

IEEE 802.1Q Tunneling (QinQ) for AToM

This feature allows you to configure IEEE 802.1Q Tunneling (QinQ) for AToM. It also permits the rewriting of QinQ tags for Multiple Protocol Label Switching (MPLS) Layer 2 VPNs (L2VPNs).

- Prerequisites for IEEE 802.1Q Tunneling (QinQ) for AToM, on page 1
- Restrictions for IEEE 802.1Q Tunneling (QinQ) for AToM, on page 1
- Information About IEEE 802.1Q Tunneling (QinQ) for AToM, on page 2
- How to Configure IEEE 802.1Q Tunneling (QinQ) for AToM, on page 3
- Configuration Examples for IEEE 801.2 Tunneling (QinQ) for ATM, on page 12
- Additional References, on page 13
- Feature Information for IEEE 802.1Q Tunneling (QinQ) for AToM, on page 14

Prerequisites for IEEE 802.1Q Tunneling (QinQ) for AToM

The QinQ (short for 802.1Q-in-802.1Q) tunneling and tag rewrite feature is supported on the following line cards:

- 8-port Fast Ethernet line card (ESR-HH-8FE-TX)
- 2-port half-height Gigabit Ethernet line card (ESR-HH-1GE)
- 1-port full-height Gigabit Ethernet line card (ESR-1GE)

Restrictions for IEEE 802.1Q Tunneling (QinQ) for AToM

- Up to a maximum of 447 outer-VLAN IDs and up to 4095 inner VLAN IDs can be supported by this feature.
- Only Unambiguous VLAN tagged Ethernet QinQ interfaces are supported in this release. That is, the Ethernet VLAN QinQ rewrite of both VLAN Tags capability is supported only on Ethernet subinterfaces with a QinQ encapsulation and explicit pair of VLAN IDs defined.

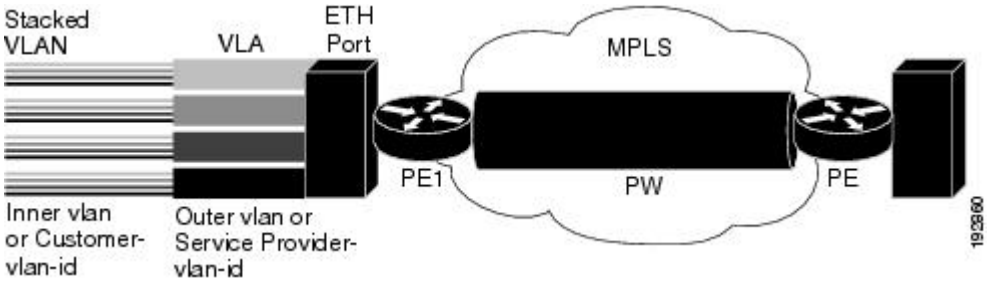
Note Ambiguous inner VLAN IDs are not supported in this release.

Information About IEEE 802.1Q Tunneling (QinQ) for AToM

Ethernet VLAN QinQ AToM

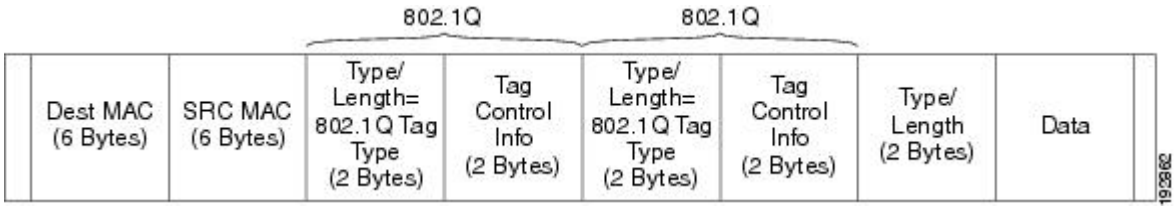
In Metro Ethernet deployment, in which CE routers and PE routers are connected through an Ethernet switched access network, packets that arrive at PE routers can contain up to two IEEE 802.1q VLAN tags (one inner VLAN tag which identifies the customer; and another outer VLAN tag which denotes the customer's service provider). This technique of allowing multiple VLAN tagging on the same Ethernet packet and creating a stack of VLAN IDs is known as QinQ (short for 802.1Q-in-802.1Q). The figure below shows how different edge devices can do L2 switching on the different levels of the VLAN stack.

