max entries」と「used entries」を比較し、使用されているエントリ数が上限に近いデータ構造を確認します。

```
RP/0/RSP0/CPU0:router# show cef resource location node-id
```

ステップ 2 枯渇しているデータ構造が特定されたら、それが予測された状況か、予測外の状況であるかを確認できます。通常は、LEAF ごとに（IPv4 でも同じ）NR_LDI 構造のエントリが 4 つ必要です。そのため、NR_LDI 構造でリソース枯渇が起こっている場合は、IP リーフの数は適切で、NR_LDI の数が上限に達するのは妥当であるのかどうかを確認します。

ステップ 3 **show cef resource location node-id** で状態が「GREEN」と示される場合は、OOR 状態ではありません。ルートを追加できない原因は他にあります。場合によっては、何が起こっているかを把握するために、次のデバッグをイネーブルにする必要があります。

a. `RP/0/RSP0/CPU0:router# debug cef errors location node-id`

b. `RP/0/RSP0/CPU0:router# debug cef {ipv4} error location node-id`

c. トレースバックを監視する場合は、SBT ツールを使用してトレースバックをデコードします。

回避策

OOR 状態が発生していて、これが予測されたものである場合は、既存のルートをいくつか削除します。

継続的なトレースバック

トレースバックが継続的に（通常は 15 秒間隔で）コンソールに表示される場合は、ハードウェア内のエントリのプログラミングが正常ではありません。これは、15 秒経過するたびにソフトウェアが試行を繰り返すために起こります。

- **show cef platform trace common all all reverse location** *node-id*— : すべてのプラットフォーム ltrace 一般メッセージを表示します。
- **show cef platform trace {ipv4} all all reverse location** *node-id* : IPv4 のすべてのプラットフォーム ltrace プロトコル メッセージを表示します。

回避策

ステップ 1 prm_server プロセスを再起動します。

ステップ 2 LC を再起動します。

fib_mgr が起動しない

fib_mgr は、基盤となるハードウェアに依存します。基盤となるプロセスまたはハードウェアが起動しない場合は、おそらく fib_mgr も起動しません。

- **show cef platform trace common all reverse location** *node-id* : プラットフォーム ltrace 一般メッセージを表示します。
- **show cef platform trace common event reverse location** *node-id* : プラットフォーム ltrace 一般イベントを表示します。
- **show cef platform trace {ipv4 or mpls} error reverse location** *node-id*:プロトコル IPv4 または MPLS に対して記録されたプラットフォーム ltrace エラー メッセージを表示します。
- **show cef trace reverse location** *node-id* : すべての CEF ltrace メッセージを表示します。

コア ファイルを収集し、SBT を使用してトレースバックをデコードします。

回避策

ステップ 1 prm_server プロセスを再起動します。

ステップ 2 LC を再起動します。

CEF エントリが同期しない

RSP の **cef** エントリが管理インターフェイスを指していて、その結果、ルータから発信されるトラフィックが LC インターフェイスではなく管理インターフェイスから送出される場合があります。

- **show cef trace event reverse location** *node-id* : PI CEF ltrace イベントを表示します。
- **show cef trace error reverse location** *node-id* : PI CEF ltrace エラーを表示します。
- **show cef platform trace common event reverse location** *node-id* : CEF プラットフォーム一般イベントのトレースを表示します。
- **show cef platform trace common error reverse location** *node-id* : CEF プラットフォーム一般エラーのトレースを表示します。

ステップ 1 管理インターフェイスから送出するために設定されたデフォルト ルート 0.0.0.0/0 を探します。

ステップ 2 問題のプレフィクスについて設定されたスタティック ARP を探します。ARP により、管理インターフェイスを経由するエントリと LC インターフェイスを経由するエントリの 2 つのエントリがインストールされている可能性があります（プレフィクスがどちらのルートによっても到達できるため）。

ステップ 3 上記が該当しない場合は、管理インターフェイスを経由する ARP エントリをアドバタイズしているものがあるかどうかを確認します （これには **show arp** を使用します)。原因が特定されたら、ARP をクリアし、CEF エントリを再度確認します。

回避策

- 管理インターフェイスに対して **shut** コマンド（後に **commit** を続ける）と **no shut** コマンド（後に **commit** を続ける）を使用します。
- **clear arp-cache**
- LC を再起動します。

fib_mgr がクラッシュする

- **show cef platform trace common all reverse location** *node-id* : CEF プラットフォーム一般トレースを表示します。
- **show cef platform trace common event reverse location** *node-id* : CEF プラットフォーム一般イベントのトレースを表示します。
- **show cef platform trace common error reverse location** *node-id* : CEF プラットフォーム一般エラーのトレースを表示します。
- **show cef platform trace {ipv4 or mpls} error reverse location** *node-id* : IPv4 または MPLS の CEF プラットフォーム プロトコル トレースを表示します。

ステップ 1 トリガーが prm の再起動またはクラッシュの場合、これは予測される状況です。

ステップ 2 基盤となるプロセス（prm_server）がダウンまたはクラッシュした場合、おそらく fib_mgr は起動しません。

ステップ 3 コア ファイルを保存します。

ステップ 4 SBT を使用してトレースバックをデコードします。ワークスペースのルートから、./util/bin/sbt -p (process_name) -f (log_file) を使用します。

ステップ 5 コンソール ログを保存します。

回避策

fib_mgr または LC を再起動します。

トレースバックが表示される

これは、何らかのトリガー（インターフェイスの shut/no shut やその他の同様のトリガーなど）の結果としていくつかのエラー トレースバックがコンソールに表示されるというシナリオです。

