Configuring IEEE 802.1ad

Provider networks handle traffic from a large number of customers. It is important that one customer's traffic is isolated from the other customer's traffic. IEEE 802.1ad implements standard protocols for double tagging of data. The data traffic coming from the customer side are double tagged in the provider network where the inner tag is the customer-tag (C-tag) and the outer tag is the provider-tag (S-tag). The control packets are tunneled by changing the destination MAC address in the provider network.

Cisco 7600 series routers already support VLAN double tagging through a feature called QinQ. 802.1ad is the standardized version of QinQ. It also extends the support for Layer 2 Protocol Tunneling Protocol (L2PT). By offering transparent Layer 2 connectivity, the service provider does not get involved in the customer's Layer 3 network. This makes provisioning and maintenance simple, and reduces the operational cost.

Prerequisites for IEEE 802.1ad

- The ethertype should be programmable per port.

Restrictions for IEEE 802.1ad

Follow these restrictions and guidelines when you configure 802.1ad:

- The **l2protocol forward** command is available only on the main interface of switchports and L3 ports. The command is not available on the subinterfaces. All the subinterfaces on a port inherit the behavior from the main interface. The **l2protocol forward** command is also available on EVC service instance.
- The **l2protocol peer** and **l2protocol drop** commands are not supported.
- The **l2protocol forward** command on a main interface and on EVCs supports only cdp, dtp, vtp, stp, and dot1x.
- You cannot configure Dot1ad if custom ethertype is configured on port.

- 802.1ad is supported on the following port types:

Port	EVC	Switchport	Layer Interfaces
C-UNI	Ethertype 0x8100 C-VLAN BPDU Any EVCs	Ethertype 0x8100 C-VLAN BPDU Trunk or Access	Ethertype 0x8100 C-VLAN BPDU
S-UNI	Ethertype 0x88a8 S-VLAN BPDU (Only Encapsulation default is supported)	Ethertype 0x88a8 S-VLAN BPDU Access only	Not supported
S-NNI	Ethertype 0x88a8 S-VLAN BPDU Any EVC	Ethertype 0x88a8 S-VLAN BPDU Trunk	Ethertype 0x88a8 S-VLAN BPDU Trunk

Information About IEEE 802.1ad

To configure IEEE 802.1ad support, you should understand the following concepts:

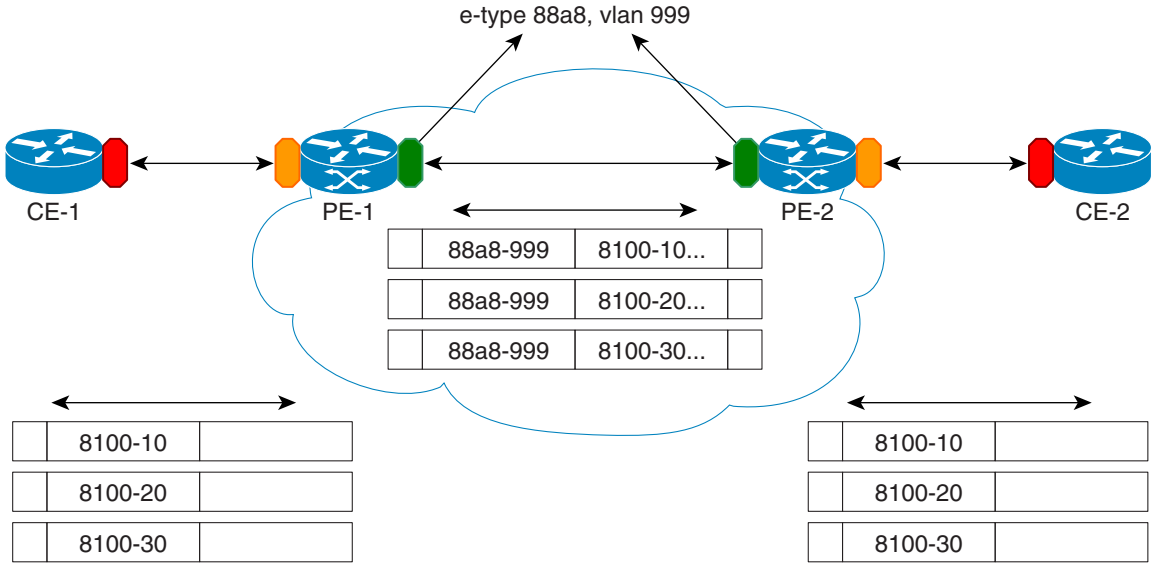
- How Provider Bridges Work
- Guidelines for Handling BPDU
- Interoperability of QinQ and Dot1ad

How Provider Bridges Work

Provider bridges pass the network traffic of many customers, and each customer's traffic flow must be isolated from one another. For the Layer 2 protocols within customer domains to function properly, geographically separated customer sites must appear to be connected through a LAN, and the provider network must be transparent.