Figure 1: Ethernet VLAN QinQ



When the outer VLAN tag is the service-delimiting VLAN tag, QinQ packets are processed similar to the ones with one VLAN tag (case previously named Ethernet VLAN Q-in-Q modified, which is already supported in the 12.2(31) SB release). However, when a customer must use a combination of the outer and inner VLAN tags to delimit service for customers, the edge device should be able to choose a unique pseudowire based on a combination of the inner and outer VLAN IDs on the packet shown in the figure below. The customer may want to be able to rewrite both the inner and the outer VLAN IDs on the traffic egress side.

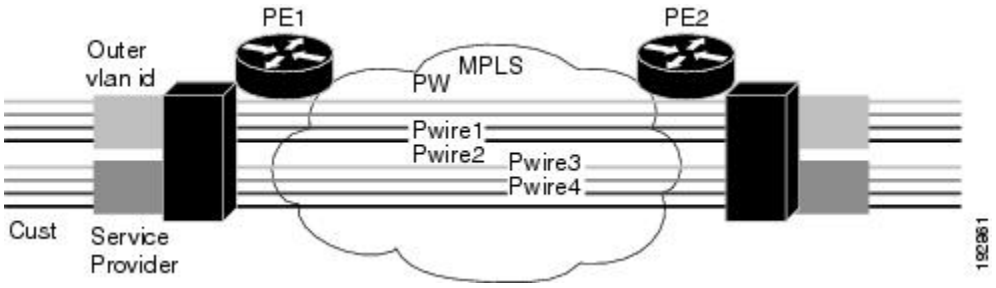
Figure 2: Ethernet VLAN QinQ Header



QinQ Tunneling Based on Inner and Outer VLAN Tags

When handling incoming QinQ Ethernet traffic, the edge router allows a customer to choose a unique pseudowire endpoint to switch the traffic based on the combination of inner and outer VLAN IDs. For example, the figure below shows how a unique pseudowire is selected depending upon the combination of inner (customer edge) and outer (service provider) VLAN IDs. Thus, traffic for different customers can be kept separate.

Figure 3: QinQ Connection



Rewritten Inner and Outer VLAN Tags on QinQ Frames

When managing incoming AToM Ethernet QinQ traffic, the edge router does the following tasks:

1. Strips off the MPLS labels.
2. Allows the customer to rewrite both the inner and outer VLAN IDs before sending the packets to the egress QinQ interface. Note this capability is provided only for AToM like-to-like Ethernet QinQ traffic.

The QinQ AToM feature is a like-to-like interworking case over AToM. This feature requires changes to the microcode to allow it to overwrite two layers of VLAN tags on Ethernet QinQ traffic, transported across AToM pseudowires.

- On the ingress side--The packets preserve their L2 header with the two VLAN tags, and it is sent across the pseudowire with VC type of 4.
- On the egress side--The MPLS label is stripped, and up to two levels of VLAN tags are rewritten per the configuration.

Only Unambiguous VLAN tagged Ethernet QinQ interfaces are supported in this release. The Ethernet VLAN Q-in-Q rewrite of both VLAN Tags capability is supported only on Ethernet subinterfaces with a QinQ encapsulation and explicit pair of VLAN IDs defined.

How to Configure IEEE 802.1Q Tunneling (QinQ) for AToM

This section explains how to configure IEEE 802.1Q Tunneling (QinQ) for AToM and includes the following procedures. While all of the procedures are listed as optional, you must choose one of the first two listed.