- **show cef trace event location** *node-id* : 主要イベントの CEF トレースを表示します。
- **show cef trace errors location** *node-id* : 主要エラーの CEF トレースを表示します。
- **show cef platform trace common errors location** *node-id* : すべてのプロトコルにわたる一般エラーの CEF プラットフォーム トレースを表示します。
- **show cef platform trace {ipv4 or mpls} errors location** *node-id* : プロトコル IPv4 または MPLS のエラーの CEF プラットフォーム トレースを表示します。
- **show logging**

ステップ 1 SBT ツールを使用してトレースバックをデコードします。ワークスペースのルートから、./util/bin/sbt -p (process_name) -f (log_file) を使用します。

ステップ 2 コア ファイルを保存します。

回避策

トレースバックがサービスに影響を与える場合は、次の手順を実行します。

ステップ 1 fib_mgr プロセスを再起動します。

ステップ 2 LC を再起動します。

サブインターフェイスでのカプセル化の変更が原因でトラフィックが失われる

トラフィックが L3 サブインターフェイス経由で転送されている場合に、そのサブインターフェイスでカプセル化が変更されたとき、15 秒経過するまでトラフィックが再開されないことがあります。

- **show cef trace event reverse location** *node-id* : 主要イベントの CEF トレース メッセージを表示します。
- **show cef trace error reverse location** *node-id* : 主要エラーの CEF トレース メッセージを表示します。
- **show cef platform trace common error location** *node-id* : すべてのプロトコルにわたる一般エラーの CEF プラットフォーム トレースを表示します。
- **show cef platform trace {ipv4 or mpls} event location** *node-id* : プロトコル IPv4 または MPLS の主要イベントの CEF プラットフォーム トレースを表示します。
- **show cef platform trace {ipv4 or mpls} error location** *node-id* : プロトコル IPv4 または MPLS の主要エラーの CEF プラットフォーム トレースを表示します。
- **show arp location** *node-id* : ARP 関連の情報を表示します。

サブインターフェイスでカプセル化が **dot1q vlan 300** から **dot1q vlan 200** に変更されると、fib_mgr はこのインターフェイスに対応するすべてのプレフィクスを削除してから再作成します。すべてのプレフィクスを追加するまでに 15 秒かかります。その間、トラフィックは転送されません。たとえば、アドレス 192.0.0.0/8 のインターフェイスがあり、192.2.2.2 のスタティック ARP エントリがあるとします。

```
RP/0/RSP0/CPU0:router#show run | inc arp
```

（スタティック ARP ではなく）通常の隣接関係では、遅延が起こる可能性は低くなります。

VLAN カラーが変更されると、次のことが起こります。

- 隣接関係が削除され、隣接ルート 192.2.2.2 が削除されます。
- 接続ルートが削除されます。
- 接続ルートが追加される前に隣接関係が追加されます。FIB では、対応する接続ルートのない隣接関係の追加はエラーとして扱われるため、ルート 192.2.2.2 の設定は再試行に回されます。
- 接続ルート 192.0.0.0/8 が追加されます。
- FIB の再試行タイマーは 15 秒なので、隣接ルート 192.2.2.2 は 15 秒後に追加されます。

回避策

スタティック ARP エントリを削除します。

RSP フェールオーバー中にトラフィックが失われる

RSP フェールオーバーが原因でトラフィックが失われることがあります。これは、プレフィクスの学習に使用される IGP がダウンしていることを意味する場合があります。次の説明は、IGP として OSPF を使用していることを前提とします。

- **show process failover**：フェールオーバー中のプロセスの詳細を表示します。
- **debug ospf ha**：OSPF HA 関連のデバッグをイネーブルにします。
- **debug ospf** *instance* **nsf**：フェールオーバーの前に実行し、デバッグ ログを収集します。
- **show process failover**：フェールオーバーの後に実行します。

ステップ 1 ネクストホップ ルータでフェールオーバーが作動したかどうかをチェックします。

a. フェールオーバーが作動した場合、OSPF はダウンします。

a. フェールオーバーが作動していない場合は、OSPF で **nsf cisco** が設定されていることを確認します。

nsf cisco が設定されている場合は、フェールオーバー中にネクストホップに到達可能かどうかを確認します。
到達不能な場合は、リンクがダウンしているか、ネゴシエーションに問題がある可能性があります。

回避策

ルータをリロードします。

IP マルチキャスト

IP マルチキャストは、単一の情報ストリームを企業や家庭の何千もの受信者に同時に送信することによってトラフィックを削減する技術であり、帯域幅を大量に消費します。このセクションの内容は次のとおりです。

show および debug コマンドの使用方法

手順概要

1. **show igmp {global-interface | groups | | interface | nsf | old-output | snooping | ssm | summary | traffic | vrf** *name*}
2. **show pim {bgp-safi | bsr | context | df | group-map | interface | ipv4 | ipv6 | join-prune | mdt | mstatic | multicast | neighbor | nsf | old-output | range-list | rpf | safi-all | summary | table-context | topology | traffic | tunnel | unicast | vrf}**
3. **show mrib {client | route | table-info}**
4. **show mfib {connections | counter | encap-info | hardware | interface | ipv4 | ipv6 | lsm | mdt | nsf | route | svd | table-info | vrf)**
5. **show mfib hardware { | interface { location** *node-id*} | **ltrace | resource-counters | route }**
6. **show mfib hardware route { accept-bitmap | olist | statistics | summary } {*** | *A.B.C.D* | *A.B.C.D/length* | **detail | hex-dump} location** *node-id*
7. **show mfib hardware route summary location** *node-id*
8. **debug mrib errors**
9. **debug mrib events**
10. **debug mfib warning**
11. **debug mfib errors**
12. **debug mlib errors**
13. **debug mlib warning**

