



## イーサネット機能

このモジュールでは、Cisco IOS XR ソフトウェアをサポートする Cisco ASR 9000 シリーズ アグリゲーション サービス ルータでのレイヤ 2 (L2) イーサネット機能の設定方法について説明します。

イーサネット インターフェイスの設定の詳細については、このコンフィギュレーション ガイドの「[Cisco ASR 9000 シリーズ ルータ キャリア イーサネット モデル](#)」モジュールを参照してください。

### Cisco ASR 9000 シリーズ ルータ のイーサネット インターフェイス設定の機能履歴

リリース	変更内容
リリース 3.9.1	ポリシー ベースの転送およびレイヤ 2 プロトコル トンネリング機能のサポートが追加されました。

## 内容

- 「イーサネット機能を実装するための前提条件」 (P.61)
- 「イーサネットの機能の実装に関する情報」 (P.62)
- 「イーサネット機能の実装方法」 (P.69)
- 「設定例」 (P.75)
- 「その他の関連資料」 (P.78)

## イーサネット機能を実装するための前提条件

このコマンドリファレンスには、各コマンドに必要なタスク ID が含まれます。このコマンドリファレンスには、各コマンドに必要なタスク ID が含まれます。

ユーザ グループの割り当てが原因でコマンドを使用できないと考えられる場合、AAA 管理者に連絡してください。

# イーサネットの機能の実装に関する情報

10 ギガビット イーサネット インターフェイスを設定するには、次の概念を理解しておく必要があります。

- 「ポリシー ベースの転送」(P.62)
- 「レイヤ 2 プロトコル トンネリング」(P.62)

## ポリシー ベースの転送

Cisco ASR 9000 シリーズ ルータでは、単一の MAC アドレスを、ポートの設定済みの VLAN とは異なる VLAN にマップできます。2 つの異なる EFP に入るトラフィックを分離するためには、送信元 VLAN タグおよび送信元 MAC アドレスを使用して EFP を定義する必要があります。

## レイヤ 2 プロトコル トンネリング

レイヤ 2 プロトコル トンネリング (L2PT) は、レイヤ 2 (L2) スイッチング ドメイン間でイーサネット プロトコル フレームをトンネリングするための、シスコ独自のプロトコルです。

L2 プロトコル フレームが L2 スイッチング デバイスのインターフェイスに着信すると、スイッチまたはルータはフレームで次のいずれかのアクションを実行します。

- 転送：フレームは例外的な処理なしでスイッチングまたはルーティングされます。
- ドロップ：フレームはルータで廃棄されます。
- 終端：ルータは、フレームが L2 プロトコル フレームであると認識し、プロトコル処理のためにこれをルータのコントロール プレーンに送信します。
- トンネリング：ルータは、フレームをカプセル化して、プロトコル フレームとしてのアイデンティティを非表示にします。これにより、フレームが別のルータで終端することを防ぎます。トンネルの反対側ではカプセル化を解除して、フレームを元の状態に戻します。

## L2PT の機能

Cisco ASR 9000 シリーズ ルータ は、次の機能を提供します。

- 次のプロトコルをトンネリングします。
  - Cisco Discovery Protocol (CDP)
  - スパニングツリー プロトコル (STP およびそのバリエーション)
  - 仮想トランッキング プロトコル (VTP)
- 次のトンネリング モードをサポートします。
  - 進む
  - 反転
- L2PT は VLAN ヘッダーを持つプロトコル フレームをカプセル化し、カプセル化を解除します。
- 巨大フレーム レートの処理機能をサポートします。Cisco ASR 9000 シリーズ ルータは、インターフェイス ライン レートで L2PT カプセル化とカプセル化解除を実行します。