Configuring Unambiguous IEEE 802.1Q Tunneling (QinQ) for AToM

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet** *slot* / *subslot* / *port* **.** [*subinterface*]
4. **encapsulation dot1q** *vlan-id* **second-dot1q** {**any** | *vlan-id*[,*vlan-id*[*-vlan-id*]]}
5. **xconnect** *peer-router-id vcid* **encapsulation mpls**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface gigabitethernet** *slot* **/** *subslot* **/** *port* **.** [*subinterface*] **Example:** `Router(config)# interface GigabitEthernet1/0/0.100`	Specifies the Gigabit Ethernet interface and enters interface configuration mode.
Step 4	**encapsulation dot1q** *vlan-id* **second-dot1q** {**any** | *vlan-id*[,*vlan-id*[-*vlan-id*]]} **Example:** `Router(config-if)# encapsulation dot1q 100 second-dot1q 200`	Defines the matching criteria to map Q-in-Q ingress frames on an interface to the appropriate service instance.
Step 5	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-if)# xconnect 10.0.0.16 410 encapsulation mpls`	Creates the VC to transport the Layer 2 packets.

Configuring Unambiguous IEEE 802.1Q Tunneling (QinQ) for AToM using the commands associated with the L2VPN Protocol-Based CLIs feature

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet** *slot* **/** *subslot* **/** *port* **.** [*subinterface*]
4. **encapsulation dot1q** *vlan-id* **second-dot1q** {**any** | *vlan-id*[,*vlan-id*[-*vlan-id*]]}
5. **interface pseudowire** *number*
6. **encapsulation mpls**
7. **neighbor** *peer-address vcid-value*

8. **exit**
9. **l2vpn xconnect context** *context-name*
10. **member pseudowire** *interface-number*
11. **member gigabitethernet** *interface-number*
12. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface gigabitethernet** *slot* / *subslot* / *port* **.** [*subinterface*] **Example:** `Router(config)# interface GigabitEthernet1/0/0.100`	Specifies the Gigabit Ethernet interface and enters interface configuration mode.
Step 4	**encapsulation dot1q** *vlan-id* **second-dot1q** {**any** | *vlan-id*[,*vlan-id*[-*vlan-id*]]} **Example:** `Router(config-if)# encapsulation dot1q 100 second-dot1q 200`	Defines the matching criteria to map Q-in-Q ingress frames on an interface to the appropriate service instance.
Step 5	**interface pseudowire** *number* **Example:** `Router(config-if)# interface pseudowire 100`	Specifies the pseudowire interface and enters interface configuration mode.
Step 6	**encapsulation mpls** **Example:** `Router(config-if)# encapsulation mpls`	Specifies that Multiprotocol Label Switching (MPLS) is used as the data encapsulation method.
Step 7	**neighbor** *peer-address vcid-value* **Example:** `Router(config-if)# neighbor 10.0.0.1 123`	Specifies the peer IP address and virtual circuit (VC) ID value of the Layer 2 VPN (L2VPN) pseudowire.

	Command or Action	Purpose
Step 8	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 9	**l2vpn xconnect context** *context-name* **Example:** `Router(config)# l2vpn xconnect context con1`	Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.
Step 10	**member pseudowire** *interface-number* **Example:** `Router(config-xconnect)# member pseudowire 100`	Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.
Step 11	**member gigabitethernet** *interface-number* **Example:** `Router(config-xconnect)# member GigabitEthernet1/0/0.100`	Specifies the location of the Gigabit Ethernet member interface.
Step 12	**end** **Example:** `Router(config-xconnect)# end`	Exits to privileged EXEC mode.