詳細手順

	コマンドまたはアクション	目的
ステップ 1	**show igmp {global-interface | groups | interface | nsf | old-output | snooping | ssm | summary | traffic | vrf** *name*} **例：** `RP/0/RSP0/CPU0:router# show igmp groups`	コントロール プレーンに保持されている Internet Group Management Protocol（IGMP; インターネット グループ管理プロトコル）関連の情報をすべて表示します。IGMP は、IP マルチキャスト メンバシップを隣接するマルチキャスト ルータに報告するために IPv4 システムで使用されるプロトコルです。 結果をフィルタリングするには、次のパラメータとキーワードを使用します。 • **global-interface**：IGMP グローバル Interface Descriptor Block（IDB; インターフェイス記述ブロック）データ構造。IDB には、IP アドレス、インターフェイスの状態、パケットの統計情報などの情報が含まれます。IDB は、インターフェイスごとに 1 つ、およびサブインターフェイスごとに 1 つ存在します。 • **groups**：IGMP グループ メンバシップ。 • **interface**：IGMP インターフェイス情報。 • **nsf**：IGMP の現在のマルチキャスト NSF 状態（通常、または NSF のために作動）。後者の状態は、IGMP の障害のために回復が進行中であることを示します。NSF が満了するまで、NSF の総タイムアウト時間と残り時間が表示されます。 • **old-output**：下位互換性を提供します。 • **snooping**：IGMP スヌーピング パラメータ。 • **ssm**：Source Specific Multicast（SSM）関連の情報。 • **summary**：IGMP の概要。 • **traffic**：IGMP トラフィック カウンタ。 • **vrf** *name*：Virtual Private Network（VPN; バーチャル プライベート ネットワーク）Routing and Forwarding（VRF; VPN ルーティングおよび転送）を指定します。

	コマンドまたはアクション	目的
ステップ 2	**show pim {bgp-safi | bsr | context | df | group-map | interface | ipv4 | ipv6 | join-prune | mdt | mstatic | multicast | neighbor | nsf | old-output | range-list | rpf | safi-all | summary | table-context | topology | traffic | tunnel | unicast | vrf}** **例：** `RP/0/RSP0/CPU0:router# show pim neighbor`	コントロール プレーンに保持されている Protocol Independent Multicast（PIM）関連の情報を表示します。結果をフィルタリングするには、次のパラメータとキーワードを使用します。 • **bgp-safi**：Border Gateway Protocol（BGP; ボーダー ゲートウェイ プロトコル）の Secondary Address Family（SAFI; セカンダリ アドレス ファミリ）データベース。 • **bsr**：PIM Bootstrap Router（BSR; ブートストラップ ルータ）情報。 • **context**：PIM VRF コンテキスト。 • **df**：双方向 Designated Forwarder（DF; 指定フォワーダ）。 • **group-map**：PIM のグループとプロトコルのマッピング情報。 • **interface**：PIM インターフェイス情報。 • **ipv4**：IPv4 アドレス ファミリ。 • **ipv6**：IPv6 アドレス ファミリ。 • **join-prune**：PIM Join/Prune 情報。 • **mdt**：データ MDT 情報。 • **mstatic**：マルチキャスト スタティック ルート情報。 • **multicast**：SAFI マルチキャスト。 • **neighbor**：PIM ネイバー情報。 • **nsf**：ノンストップ フォワーディング。 • **old-output**：下位互換性を提供します。 • **range-list**：PIM 範囲リスト情報。 • **rpf**：RPF 情報。 • **safi-all**：SAFI ワイルドカード。 • **summary**：PIM 概要情報。 • **table-context**：PIM テーブル コンテキスト。 • **topology**：PIM トポロジ テーブル情報。 • **traffic**：PIM トラフィック カウンタ。 • **tunnel**：トンネル インターフェイス。 • **unicast**：SAFI ユニキャスト。 • **vrf**：VRF。

	コマンドまたはアクション	目的
ステップ 3	**show mrib {client | route | table-info}** **例：** `RP/0/RSP0/CPU0:router# show mrib client`	Multicast Routing Information Base（MRIB）情報を表示します。結果をフィルタリングするには、次のパラメータとキーワードを使用します。 • **client**：MRIB クライアント接続。 • **route**：ルーティング データベース。 • **table-info**：MRIB VRF テーブル情報。
ステップ 4	**show mfib {connections | counter | encap-info | hardware | interface | ipv4 | ipv6 | lsm | mdt | nsf | route | svd | table-info | vrf)** **例：** `RP/0/RSP0/CPU0:router# show mfib nsf`	コントロール プレーンに保持されている Multicast Forwarding Information Base（MFIB; マルチキャスト転送情報ベース）情報を表示します。結果をフィルタリングするには、次のパラメータとキーワードを使用します。 • **connections**：サーバへの MFIB 接続のステータス。 • **counter**：MFIB グローバル カウンタ。 • **encap-info**：Multicast Virtual Private Network（MVPN; マルチキャスト バーチャル プライベート ネットワーク）カプセル化情報。 • **hardware**：Cisco ASR 9000 シリーズ ルータ ハードウェア。 • **interface**：MFIB インターフェイス固有の情報。 • **ipv4**：IPv4 アドレス ファミリ。 • **ipv6**：IPv6 アドレス ファミリ。 • **lsm**：Label Switched Multicast。 • **mdt**：MDT トンネル情報。 • **nsf**：マルチキャスト NSF ステータス。 • **route**：ルーティング データベース。 • **svd**：Singular Value Decomposition（SVD; 特異値分解）イベント。 • **table-info**：テーブル情報。 • **vrf**：VRF。