(注) 専用の L2PT カウンタはありません。QoS またはその他のパラメータの L2PT 特定の調整はありません。

## 転送モードの L2PT

図 1 に、転送モードで設定された L2PT を示します。

図 1 転送モードの L2PT

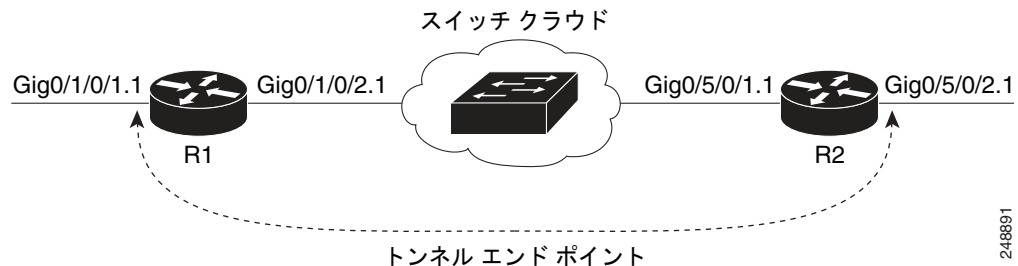


図 1 では、サービス プロバイダー ネットワーク (S ネットワーク) について説明します。カスタマー ネットワーク (C ネットワーク) は、GigabitEthernet サブインターフェイス 0/1/0/1.1 でルータ R1 に接続し、GigabitEthernet サブインターフェイス 0/5/0/2.1 でルータ R2 に接続します。C ネットワークは図に示されていません。ただし、C ネットワークは、S ネットワーク経由で L2 トラフィックを送信し、S ネットワークはエンド ツー エンドでトラフィックを切り替えます。カスタマー トラフィックは、L2 プロトコル フレームを伝送します。L2PT の目的は、これらのプロトコル フレームが S ネットワークを通過できるようにすることです。転送モードでは、L2PT は、S ネットワークのカスタマー側インターフェイスである R1 GigabitEthernet 0/1/0/1.1 と R2 GigabitEthernet 0/5/0/2.1 に適用されます。

図 1 は、転送モードの L2PT の設定を示します。

R1 :

```
!
interface GigabitEthernet0/1/0/1
 negotiation auto
!
interface GigabitEthernet0/1/0/1.1 l2transport
 encapsulation default
 l2protocol cpsv tunnel
!
interface GigabitEthernet0/1/0/2
 negotiation auto
!
interface GigabitEthernet0/1/0/2.1 l2transport
 encapsulation default
!
l2vpn
 xconnect group examples
 p2p r1-connect
 interface GigabitEthernet0/1/0/1.1
 interface GigabitEthernet0/1/0/2.1
!
```

```

R2 :
!
interface GigabitEthernet0/5/0/1
 negotiation auto
!
interface GigabitEthernet0/5/0/1.1 l2transport
 encapsulation default
!
interface GigabitEthernet0/5/0/2
 negotiation auto
!
interface GigabitEthernet0/5/0/2.1 l2transport
 encapsulation default
 l2protocol cpsv tunnel
!
l2vpn
 xconnect group examples
  p2p r2-connect
   interface GigabitEthernet0/5/0/1.1
   interface GigabitEthernet0/5/0/2.1
!
!
!
!

```

プロトコルトラフィックは、GigabitEthernet サブインターフェイス 0/1/0/1.1 でルータ R1 に入ります。ルータ R1 はプロトコル フレームとしてフレームを検出して、カスタマー側インターフェイスで L2PT カプセル化を実行します。R1 内では、ローカル接続 *r1-connect* は、R1 のカスタマー側インターフェイスとサービス プロバイダー側インターフェイスを接続します。トラフィックは、他の複数のサービス プロバイダー ネットワークのルータまたはスイッチ（スイッチクラウド）を介して GigabitEthernet サブインターフェイス 0/1/0/2.1 のルータ R1 から GigabitEthernet サブインターフェイス 0/5/0/1.1 のルータ R2 に通過します。ルータ R2 は、ローカル接続 *r2-connect* を介してカスタマー側インターフェイスとサービス プロバイダー側インターフェイスを接続します。したがって、トラフィックは、カスタマー側インターフェイスの GigabitEthernet 0/5/0/2.1 に送信されます。このインターフェイスで、L2PT のカプセル化が解除され、プロトコルトラフィックはルータ R2 からカスタマー ネットワークに流れます。