Configuring Ambiguous IEEE 802.1Q Tunneling (QinQ) for AToM

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet** *slot* **/** *subslot* **/** *port* **.** [*subinterface*]
4. **encapsulation dot1q** *vlan-id* **second-dot1q** {**any** | *vlan-id*[,*vlan-id*[-*vlan-id*]]}
5. **xconnect** *peer-router-id vcid encapsulation mpls*
6. **exit**
7. **interface gigabitethernet** *slot* **/** *subslot* **/** *port* **.** [*subinterface*]
8. **encapsulation dot1q** *vlan-id* **second-dot1q** {**any** | *vlan-id*[,*vlan-id*[-*vlan-id*]]}
9. **xconnect** *peer-router-id vcid encapsulation mpls*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface gigabitethernet** *slot* / *subslot* / *port* **.** [*subinterface*] **Example:** `Router(config)# interface GigabitEthernet1/0/0.200`	Specifies the Gigabit Ethernet subinterface and enters interface configuration mode.
Step 4	**encapsulation dot1q** *vlan-id* **second-dot1q** {**any** | *vlan-id*[,*vlan-id*[*-vlan-id*]]} **Example:** `Router(config-if)# encapsulation dot1q 200 second-dot1q 1000-2000,3000,3500-4000`	Defines the matching criteria to map Q-in-Q ingress frames on an interface to the appropriate service instance.
Step 5	**xconnect** *peer-router-id vcid encapsulation mpls* **Example:** `Router(config-if)# xconnect 10.0.0.16 420 encapsulation mpls`	Creates the VC to transport the Layer 2 packets.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 7	**interface gigabitethernet** *slot* / *subslot* / *port* **.** [*subinterface*] **Example:** `Router(config)# interface GigabitEthernet1/0/0.201`	Specifies the next Gigabit Ethernet interface and enters interface configuration mode.
Step 8	**encapsulation dot1q** *vlan-id* **second-dot1q** {**any** | *vlan-id*[,*vlan-id*[*-vlan-id*]]} **Example:**	Defines the matching criteria to map Q-in-Q ingress frames on an interface to the appropriate service instance.

	Command or Action	Purpose
	`Router(config-if)# encapsulation dot1q 201 second-dot1q any`	
Step 9	**xconnect** *peer-router-id vcid encapsulation mpls* **Example:** `Router(config-if)# xconnect 10.0.0.16 430 encapsulation mpls`	Creates the VC to transport the Layer 2 packets.

Configuring Ambiguous IEEE 802.1Q Tunneling (QinQ) for AToM using the commands associated with the L2VPN Protocol-Based CLIs feature

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet** *slot* **/** *subslot* **/** *port* **.** [*subinterface*]
4. **encapsulation dot1q** *vlan-id* **second-dot1q** {**any** | *vlan-id*[,*vlan-id*[-*vlan-id*]]}
5. **interface pseudowire** *number*
6. **encapsulation mpls**
7. **neighbor** *peer-address vcid-value*
8. **exit**
9. **interface gigabitethernet** *slot* **/** *subslot* **/** *port* **.** [*subinterface*]
10. **encapsulation dot1q** *vlan-id* **second-dot1q** {**any** | *vlan-id*[,*vlan-id*[-*vlan-id*]]}
11. **interface pseudowire** *number*
12. **encapsulation mpls**
13. **neighbor** *peer-address vcid-value*
14. **exit**
15. **l2vpn xconnect context** *context-name*
16. **member pseudowire** *interface-number*
17. **member gigabitethernet** *interface-number*
18. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface gigabitethernet** *slot* / *subslot* / *port* **.** [*subinterface*] **Example:** `Router(config)# interface GigabitEthernet1/0/0.200`	Specifies the Gigabit Ethernet subinterface and enters interface configuration mode.
Step 4	**encapsulation dot1q** *vlan-id* **second-dot1q** {**any** | *vlan-id*[,*vlan-id*[-*vlan-id*]]} **Example:** `Router(config-if)# encapsulation dot1q 200 second-dot1q 1000-2000,3000,3500-4000`	Defines the matching criteria to map Q-in-Q ingress frames on an interface to the appropriate service instance.
Step 5	**interface pseudowire** *number* **Example:** `Router(config-if)# interface pseudowire 100`	Specifies the pseudowire interface and enters interface configuration mode.
Step 6	**encapsulation mpls** **Example:** `Router(config-if)# encapsulation mpls`	Specifies that Multiprotocol Label Switching (MPLS) is used as the data encapsulation method.
Step 7	**neighbor** *peer-address vcid-value* **Example:** `Router(config-if)# neighbor 10.0.0.1 123`	Specifies the peer IP address and virtual circuit (VC) ID value of the Layer 2 VPN (L2VPN) pseudowire.
Step 8	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 9	**interface gigabitethernet** *slot* / *subslot* / *port* **.** [*subinterface*] **Example:** `Router(config)# interface GigabitEthernet1/0/0.201`	Specifies the next Gigabit Ethernet interface and enters interface configuration mode.
Step 10	**encapsulation dot1q** *vlan-id* **second-dot1q** {**any** | *vlan-id*[,*vlan-id*[-*vlan-id*]]} **Example:**	Defines the matching criteria to map Q-in-Q ingress frames on an interface to the appropriate service instance.