	コマンドまたはアクション	目的
ステップ 5	**show mfib hardware {connection | interface | ltrace | resource-counters | route} location** *node-id* **例：** RP/0/RSP0/CPU0:router# show mfib hardware connection location 0/4/CPU0	マルチキャスト PD 内のすべてのハードウェア データを表示します。結果をフィルタリングするには、次のパラメータとキーワードを使用します。 • **connection**：サーバへの MFIB 接続。 • **interface**：Cisco ASR 9000 シリーズ ルータ ハードウェア。 • **ltrace**：IP マルチキャスト プラットフォーム固有のトレース情報。 • **resource-counters**：割り当て済みハードウェア リソースと解放されたハードウェア リソース。 • **route**：ルーティング データベースのプラットフォーム固有の情報。 • **location**：MFIB ロケーションを指定します。 **（注）** 設定されているルートおよび olist の数が多い場合、これらのコマンドの出力は長くなることがあります。
ステップ 6	**show mfib hardware route {accept-bitmap | olist | statistics | summary} {*** | *A.B.C.D* | *A.B.C.D/length* | **detail | hex-dump} location** *node-id* **例：** RP/0/RSP0/CPU0:router# show mfib hardware route olist location 0/4/CPU0	マルチキャスト PD 内のすべてのハードウェア ROUTE データを表示します。結果をフィルタリングするには、次のパラメータとキーワードを使用します。 • **accept-bitmap**：双方向ルートの受け入れインターフェイスの一覧。 • **olist**：ハードウェアに格納された output interface list（olist; 出力インターフェイス リスト）。 • **statistics**：ルートごとのパケットとバイトのカウンタ。 • **summary**：ルートの概要。 • ***：共有ツリー エントリ。 • **A.B.C.D**：送信元/グループ IP アドレス。 • **A.B.C.D/length**：グループ IP アドレス/プレフィクス長。 • **detail**：各ルートの詳細（140 カラムが必要）。 • **hex-dump**：PLU および TLU の 16 進ダンプ。 • **location**：MFIB ロケーションを指定します。 **（注）** 設定されているルートおよび olist の数が多い場合、これらのコマンドの出力は長くなることがあります。

	コマンドまたはアクション	目的
ステップ 7	**show mfib hardware route summary location** *node-id* **例**： `RP/0/RSP0/CPU0:router# show mfib hardware route summary location 0/4/CPU0`	マルチキャスト PD 内のすべてのハードウェア ROUTE データを表示します。結果をフィルタリングするには、次のパラメータとキーワードを使用します。 • **summary**：ルートの概要。 • **location**：MFIB ロケーション。
ステップ 8	**debug mrib errors** **例**： `RP/0/RSP0/CPU0:router# debug mrib errors`	Multicast Routing Information Base（MRIB）内部エラーを監視するには、EXEC モードで debug mrib errors コマンドを使用します。デバッグ出力をディセーブルにするには、このコマンドの no 形式を使用します。
ステップ 9	**debug mrib events**	—
ステップ 10	**debug mfib warning**	—
ステップ 11	**debug mfib errors**	—
ステップ 12	**debug mlib errors**	—
ステップ 13	**debug mlib warning**	—

マルチキャスト PIE のインストールが失敗する

ステップ 1 指定したインストール ID の詳細情報を表示します。

```
RP/0/RSP0/CPU0:router# show install log [1-4294967295] detail
```

ステップ 2 PIE ファイルの名前が正しいことを確認し、コマンドを再度実行します。

ステップ 3 PIE ファイルの場所が正しいことを確認し、コマンドを再度実行します。

ステップ 4 PIE ファイルのアクセス権が適切である（755 である）ことを確認し、コマンドを再度実行します。

ステップ 5 TFTP ディレクトリからロードする場合は、次のことを確認します。

a. ルータがネットワークに接続されている。

b. TFTP アドレスが適切に設定されている。

c. TFTP サーバに接続できる。

```
RP/0/RSP0/CPU0:router# ping tftp-server-addr
```

d. ルータからローカルにロードする場合は、PIE ファイルがルータに格納されていることを確認します。

ステップ 6 すべてのノードの状態が「IOS XR RUN State」であることを確認します。

```
RP/0/RSP0/CPU0:router# show platform
```

PIE がインストールされているにもかかわらず、マルチキャスト CLI が使用できない

show install active：アクティブ パッケージ情報を表示します。

次のものが正しいことを確認します。

- PIE ファイルの名前
- PIE ファイルの場所
- すべてのノードでの PIE ファイルのインストール

「This command not authorized」エラー メッセージ

設定モードまたは EXEC モードで特定のコマンドを実行するとき、「This command not authorized」というエラー メッセージが表示され、それ以降アクセスできなくなります。これは、コマンドを実行したユーザが適切な権限を持っていないことを意味します。目的のコマンドを使用するための「Cisco-Support」権限および「root」権限をユーザが持っているかどうかをチェックします。

show config run：（管理モードから実行）システムで現在動作している管理コンフィギュレーションを表示します。

ダイナミック IGMP 障害

ダイナミック IGMP 障害とは、Dynamic (*,G) がタイムアウトすることです。グループとルートが正しく設定およびセットアップされているにもかかわらず、テスターから Cisco ASR 9000 シリーズ ルータに送信されたトラフィックが Rx テスター ポートで受信されません。

ステップ 1 考えられる原因の 1 つとして、IGMP グループがタイムアウトしていることが挙げられます。これをチェックする 1 つの方法は、(*,G) のスタティック ルートを作成し、トラフィックが受信されるようになるかどうかを確認することです。トラフィックが受信される場合は、グループがタイムアウトしていることを意味します。

ステップ 2 ステップ 1 の結果を立証するには、スタティック ルートを削除し、現象をより明確にするためにクエリー間隔を短くします（その結果、1 分あたりのクエリー数が増加します）。

```
RP/0/RSP0/CPU0:router# conf t
RP/0/RSP0/CPU0:router(config)# router igmp
RP/0/RSP0/CPU0:router(config-igmp)# query-interval 1
RP/0/RSP0/CPU0:router(config-igmp)# commit
```