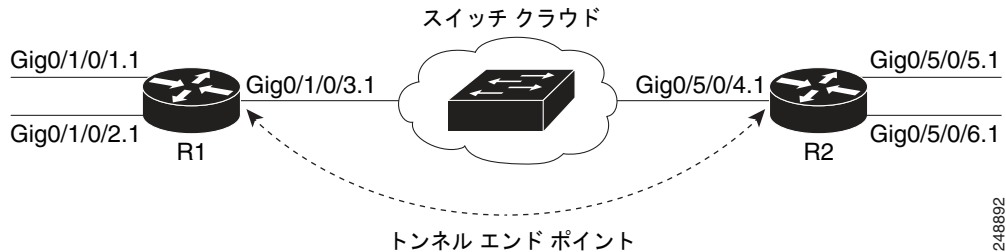
L2PT が設定されていない場合、R1 に送信されるカスタマー プロトコル フレームは終端します。カスタマー トラフィックは、さまざまなトラフィックで構成できます。プロトコル フレームは、全体的なトラフィック ストリームのうちわずかな割合で構成されます。

## プロトコル フレーム タギングを使用した反転モードの L2PT

Cisco ASR 9000 シリーズ ルータは、VLAN ヘッダーを持つサポートされている L2 プロトコル フレームで L2PT カプセル化およびカプセル化解除を実行できます。L2 プロトコル フレームに VLAN ヘッダーは含まれません。ただし、カスタマー キャンパス間でカスタマー プロトコルトラフィックを転送するサービス プロバイダー（SP）ネットワークでは、この機能を配置して、SP ネットワーク内で使用できます。

図 2 に、反転モードで設定された L2PT を示します。R1 に入るカスタマー トラフィックはトランキン グされており、すべてのトラフィックがタグ付きであると想定します。唯一のタグなしトラフィックは、カスタマー ネットワークから発信されるプロトコルトラフィックです。

図 2 反転モードの L2PT



反転モードで L2PT が設定されている場合、L2PT カプセル化は、フレームがインターフェイスを出ると行われます。同様に、反転モードのカプセル化解除は、フレームがインターフェイスに入ったときに実行されます。したがって、L2PT トンネルは、カスタマー側インターフェイスではなく、サービスプロバイダー側インターフェイス間で形成されます。

この例では、プロトコルトラフィックがルータ R1 に入ると、VLAN タグが追加されます。トラフィックがサービスプロバイダーネットワークを通じて送信される前に、2 番目の VLAN タグが追加されます (100)。Cisco ASR 9000 シリーズ ルータは、二重タグ付きプロトコルフレームで L2PT カプセル化を実行します。

図 2 に、4 つのカスタマー側インターフェイス (R1 : GigabitEthernet サブインターフェイス 0/1/0.1.1、GigabitEthernet サブインターフェイス 0/1/0/2.1 および R2 : GigabitEthernet サブインターフェイス 0/5/0/5.1、GigabitEthernet サブインターフェイス 0/5/0/6.1)、および 2 つのサービスプロバイダー側インターフェイス (R1 : GigabitEthernet サブインターフェイス 0/1/0/3.1 と R2 : GigabitEthernet サブインターフェイス 0/5/0/4.1) を示します。

図 2 は、反転モードの L2PT の設定を示します。

R1 :

```

!
interface GigabitEthernet0/1/0/1
 negotiation auto
!
interface GigabitEthernet0/1/0/1.1 l2transport
 encapsulation untagged
 rewrite ingress tag push dot1q 100 symmetric
 ethernet egress-filter strict
!
interface GigabitEthernet0/1/0/2
 negotiation auto
!
interface GigabitEthernet0/1/0/2.1 l2transport
 encapsulation untagged
 rewrite ingress tag push dot1q 200 symmetric
 ethernet egress-filter strict
!
interface GigabitEthernet0/1/0/3
 negotiation auto
!
interface GigabitEthernet0/1/0/3.1 l2transport
 encapsulation dot1q 500
 rewrite ingress tag pop 1 symmetric
 l2protocol cpsv reverse-tunnel
 ethernet egress-filter strict
!
l2vpn
 bridge group examples
 bridge-domain r1-bridge
 interface GigabitEthernet0/1/0/1.1

```

```

!
interface GigabitEthernet0/1/0/2.1
!
interface GigabitEthernet0/1/0/3.1
!
!
!
!
!