	Command or Action	Purpose
	`Router(config-if)# encapsulation dot1q 201 second-dot1q any`	
Step 11	**interface pseudowire** *number* **Example:** `Router(config-if)# interface pseudowire 100`	Specifies the pseudowire interface and enters interface configuration mode.
Step 12	**encapsulation mpls** **Example:** `Router(config-if)# encapsulation mpls`	Specifies that Multiprotocol Label Switching (MPLS) is used as the data encapsulation method.
Step 13	**neighbor** *peer-address vcid-value* **Example:** `Router(config-if)# neighbor 10.0.0.1 123`	Specifies the peer IP address and virtual circuit (VC) ID value of the Layer 2 VPN (L2VPN) pseudowire.
Step 14	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 15	**l2vpn xconnect context** *context-name* **Example:** `Router(config)# l2vpn xconnect context con1`	Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.
Step 16	**member pseudowire** *interface-number* **Example:** `Router(config-xconnect)# member pseudowire 100`	Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.
Step 17	**member gigabitethernet** *interface-number* **Example:** `Router(config-xconnect)# member GigabitEthernet1/0/0.201`	Specifies the location of the Gigabit Ethernet member interface.
Step 18	**end** **Example:** `Router(config-xconnect)# end`	Exits to privileged EXEC mode.

Verifying the IEEE 802.1Q Tunneling (QinQ) for ATM Configuration

SUMMARY STEPS

1. **enable**
2. **show mpls l2transport vc**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show mpls l2transport vc** **Example:** `Router# show mpls l2transport vc`	Displays information about Any Transport over MPLS (AToM) virtual circuits (VCs) and static pseudowires that have been enabled to route Layer 2 packets on a router.

Verifying the IEEE 802.1Q Tunneling (QinQ) for ATM Configuration using the commands associated with the L2VPN Protocol-Based CLIs feature

SUMMARY STEPS

1. **enable**
2. **show l2vpn atom vc**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show l2vpn atom vc** **Example:** `Device# show l2vpn atom vc`	Displays information about Any Transport over MPLS (AToM) virtual circuits (VCs) and static pseudowires that have been enabled to route Layer 2 packets on a router.

Configuration Examples for IEEE 801.2 Tunneling (QinQ) for ATM

Example Configuring Unambiguous IEEE 802.1Q Tunneling (QinQ) for ATM

```
Router> enable
Router# configure terminal
Router(config)# interface GigabitEthernet1/0/0.100
Router(config-if)# encapsulation dot1q 100 second-dot1q 200
Router(config-if)# xconnect 10.0.0.16 410 encapsulation mpls
```

Example Configuring Unambiguous IEEE 802.1Q Tunneling (QinQ) for ATM using the commands associated with the L2VPN Protocol-Based CLIs feature

```
Router> enable
Router# configure terminal
Router(config)# interface GigabitEthernet1/0/0.100
Router(config-if)# encapsulation dot1q 100 second-dot1q 200
Router(config-if)# interface pseudowire 100
Router(config-if)# encapsulation mpls
Router(config-if)# neighbor 10.0.0.1 123
Router(config-if)# exit
Router(config)# l2vpn xconnect context A
Router(config-xconnect)# member pseudowire 100
Router(config-xconnect)# member GigabitEthernet1/0/0.100
```