ステップ 3 設定した間隔でインターフェイスからパケットが送出されることを確認します。

ステップ 4 テスターが IGMP メンバシップ レポートで応答することをチェックします。パケットがテスターで受信された場合、ステップ 1 の結果は立証されます。回避策を実施してください。

回避策

一時的な回避策としてスタティック グループを設定します。

一部のインターフェイスでトラフィックが失敗する

一部のインターフェイスまたはチャネルでトラフィックが失敗します。グループとルートは正しく設定され、セットアップされています。トラフィックはテスターから Cisco ASR 9000 シリーズ ルータに送信されます。送信されたトラフィックは一部のインターフェイスでは正しく受信されますが、残りのインターフェイスでは受信されません。また、あるインターフェイスで一部のビデオ チャネルだけが正しく受信され、残りのビデオ チャネルが受信されないこともあります。これには次のような原因が考えられます。

- OLIST が適切に設定されていない。
- UIDB 値がハードウェアに正しく設定されていない。
- MGID が正しく設定されていない。

ステップ 1 パケットが入力 NP からファブリックを経由して出力 NP に移動していることを確認します。

```
RP/0/RSP0/CPU0:router# show mfib hardware route statistics location {ingress node-id | egress node-id }
```

ステップ 2 ルートの olist インターフェイスを表示します。

```
RP/0/RSP0/CPU0:router# show mfib hardware route olist location {ingress node-id | egress node-id }
```

ステップ 3 特定のルートおよび送信元の統計情報を表示します。

```
RP/0/RSP0/CPU0:router# show mfib hardware route stat [src ip addr] location {ingress node-id | egress node-id }
```

一部のインターフェイスでトラフィックが失敗する：MGID

ステップ 1 パケットが入力 NP からファブリックを経由して出力 NP に移動していることを確認します。

```
RP/0/RSP0/CPU0:router# show mfib hardware route statistics location {ingress node-id | egress node-id }
```

ステップ 2 ルートの olist インターフェイスを表示します。

```
RP/0/RSP0/CPU0:router# show mfib hardware route olist location {ingress node-id | egress node-id }
```

ステップ 3 パケットが入力 NP からファブリックに送出され、出力 NP でファブリックから受信されていることを確認します。

```
RP/0/RSP0/CPU0:router# show mfib hardware route statistics location {ingress node-id | egress node-id }
```

ステップ 4 ルートの MGID を表示します。

```
RP/0/RSP0/CPU0:router# show controllers bundle bundle-ether bundle-id location {ingress node-id | egress node-id }
```

受信側インターフェイスでのスループットの損失

トラフィックはルート上で送受信されますが、受信側でスループットの損失が起こります。

ステップ 1 パケットが入力 NP からファブリックを経由して出力 NP に移動していることを確認します。

```
RP/0/RSP0/CPU0:router# show mfib hardware route statistics location {ingress node-id |
egress node-id }
```

ステップ 2 上記のコマンドから、パケットが RP にパントされているかどうかがわかります。パントされている場合は、そのチャネルの送信元が一部の IP オプションを設定しているかどうかをチェックします。

マルチ プロトコル ラベル スイッチング（MPLS）

Multi Protocol Label Switching（MPLS; マルチ プロトコル ラベル スイッチング）は、IP パケットやイーサネット フレームなどの異なる種類のトラフィックを伝送します。このセクションの内容は次のとおりです。

show および debug コマンドの使用方法

手順概要

1. **debug mpls ldp transport events**
2. **debug mpls ldp transport connections**
3. **show mpls forwarding tunnels**
4. **debug mpls ea platform {all | errors | events | info} [location]**
5. **show cef platform trace [adj | all } common | ipv4 | ipv6 | mpls | rpf | te]**
6. **show cef platform { resource | trace }**
7. **show mpls forwarding labels** *label* **hardware egress location** *node id*

詳細手順

	コマンドまたはアクション	目的
ステップ 1	**debug mpls ldp transport event**	検出および接続セットアップ/シャットダウン イベント
ステップ 2	**debug mpls ldp transport connection**	接続セットアップ/シャットダウン イベント
ステップ 3	**show mpls forwarding tunnels**	—
ステップ 4	**debug mpls ea platform {info | errors | events | all } [location]**	—
ステップ 5	**show cef platform trace [adj | all | common | ipv4 | ipv6 | mpls | rpf | te]**	—
ステップ 6	**show cef platform [resource | trace]**	—
ステップ 7	**show mpls forwarding label** *label* **hardware egress location** *node id*	—

IP パケットが LSP に転送されない

ステップ 1 ハードウェア ラベル FIB をチェックします。

```
RP/0/RSP0/CPU0:router# show mpls forwarding labels label hardware egress location node id
```

ステップ 2 ネクストホップ プレフィクスが ARP によって解決されているかどうかを確認します。

```
RP/0/RSP0/CPU0:router# show arp prefix location node id
```

回避策

直接接続されたリモート インターフェイスの MAC アドレスを変更し、ルータで隣接関係の変更を発生させます。

IP パケットが MPLS TE トンネルに転送されない

ステップ 1 プレフィクスのトンネル隣接関係をチェックします。

```
RP/0/RSP0/CPU0:router# show cef prefix hardware egress location node-id
```

ステップ 2 MPLS トラフィック トンネルがアップしていることを確認します。

```
RP/0/RSP0/CPU0:router# show mpls traffic-eng tunnels up
```

ステップ 3 ハードウェア TE ラベル FIB をチェックします。

```
RP/0/RSP0/CPU0:router# show cef adjacency tunnel-te tunnel-id hardware egress location
node-id
```