```

R2 :

```

!
interface GigabitEthernet0/5/0/4
 negotiation auto
!
interface GigabitEthernet0/5/0/4.1 l2transport
 encapsulation dot1q 500
 rewrite ingress tag pop 1 symmetric
 l2protocol cpsv reverse-tunnel
 ethernet egress-filter strict
!
interface GigabitEthernet0/5/0/5
 negotiation auto
!
interface GigabitEthernet0/5/0/5.1 l2transport
 encapsulation untagged
 rewrite ingress tag push dot1q 100 symmetric
 ethernet egress-filter strict
!
interface GigabitEthernet0/5/0/6
 negotiation auto
!
interface GigabitEthernet0/5/0/6.1 l2transport
 encapsulation untagged
 rewrite ingress tag push dot1q 200 symmetric
 ethernet egress-filter strict
!
l2vpn
 bridge group examples
 bridge-domain r2-bridge
 interface GigabitEthernet0/5/0/4.1
 !
 interface GigabitEthernet0/5/0/5.1
 !
 interface GigabitEthernet0/5/0/6.1
 !
 !
 !
 !
 !

```

次のことが前提となっています。

- ルータ R1 に入るカスタマー トラフィックはトランキングされます。つまり、すべてのトラフィックがタグ付けされています。唯一のタグなしトラフィックは、カスタマー ネットワークから到着するプロトコルトラフィックです。
- ルータ R1 の GigabitEthernet 0/1/0/1 とルータ R2 の GigabitEthernet 0/5/0/5 のカスタマー側インターフェイスは、同じカスタマーに属しています。ルータ R1 の GigabitEthernet 0/1/0/2 とルータ R2 の GigabitEthernet 0/5/0/6 のカスタマー側インターフェイスは、別のカスタマーに属していません。
- 異なるカスタマーからのトラフィックは分離されたままになります。
- L2 プロトコルトラフィックだけがカスタマー側インターフェイスを経由して送信されます。

- カスタマー側インターフェイスに入る L2 プロトコル トラフィックはタグなしです。
- トラフィックは、スイッチ クラウドを正常にパススルーするには、L2PT カプセル化されている必要があります。

このトポロジの目的は、ルータ R1 と R2 が複数のカスタマー インターフェイスからカスタマー プロトコル トラフィックを受信する必要があり、単一のサービス プロバイダー インターフェイスとリンク間でトラフィックを多重化する必要があることです。カプセル化解除の最後に、反転が実行されます。

GigabitEthernet サブインターフェイス 0/1/0/2.1 のルータ R1 に入るトラフィックは、GigabitEthernet サブインターフェイス 0/5/0/6.1 だけからルータ R2 を出るのに対して、GigabitEthernet サブインターフェイス 0/1/0/1.1 のルータ R1 に入るトラフィックは、GigabitEthernet サブインターフェイス 0/5/0/5.1 だけからルータ R2 を出します。

GigabitEthernet インターフェイス 0/1/0/1 のルータ R1 に入るプロトコル フレームは、この方法でネットワークを通過します。

- プロトコル フレームは、フレームがタグなしであるため、GigabitEthernet サブインターフェイス 0/1/0/1.1 に送信されます。
- GigabitEthernet サブインターフェイス 0/1/0/1.1 で rewrite ステートメントを使用すると、ID 100 のタグがフレームに追加されます。
- フレームは、ルータ R1 のブリッジ ドメイン r1-bridge に入ります。
- ブリッジ (r1-bridge) は、発信元 AC (スプリット ホライズン AC) を除き、ブリッジ ドメイン上のすべての接続回線 (AC) にフレームをフラッディングします。
- GigabitEthernet サブインターフェイス 0/1/0/2.1 でのイーサネット出力フィルタリングは、タグ ID のミスマッチを検出し、フレームをドロップします。このように、ブリッジ ドメインのフラッディングされたトラフィックは、他のカスタマー インターフェイスを出ることができません。
- フレームのフラッディングされたコピーは GigabitEthernet サブインターフェイス 0/1/0/3.1 に送信されます。
- GigabitEthernet サブインターフェイス 0/1/0/3.1 は 2 番目のタグを追加します。
- フレームは、GigabitEthernet インターフェイス 0/1/0/3 を介してルータ R1 を出る前に GigabitEthernet サブインターフェイス 0/1/0/3.1 によって L2PT カプセル化を受信します。