Example Configuring Ambiguous IEEE 802.1Q Tunneling (QinQ) for ATM

The following is an example of an ambiguous IEEE 802.1Q Tunneling (QinQ) for ATM configuration.

```
Router> enable
Router# configure terminal
Router(config)# interface GigabitEthernet1/0/0.200
Router(config-if)# encapsulation dot1q 200 second-dot1q 1000-2000,3000,3500-4000
Router(config-if)# xconnect 10.0.0.16 420 encapsulation mpls
Router(config-if)# exit
Router(config)# interface GigabitEthernet1/0/0.201
Router(config-if) encapsulation dot1q 201 second-dot1q any
Router(config-if) xconnect 10.0.0.16 430 encapsulation mpls
```

Example Configuring Ambiguous IEEE 802.1Q Tunneling (QinQ) for ATM using the commands associated with the L2VPN Protocol-Based CLIs feature

The following is an example of an ambiguous IEEE 802.1Q Tunneling (QinQ) for ATM configuration.

```
Router> enable
Router# configure terminal
Router(config)# interface GigabitEthernet1/0/0.200
Router(config-if)# encapsulation dot1q 200 second-dot1q 1000-2000,3000,3500-4000
```

```
Router(config-if)# interface pseudowire 100
Router(config-if)# encapsulation mpls
Router(config-if)# neighbor 10.0.0.1 123
Router(config-if)# exit
Router(config)# l2vpn xconnect context A
Router(config-xconnect)# member pseudowire 100
Router(config-xconnect)# member GigabitEthernet1/0/0.200
Router(config-xconnect)# exit
Router(config)# interface GigabitEthernet1/0/0.201
Router(config-if) encapsulation dot1q 201 second-dot1q any
Router(config-if)# interface pseudowire 100
Router(config-if)# encapsulation mpls
Router(config-if)# neighbor 10.0.0.1 123
Router(config-if)# exit
Router(config)# l2vpn xconnect context A
Router(config-xconnect)# member pseudowire 100
Router(config-xconnect)# member GigabitEthernet1/0/0.201
```

Example Verifying the IEEE 802.1Q Tunneling (QinQ) for ATM Configuration

The following is sample output of the **show mpls l2transport vc** command, which is usedto verify the VC set up in EoMPLS QinQ mode.

```
router# show mpls l2transport vc
Local intf     Local circuit              Dest address    VC ID      Status
-------------  -------------------------- --------------- ---------- ----------
Gi1/0/0.1      Eth VLAN:100/200           10.1.1.2        1          UP
```

Example Verifying the IEEE 802.1Q Tunneling (QinQ) for ATM Configuration using the commands associated with the L2VPN Protocol-Based CLIs feature

The following is sample output of the **show l2vpn atom vc** command, which is used to verify the virtual circuit (VC) set up in EoMPLS QinQ mode.

```
Device# show l2vpn atom vc
Local intf     Local circuit              Dest address    VC ID      Status
-------------  -------------------------- --------------- ---------- ----------
Gi1/0/0.1      Eth VLAN:100/200           10.1.1.2        1          UP
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Description of commands associated with MPLS and MPLS applications	*Cisco IOS Multiprotocol Label Switching Command Reference*
AToM and MPLS	Any Transport over MPLS

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IEEE 802.1Q Tunneling (QinQ) for AToM

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1: Feature Information for IEEE 802.1Q Tunneling (QinQ) for AToM

Feature Name	Releases	Feature Information
IEEE 802.1Q Tunneling (QinQ) for AToM	Cisco IOS XE Release 2.4	This feature allows you to configure IEEE 802.1Q Tunneling (QinQ) for AToM. It also permits the rewriting of QinQ tags for Multiple Protocol Label Switching (MPLS) layer 2 VPNs (L2VPNs). In Cisco IOS XE Release 2.4, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified: **interface , encapsulation dot1q second-dot1q , xconnect .**