回避策

トンネル インターフェイスに対して **shut** コマンド（後に **commit** を続ける）と **no shut** コマンド（後に **commit** を続ける）を入力し、ハードウェアを再プログラムします。

MPLS パケットが MPLS TE トンネルに転送されない

ステップ 1 送信隣接関係が COMPLETE であることを確認します。

```
RP/0/RSP0/CPU0:router# show mpls for labels label-id ha eg location node-id
```

ステップ 2 ハードウェア トンネル隣接関係が COMPLETE であることを確認します。

```
RP/0/RSP0/CPU0:router# show cef adj tunnel-te te-id hardware egress location node-id
```

回避策

トンネル インターフェイスに対して **shut** コマンド（後に **commit** を続ける）と **no shut** コマンド（後に **commit** を続ける）を実行し、ハードウェアを再プログラムします。

MPLS TE トンネルが起動しない

ステップ 1 RSVP でトンネル出力インターフェイスが設定されていることを確認します。

```
rsvp
 interface Bundle-Ether1
  bandwidth 100000
 !
 interface GigabitEthernet0/1/0/2
  bandwidth 100000
 !
 interface GigabitEthernet0/4/0/8  <<---- tunnel egress interface
  bandwidth 100000
 !
 interface GigabitEthernet0/4/0/20
  bandwidth 100000
 !
mpls traffic-eng <<---- Ensure that the tunnel egress interface is configured in mpls
traffic-engineering config
 interface Bundle-Ether1
 !
 interface GigabitEthernet0/1/0/2
 !
 interface GigabitEthernet0/4/0/8 <<---- tunnel egress interface
 !
 interface GigabitEthernet0/4/0/20
 !
```

ステップ 2 OSPF でトラフィック エンジニアリングが設定されていることを確認します。

```
router ospf te
 log adjacency changes detail
 router-id 192.1.1.30
 area 0
  mpls traffic-eng
```

ステップ 3 トンネル宛先 IP への ping が成功することを確認します。

FRR によって保護されたトンネルが FRR の作動後にダウンする

FRR は、障害ポイントで LSP をローカルに修復することによって MPLS Traffic Engineering（TE; トラフィック エンジニアリング）Label-Switched Path（LSP; ラベル スイッチド パス）をリンク障害やノード障害から保護するメカニズムです。これにより、ヘッドエンド ルータが新しい代替エンドツーエンド LSP を確立する間も、これらの LSP 上のデータ フローは中断されません。FRR は、障害が起こったリンクまたはノードを迂回するバックアップ トンネルに LSP を再ルーティングすることで、保護された LSP をローカルに修復します。

ステップ 1 更新された送信者テンプレート内のアドレスに対して ping を実行します。

```
RP/0/RSP0/CPU0:router# ping PLR_Address
```

ステップ 2 MP アドレスが到達可能であることを確認します。バックアップ トンネル上での転送が機能していることをチェックします。

```
RP/0/RSP0/CPU0:router# ping backup_tunnel_destination
```

ステップ 3 バックアップ トンネルが Up, Up 状態であることを確認します。

```
RP/0/RSP0/CPU0:router# show mpls traffic-eng tunnels
```

ステップ 4 RSVP トレースをチェックし、トンネルがダウンする原因を探します。

```
RP/0/RSP0/CPU0:router# show mpls traffic-eng trace event
```

MPLS TE FRR データベースが構築されない

ステップ 1 保護トンネルが高速再ルーティング可能であることを確認します。

a. `RP/0/RSP0/CPU0:router# show mpls traffic-eng fast-reroute database`

ステップ 2 バックアップが保護インターフェイスを通過しないことを確認します。

```
RP/0/RSP0/CPU0:router# show mpls traffic-eng tunnel backup protected-interface
```

ステップ 3 バックアップに十分なバックアップ帯域幅があることを確認します（設定されている場合）。

```
RP/0/RSP0/CPU0:router# show mpls traffic-eng tunnels backup
```

ステップ 4 バックアップ トンネルと保護トンネルに合流点があることを確認します（ホップ情報をチェックします）。

```
RP/0/RSP0/CPU0:router# show mpls traffic-eng tunnels
```

ステップ 5 保護トンネルとバックアップ トンネルが Up, Up 状態であることを確認します。

```
RP/0/RSP0/CPU0:router# show mpls traffic-eng tunnels brief
```

（注） 指定された帯域幅が 0 の保護トンネルは、制限された backup-bw トンネルでは保護できません。

ステップ 6 デバッグをイネーブルにし、バックアップ tunnel-te を削除して再適用します。

```
RP/0/RSP0/CPU0:router# debug mpls traffic-eng frr
```

ステップ 7 バックアップ トンネルまたは保護トンネル（またはその両方）に対して shut コマンドと no shut コマンドを実行します（可能な場合）。これにより、バックアップ トンネルの割り当てがリセットされます。

ステップ 8 バックアップ トンネルに fast-reroute オプションが設定されていないことを確認します。

```
RP/0/RSP0/CPU0:router# show running-config interface tunnel-te15
```

ステップ 9 バックアップが保護 LSP に割り当てられているかどうかを確認します。

a. `RP/0/RSP0/CPU0:router# show mpls traffic-eng fast-reroute database`

b. `RP/0/RSP0/CPU0:router# show mpls traffic-eng forwarding`

c. `RP/0/RSP0/CPU0:router# show rsvp fast-reroute`

ステップ 10 保護 LSP 帯域幅のプールタイプとバックアップ トンネルの backup-bw のプールタイプが一致していることを確認します。

MPLS FRR 切り替え時間のデバッグ

ステップ 1 FRR データベースが構築されていて、使用可能な状態であることを確認します。

```
RP/0/RSP0/CPU0:router# show mpls traffic-eng fast-reroute database
```