(注) 現在フレームには二重のタグが付いており (内部が 100、外部が 500) になっており、L2PT MAC DA があります。

- フレームは、L2PT カプセル化が原因で、ルータ R2 GigabitEthernet インターフェイス 0/5/0/4 に渡されます。
- フレームは、GigabitEthernet インターフェイス 0/5/0/4 のルータ R2 に入った後、GigabitEthernet サブインターフェイス 0/5/0/4.1 に送信されます。
- GigabitEthernet サブインターフェイス 0/5/0/4.1 に入るときに、L2PT カプセル解除動作がフレームで実行されます。
- 外部タグ ID 500 は、GigabitEthernet サブインターフェイス 0/5/0/4.1 によって削除されます。
- ルータ R2 のブリッジ (r2-bridge) は、すべての AC にフレームをフラッディングします。
- イーサネット出力フィルタリングは、フレームが出る AC を除くすべての AC でフレームをドロップします。
- フレームが GigabitEthernet サブインターフェイス 0/5/0/5.1 のルータ R2 を出るため、ID 100 のタグが削除されます。

- GigabitEthernet インターフェイス 0/5/0/5 のルータ R2 から出るフレームは、GigabitEthernet インターフェイス 0/1/0/1 を介してルータ R1 に入った元のフレームと同じです。

## L2PT 設定メモ

L2PT を設定する際は、次の点に注意してください。

- **l2protocol** コマンドは、メインまたは L2 のいずれかのサブインターフェイスで設定できます。
- **l2protocol** コマンドは、物理またはバンドル インターフェイスで設定できます。
- **l2protocol** および **ethernet filtering** コマンドが同じインターフェイスで設定されている場合、L2PT カプセル化はイーサネット フィルタリングの前に発生します。これは、L2PT によって、CDP、STP、および VTP プロトコル フレームがイーサネット フィルタリングによってドロップされないようにすることを意味します。
- L2PT が他のインターフェイス機能で設定されている場合、L2PT カプセル化は、他のインターフェイス機能の処理の前に発生します。
- L2PT カプセル化およびカプセル化解除は、タグなしプロトコル フレーム、一重タグ フレーム、および二重タグ付きフレームでサポートされます。タグ Ethertype 0x8100、0x88A8、および 0x9100 はサポートされていますが、0x9200 はサポートされていません。



# イーサネット機能の実装方法

この項では、次の作業について説明します。

- 「ポリシーベースの転送の設定」(P.69)
- 「レイヤ 2 プロトコル トンネリングの設定 : 例」(P.75)



(注) イーサネット インターフェイスの設定については、『Cisco ASR 9000 Series Aggregation Services Router Interface and Hardware Component Configuration Guide』を参照してください。

## ポリシーベースの転送の設定

この項では、次の手順について説明します。

- 「ポリシーベースの転送のイネーブル化」(P.69)
- 「送信元バイパス フィルタの設定」(P.72)

## ポリシーベースの転送のイネーブル化

ポリシーベースの転送をイネーブルにするには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **interface type interface-path-id.subinterface l2transport**
3. **encapsulation dot1q vlan-id ingress source-mac mac-address**  
または  
**encapsulation dot1ad vlan-id ingress source-mac mac-address**  
または  
**encapsulation untagged ingress source-mac mac-address**  
または  
**encapsulation dot1ad vlan-id dot1q vlan-id ingress source-mac mac-address**  
または  
**encapsulation dot1q vlan-id second-dot1q vlan-id ingress source-mac mac-address**
4. **rewrite ingress tag translate 1-to-1 dot1q vlan-id symmetric**  
または  
**rewrite ingress tag push dot1q vlan-id symmetric**
5. **ethernet egress-filter strict**
6. **end**  
または  
**commit**

## 手順の詳細

	コマンドまたはアクション	目的
ステップ1	<b>configure</b>  <b>例 :</b> RP/0/RSP0/CPU0:router# configure	グローバル コンフィギュレーション モードを開始します。
ステップ2	<b>interface type interface-path-id.subinterface</b> <b>l2transport</b>  <b>例 :</b> RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet 0/2/0/4.10 l2transport	サブインターフェイス コンフィギュレーション モードを開始し、ポートでレイヤ2 トランスポート モードをイネーブルにし、レイヤ2 トランスポート コンフィギュレーション モードを開始します。
ステップ3	<b>encapsulation dot1q vlan-id ingress source-mac</b> <b>mac-address</b> or <b>encapsulation dot1ad vlan-id ingress source-mac</b> <b>mac-address</b> or <b>encapsulation untagged ingress source-mac</b> <b>mac-address</b> or <b>encapsulation dot1ad vlan-id dot1q vlan-id</b> <b>ingress source-mac mac-address</b> or <b>encapsulation dot1q vlan-id second-dot1q</b> <b>vlan-id ingress source-mac mac-address</b>  <b>例 :</b> RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 10 ingress source-mac 0.1.2 or RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1ad 10 ingress source-mac 0.1.4 or RP/0/RSP0/CPU0:router(config-subif)# encapsulation untagged ingress source-mac 0.1.3 or RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1ad 10 dot1q 10 ingress source-mac 0.1.2 or RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 10 second-dot1q 20 ingress source-mac 0.1.2	一致する VLAN ID および EtherType をインターフェイスに割り当てます。

コマンドまたはアクション	目的
<p><b>ステップ4</b></p> <pre>rewrite ingress tag translate 1-to-1 dot1q vlan-id symmetric or rewrite ingress tag push dot1q vlan-id symmetric</pre> <p><b>例:</b>  RP/0/RSP0/CPU0:router(config-subif)# rewrite ingress tag translate 1-to-1 dot1q 100 symmetric or rewrite ingress tag push dot1q 101 symmetric</p>	<p>サービス インスタンスへのフレーム入力で行われるカプセル化調整を指定します。</p>
<p><b>ステップ5</b></p> <pre>ethernet egress-filter strict</pre> <p><b>例:</b>  RP/0/RSP0/CPU0:router(config-subif)# ethernet egress-filter strict</p>	<p>すべてのサブインターフェイスで厳密な出力フィルタリングをイネーブルにします。</p>
<p><b>ステップ6</b></p> <pre>end または commit</pre> <p><b>例:</b>  RP/0/RSP0/CPU0:router(config-subif)# end または RP/0/RSP0/CPU0:router(config-subif)# commit</p>	<p>設定変更を保存します。</p> <ul style="list-style-type: none"> <li>• <b>end</b> コマンドを実行すると、変更をコミットするように要求されます。  Uncommitted changes found, commit them before exiting (yes/no/cancel)?  [cancel]:</li> <li>– <b>yes</b> と入力すると、実行コンフィギュレーションファイルに設定変更が保存され、コンフィギュレーションセッションが終了し、ルータが EXEC モードに戻ります。</li> <li>– <b>no</b> と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。</li> <li>– <b>cancel</b> と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。</li> <li>• 実行コンフィギュレーションファイルに変更を保存し、コンフィギュレーションセッションを継続するには、<b>commit</b> コマンドを使用します。</li> </ul>

## 送信元バイパス フィルタの設定

送信元バイパス フィルタを追加するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **interface type interface-path-id.subinterface l2transport**
3. **encapsulation dot1q vlan-id**  
 または  
**encapsulation dot1ad vlan-id**  
 または  
**encapsulation untagged**  
 または  
**encapsulation dot1ad vlan-id dot1q vlan-id**  
 または  
**encapsulation dot1q vlan-id second-dot1q vlan-id**
4. **rewrite ingress tag translate 1-to-1 dot1q vlan-id symmetric**
5. **ethernet egress-filter disable**
6. **ethernet source bypass egress-filter**
7. **end**  
 または  
**commit**