ステップ 2 FRR が作動したときに、FRR がアクティブな状態になることを確認します。

```
RP/0/RSP0/CPU0:router# show mpls traffic-eng fast-reroute database
```

ステップ 3 プライマリ トンネルで障害が発生した LC の FRR 切り替え時間をチェックします。

```
RP/0/RSP0/CPU0:router# show mpls traffic-eng fast-reroute log location node-id
```

ステップ 4 LC のプライマリ トンネルとバックアップ トンネルの両方が FRR トリガーを受信したことを確認します。

```
RP/0/RSP0/CPU0:router# show cef platform trace te all location node-id
```

Reverse Path Forwarding（RPF）

Reverse Path Forwarding（RPF）は、マルチキャスト ルーティングにおいてループのないマルチキャスト パケットの転送を実現します。このセクションの内容は次のとおりです。

show および debug コマンドの使用方法

show cef ipv4 interface：インターフェイスの IPv4 Cisco Express Forwarding（CEF; シスコ エクスプレス フォワーディング）関連の情報を表示します。

間違った IP アドレスからのパケット：Loose RPF

Loose RPF では、その特定のインターフェイスにパケットが到着すると、ボックス上のいずれかのインターフェイスを通じてそのパケットの送信元 IP に到達できるかどうかがチェックされます。到達できない場合、パケットはドロップされます。

システムが RPF リスト（uidb1: 12 uidb2: 0 uidb3: 0 uidb4: 0）を返すことを確認します。RPF リストには、インデックスが 0 でない UIDB が少なくとも 1 つ含まれている必要があります。すべて 0 の場合は、Loose RPF がハードウェア内部で適切に設定されていないことを意味します。表示されたインデックスがすべて 0 の場合は、Loose RPF の設定に「allow-default」オプションが付いている可能性があります。この場合は、システムにデフォルト ルートが設定されている場合でも、RPF チェックは合格します。

```
RP/0/RSP0/CPU0:router# show cef {ipv4} prefix hardware egress detail location node-id
```

回避策

インターフェイスで Loose RPF の設定をいったん解除し、再設定します。

間違った IP アドレスで転送されたパケット：Strict RPF

Strict RPF では、その特定のインターフェイスにパケットが到着すると、ボックス上のパケットが到着したインターフェイス自体を通じてそのパケットの送信元 IP に到達できるかどうかがチェックされます。到達できない場合、パケットはドロップされます。

システムが RPF リスト（uidb1: 12 uidb2: 0 uidb3: 0 uidb4: 0v）を返すことを確認します。RPF リストには、インデックスが 0 でない UIDB が少なくとも 1 つ含まれている必要があります。さらに、0 でないインデックスが、パケットの入力インターフェイスに対応する同じインデックスであることも必要です。

```
RP/0/RSP0/CPU0:router# show cef {ipv4} prefix hardware egress detail location node-id
```

回避策

インターフェイスで Strict RPF の設定をいったん解除し、再設定します。

VRRP（仮想ルータ冗長プロトコル）

Virtual Router Redundancy Protocol（VRRP; 仮想ルータ冗長プロトコル）を使用すると、ルータのグループを 1 つの仮想ルータにすることができます。このセクションの内容は次のとおりです。

show および debug コマンドの使用方法

手順概要

1. **show vrrp [interface [***type interface-id***]] [brief]**
2. **show vrrp [interface [***type interface-id***]] detail**
3. **show vrrp [interface [***type interface-id***]] statistics [all]**
4. **show controllers** *type*
5. **debug vrrp [all] edm | events | packets]**

詳細手順

	コマンドまたはアクション	目的
ステップ 1	**show vrrp [interface** *type interface-name* **| brief]** **例：** `RP/0/0/CPU0:# show vrrp brief`	すべての VRRP グループのステータスを表示します。
ステップ 2	**show vrrp [interface** *type interface-name* **| detail]** **例：** `RP/0/0/CPU0:# show vrrp detail`	VRRP グループの詳細情報を表示します。
ステップ 3	**show vrrp [interface** *type interface-name* **| statistics [all]]** **例：** `RP/0/0/CPU0:# show vrrp statistics`	VRRP 統計情報を表示します。

	コマンドまたはアクション	目的
ステップ 4	**show controllers** *type interface-name* **例：** `RP/0/0/CPU0:# show controllers gigabitEthernet 0/3/0/9`	VRRP グループの MAC アドレスをユニキャスト フィルタ リストの一部として表示します。
ステップ 5	**debug vrrp [all] edm** | **errors** | **events** | **packets]** **例：** `RP/0/0/CPU0:# show vrrp packets tengige 0/3/0/9`	—

VRRP がアクティブな状態にならない

両方のルータで次の手順を実行します。

ステップ 1 `RP/0/RSP0/CPU0:router# show vrrp detail`

設定ミス

ステップ 1 VRRP が設定されているインターフェイスがアップしていることを確認します。

ステップ 2 インターフェイスと同じサブネット上に IP アドレスが設定されていて、遅延が設定されていることを確認します。

`RP/0/RSP0/CPU0:router# show vrrp detail`

優先度の高いルータがすでにアクティブになっている

show vrrp コマンドの出力から、次のことを確認します。

- VRRP のマスター アドレスがローカルではなく IP アドレスを示している場合は、その IP アドレスを持つルータがアクティブになっている。
- プリエンプションがイネーブルであるが、他のルータの方が優先度が高い場合、そのルータがアクティブ状態を維持する。

運用上の優先度は、設定された優先度と一致しない場合があります。インターフェイスがダウンしている場合、これは運用上の優先度にマイナスの影響を与えます。

プリエンプションがディセーブルで、別のルータがすでにアクティブになっている

show vrrp コマンドの出力から、次のことを確認します。

- プリエンプションがディセーブルかどうか。
- そのルータの方が優先度が高い場合、プリエンプションがイネーブルにされない限り、そのルータへの切り替えは起こらない。