### 手順の詳細

	コマンドまたはアクション	目的
ステップ1	<b>configure</b>  例： RP/0/RSP0/CPU0:router# configure	グローバル コンフィギュレーション モードを開始します。
ステップ2	<b>interface type interface-path-id.subinterface l2transport</b>  例： RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet 0/2/0/4.1 l2transport	サブインターフェイス コンフィギュレーション モードを開始し、ポートでレイヤ 2 トランスポート モードをイネーブルにし、レイヤ 2 トランスポート コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ3	<pre> encapsulation dot1q vlan-id or encapsulation dot1ad vlan-id or encapsulation untagged or encapsulation dot1ad vlan-id dot1q vlan-id or encapsulation dot1q vlan-id second-dot1q vlan-id  例： RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 10 or RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1ad 10 or RP/0/RSP0/CPU0:router(config-subif)# encapsulation untagged or RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1ad 10 dot1q 10 or RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 10 second-dot1q 20 </pre>	<p>一致する VLAN ID および EtherType をインターフェイスに割り当てます。</p>
ステップ4	<pre> rewrite ingress tag translate 1-to-1 dot1q vlan-id symmetric  例： RP/0/RSP0/CPU0:router(config-subif)# rewrite ingress tag translate 1-to-1 dot1q 100 symmetric </pre>	<p>サービス インスタンスへのフレーム入力で行われるカプセル化調整を指定します。</p>
ステップ5	<pre> ethernet egress-filter disable  例： RP/0/RSP0/CPU0:router(config-subif)# ethernet egress-filter strict </pre>	<p>すべてのサブインターフェイスで出力フィルタリングをディセーブルにします。</p>

コマンドまたはアクション	目的
<p>ステップ6 <code>ethernet source bypass egress-filter</code></p> <p><b>例:</b> RP/0/RSP0/CPU0:router(config-subif)# ethernet source bypass egress-filter</p>	<p>サブインターフェイスで送信元バイパス出力フィルタリングをイネーブルにします。</p>
<p>ステップ7 <code>end</code> または <code>commit</code></p> <p><b>例:</b> RP/0/RSP0/CPU0:router(config-subif)# end または RP/0/RSP0/CPU0:router(config-subif)# commit</p>	<p>設定変更を保存します。</p> <ul style="list-style-type: none"> <li>• <b>end</b> コマンドを実行すると、変更をコミットするように要求されます。  Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:</li> <li>- <b>yes</b> と入力すると、実行コンフィギュレーションファイルに設定変更が保存され、コンフィギュレーションセッションが終了し、ルータが EXEC モードに戻ります。</li> <li>- <b>no</b> と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。</li> <li>- <b>cancel</b> と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。</li> <li>• 実行コンフィギュレーションファイルに変更を保存し、コンフィギュレーションセッションを継続するには、<b>commit</b> コマンドを使用します。</li> </ul>

## 設定例

ここでは、次の設定例を示します。

- [ポリシーベースの転送の設定：例](#)
- [レイヤ2 プロトコル トンネリングの設定：例](#)

### ポリシーベースの転送の設定：例

次に、ポリシーベースの転送を設定する例を示します。

```
config
interface GigabitEthernet0/0/0/2.3 l2transport
encapsulation dot1q 10 ingress source-mac 0000.1111.2222
rewrite ingress tag translate 1-to-1 dot1q 100 symmetric
ethernet egress-filter strict
!
interface GigabitEthernet0/0/0/2.4 l2transport
encapsulation untagged ingress source-mac 0000.1111.3333
rewrite ingress tag push dot1q 101 symmetric
ethernet egress-filter strict
!

interface GigabitEthernet0/0/0/0/3.1 l2transport
encapsulation dot1q 1
rewrite ingress tag translate 1-to-1 dot1q 4094 symmetric
ethernet egress-filter disabled
ethernet source-bypass-egress-filter
!
```

### レイヤ2 プロトコル トンネリングの設定：例

ここでは、転送モードと反転モードでの L2PT の設定例を示します。

#### 転送モードでの L2PT の設定

次に、転送モードで L2PT を設定する例を示します。

カスタマー側ルータ（カプセル化側）：

```
!
interface GigabitEthernet0/1/0/1
negotiation auto
!
interface GigabitEthernet0/1/0/1.1 l2transport
encapsulation default
l2protocol cpsv tunnel
!
interface GigabitEthernet0/1/0/2
negotiation auto
!
interface GigabitEthernet0/1/0/2.1 l2transport
encapsulation default
!
l2vpn
xconnect group examples
```

```

p2p r1-connect
 interface GigabitEthernet0/1/0/1.1
 interface GigabitEthernet0/1/0/2.1
 !
 !
 !

```

カスタマー側ルータ（カプセル化解除側）：

```

!
interface GigabitEthernet0/5/0/1
 negotiation auto
!
interface GigabitEthernet0/5/0/1.1 l2transport
 encapsulation default
!
interface GigabitEthernet0/5/0/2
 negotiation auto
!
interface GigabitEthernet0/5/0/2.1 l2transport
 encapsulation default
 l2protocol cpsv tunnel
!
l2vpn
xconnect group examples
 p2p r2-connect
 interface GigabitEthernet0/5/0/1.1
 interface GigabitEthernet0/5/0/2.1
 !
 !
 !

```

## 反転モードでの L2PT の設定

次に、反転モードで L2PT を設定する例を示します。

カスタマー側ルータ（カプセル化側）：

```

!
interface GigabitEthernet0/1/0/1
 negotiation auto
!
interface GigabitEthernet0/1/0/1.1 l2transport
 encapsulation untagged
 rewrite ingress tag push dot1q 100 symmetric
 ethernet egress-filter strict
!
interface GigabitEthernet0/1/0/2
 negotiation auto
!
interface GigabitEthernet0/1/0/2.1 l2transport
 encapsulation untagged
 rewrite ingress tag push dot1q 200 symmetric
 ethernet egress-filter strict
!
interface GigabitEthernet0/1/0/3
 negotiation auto
!

```



```
interface GigabitEthernet0/1/0/3.1 l2transport
encapsulation dot1q 500
rewrite ingress tag pop 1 symmetric
l2protocol cpsv reverse-tunnel
ethernet egress-filter strict
!
l2vpn
bridge group examples
  bridge-domain r1-bridge
    interface GigabitEthernet0/1/0/1.1
    !
    interface GigabitEthernet0/1/0/2.1
    !
    interface GigabitEthernet0/1/0/3.1
    !
  !
!
!
カスタマー側ルータ（カプセル化解除側）：
!
interface GigabitEthernet0/5/0/4
negotiation auto
!
interface GigabitEthernet0/5/0/4.1 l2transport
encapsulation dot1q 500
rewrite ingress tag pop 1 symmetric
l2protocol cpsv reverse-tunnel
ethernet egress-filter strict
!
interface GigabitEthernet0/5/0/5
negotiation auto
!
interface GigabitEthernet0/5/0/5.1 l2transport
encapsulation untagged
rewrite ingress tag push dot1q 100 symmetric
ethernet egress-filter strict
!
interface GigabitEthernet0/5/0/6
negotiation auto
!
interface GigabitEthernet0/5/0/6.1 l2transport
encapsulation untagged
rewrite ingress tag push dot1q 200 symmetric
ethernet egress-filter strict
!
l2vpn
bridge group examples
  bridge-domain r2-bridge
    interface GigabitEthernet0/5/0/4.1
    !
    interface GigabitEthernet0/5/0/5.1
    !
    interface GigabitEthernet0/5/0/6.1
    !
  !
!
!
```

## その他の関連資料

ここでは、ギガビットおよび 10 ギガビット イーサネット インターフェイスの実装に関する参考資料を紹介します。

## 関連資料

関連項目	参照先
Cisco IOS XR マスター コマンド リファレンス	『Cisco IOS XR Master Commands List』

## 標準

標準	タイトル
この機能でサポートされる新規の標準または変更された標準はありません。また、既存の標準のサポートは変更されていません。	—

## MIB

MIB	MIB のリンク
このモジュールに適用できる MIB はありません。	Cisco IOS XR ソフトウェアを使用して、選択したプラットフォームの MIB を検索およびダウンロードするには、次の URL の Cisco MIB Locator を使用します。 <a href="http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml">http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml</a>

## RFC

RFC	タイトル
この機能によりサポートされた新規 RFC または改訂 RFC はありません。またこの機能による既存 RFC のサポートに変更はありません。	—

## シスコのテクニカル サポート

説明	リンク
シスコのテクニカル サポート Web サイトでは、製品、テクノロジー、ソリューション、技術的なヒント、およびツールへのリンクなどの、数千ページに及ぶ技術情報が検索可能です。Cisco.com に登録済みのユーザは、このページから詳細情報にアクセスできます。	<a href="http://www.cisco.com/en/US/support/index.html">http://www.cisco.com/en/US/support/index.html</a>