追跡インターフェイスで障害が発生してもルータの状態が変更されない

両方のルータで次の手順を実行します。

```
RP/0/RSP0/CPU0:router# show vrrp detail
```

プリエンプションがイネーブルであっても、このルータの方が他のルータよりも運用上の優先度が高い場合は、このルータがアクティブ状態を維持します。設定優先度、または追跡インターフェイスのデクリメントは、状態遷移が起こるように適切に設定する必要があります。IP アドレスがインターフェイスの IP アドレスと同じである場合、ルータの状態はスタンバイに移行しません。

VRRP 状態のフラップ

両方のルータで次の手順を実行します。

ステップ 1 `RP/0/RSP0/CPU0:router# show vrrp detail`

ステップ 2 `RP/0/RSP0/CPU0:router# debug vrrp packets`

タイムスタンプをチェックし、パケットの送信または受信に遅延が見られるかどうかを確認します。CPU の使用状況をチェックし、システム リソースを占有しているプロセスがないかどうかを確認します。

ステップ 3 `RP/0/RSP0/CPU0:router# show spp node-counters location interface-running-vrrp`

複数の VRRP ルータがアクティブになる

ステップ 1 両端で同じ IP が設定されていることを確認します。

```
RP/0/RSP0/CPU0:router# show vrrp detail
```

ステップ 2 タイムスタンプをチェックし、パケットの送信または受信に遅延が見られるかどうかを確認します。CPU の使用状況をチェックし、リソースを過度に使用しているプロセスがないかどうかを確認します。

ステップ 3 ピアで VRRP パケットのデバッグ コマンドを入力します。

```
RP/0/RSP0/CPU0:router# debug vrrp packets
```

RP/0/RSP0/CPU0:Sep 8 14:16:39.217 : vrrp[357]: Gi0/5/0/0: VR1: Pkt: ADVER: IN: pri 100 src 192.0.0.11 のような行を探します。これは、アドバタイズメント パケットが VRRP で受信されていることを意味します。このような行がない場合、パケットは受信されておらず、VRRP がアクティブになります。

RP/0/RSP0/CPU0:Sep 8 14:18:47.876 : vrrp[357]: Gi0/5/0/0: VR1: Pkt: ADVER: Out: pri 100 src 192.0.0.11 のような行を探します。これは、ピアが VRRP パケットを送信していることを意味します。

ステップ 4 両方のルータで **show spp node-counters location** *interface-running-vrrp* の出力をチェックし、パケットのドロップを探します。

```
RP/0/RSP0/CPU0:router# show spp node-counters location interface-running-vrrp
```

VRRP アクティブ ルータがトラフィックを転送しない

両方のルータで次の手順を実行します。

ステップ 1 グループの仮想 MAC アドレスを確認します。

```
RP/0/RSP0/CPU0:router# show vrrp detail
```

ステップ 2
```
RP/0/RSP0/CPU0:router# show ether-ctrl trace
```

ステップ 3 その仮想 MAC アドレスがユニキャスト アドレス フィルタ リストに含まれていることを確認します。ルータがトラフィックを受信していることを確認します。

```
RP/0/RSP0/CPU0:router# show controllers type interface-running-vrrp
```

インターフェイスの shut/no shut を実行した後にトラフィックが失われる、または予期しない VRRP 状態になる

VRRP がイネーブルにされたインターフェイスで shut/no shut を実行した場合、次の現象が起こることが認められています。

- プリエンプションがイネーブルの場合、回復時間がフェールオーバー時間よりも長くなる。これは、インターフェイスが no shut のときにトラフィックの損失が大きくなることを意味します。
- プリエンプションがディセーブルの場合、インターフェイスの no shut を実行した後に一部の VRRP グループがプリエンプトされる。

インターフェイスの no shut を実行した後に上記のいずれかの現象が見られた場合は、次の手順に従います。

両方のルータで次の手順を実行します。

ステップ 1
```
RP/0/RSP0/CPU0:router# show vrrp detail
```

ステップ 2
```
RP/0/RSP0/CPU0:router# show ether-ctrl trace
```

ステップ 3
```
RP/0/RSP0/CPU0:router# show controllers type interface-running-vrrp
```

ステップ 4 `RP/0/RSP0/CPU0:router# debug vrrp packets interface` : **no shut** を実行する対象のインターフェイスに対して実行します。

ステップ 5 **no shut** コマンドを入力します。

ステップ 6 コンソール ログを確認し、次のような行を探します。

```
RP/0/RSP0/CPU0:Sep 8 14:16:39.217 : vrrp[357]: Gi0/5/0/0: VR1: Pkt: ADVER: IN: pri 100
src 192.0.0.11.
```

no shut を実行してからこのようなメッセージが最初に表示されるまでの時間差に注意します。その間、2 台のルータ間のトラフィックは失われます。

ステップ 7 no shut イベントの後、2 台のルータ間にトラフィック フローがない場合は、Cisco ASR 9000 シリーズ ルータの STP 設定をチェックします。fwd 遅延タイマーを短くすると、トラフィック損失の低減に役立つことがあります。

ステップ 8 プリエンプションがディセーブルのケースでは、fwd 遅延タイマーを短くしてもまだグループがプリエンプトする場合、上記のステップ 1 ～ 4 を繰り返して、2 台のルータ間でトラフィックが失われる期間を確認します。最小の遅延をトラフィック損失期間よりも長い時間に設定することで、プリエンプションを回避できます。最小の遅延を設定するには、次のコマンドを使用します。

```
RP/0/RSP0/CPU0:router(config)# router vrrp interface gigabitEthernet 0/2/0/10 vrrp delay
minimum 10 reload 5
```