The IEEE has reserved 33 Layer 2 MAC addresses for customer devices operating Layer 2 protocols. If a provider bridge uses these standard MAC addresses for its Layer 2 protocols, the customers' and service provider's Layer 2 traffic will be mixed together. Provider bridges solve this traffic-mixing issue by providing Layer 2 protocol data unit (PDU) tunneling for customers using a provider bridge (S-bridge) component and a provider edge bridge (C-bridge) component. Figure 53-1 shows the topology.

Figure 53-1 Layer 2 PDU Tunneling



S-Bridge Component

The S-bridge component is capable of inserting or removing a service provider VLAN (S-VLAN) for all traffic on a particular port. IEEE 802.1ad adds a new tag called a Service tag (S-tag) to all the ingress frames from a customer to the service provider.

The VLAN in the S-tag is used for forwarding the traffic in the service provider network. Different customers use different S-VLANs, which results in each customer's traffic being isolated. In the S-tag, provider bridges use an Ethertype value that is different from the standard 802.1Q Ethertype value, and do not understand the standard Ethertype. This difference makes customer traffic tagged with the standard Ethertype appear as untagged in the provider network so customer traffic is tunneled in the port VLAN of the provider port. The 802.1ad service provider user network interfaces (S-UNIs) and network to network interfaces (NNIs) implement the S-bridge component.

For example, a VLAN tag has a VLAN ID of 1, the C-tag Ethertype value is 8100 0001, the S-tag Ethertype value is 88A8 0001, and the class of service (CoS) is zero.

C-tag S-tag

--- ---

0x8100 | Priority bits | CFI | C-VLAN-ID 0x88A8 | Priority bits | 0 | S-VLAN-ID

--- ---

C-Bridge Component

All the C-VLANs entering on a UNI port in an S-bridge component are provided the same service (marked with the same S-VLAN). Although, C-VLAN components are not supported, a customer may want to tag a particular C-VLAN packet separately to differentiate between services. Provider bridges allow C-VLAN packet tagging with a provider edge bridge, called the C-bridge component of the provider bridge. C-bridge components are C-VLAN aware and can insert or remove a C-VLAN 802.1Q tag. The C-bridge UNI port is capable of identifying the customer 802.1Q tag and inserting or removing

an S-tag on the packet on a per service instance or C-VLAN basis. A C-VLAN tagged service instance allows service instance selection and identification by C-VLAN. The 802.1ad customer user network interfaces (C-UNIs) implement the C-component.

MAC Addresses for Layer 2 Protocols

Customers' Layer 2 PDUs received by a provider bridge are not forwarded, so Layer 2 protocols running in customer sites do not know the complete network topology. By using a different set of addresses for the Layer 2 protocols running in provider bridges, IEEE 802.1ad causes customers' Layer 2 PDUs entering the provider bridge to appear as unknown multicast traffic and forwards it on customer ports (on the same S-VLAN). Customers' Layer 2 protocols can then run transparently.

Table 53-1 shows the Layer 2 MAC addresses reserved for the C-VLAN component.

Table 53-1 Reserved Layer 2 MAC Addresses for a C-VLAN Component

Assignment	Value
Bridge Group Address	01-80-c2-00-00-00
IEEE Std 802.3 Full Duplex PAUSE operation	01-80-c2-00-00-01
IEEE Std. 802.3 Slow_Protocols_Multicast address	01-80-c2-00-00-02
IEEE Std. 802.1X PAE address	01-80-c2-00-00-03
Reserved for future standardization - media access method-specific	01-80-c2-00-00-04
Reserved for future standardization - media access method- specific	01-80-c2-00-00-05
Reserved for future standardization	01-80-c2-00-00-06
Reserved for future standardization	01-80-c2-00-00-07
Provider Bridge Group Address	01-80-c2-00-00-08
Reserved for future standardization	01-80-c2-00-00-09
Reserved for future standardization	01-80-c2-00-00-0a
Reserved for future standardization	01-80-c2-00-00-0b
Reserved for future standardization	01-80-c2-00-00-0c
Provider Bridge GVRP Address	01-80-c2-00-00-0d
IEEE Std. 802.1AB Link Layer Discovery Protocol multicast address	01-80-c2-00-00-0e
Reserved for future standardization	01-80-c2-00-00-0f

Table 53-2 shows the Layer 2 MAC addresses reserved for an S-VLAN component. These addresses are a subset of the C-VLAN component addresses, and the C-bridge does not forward the provider's bridge protocol data units (BPDUs) to a customer network.

Table 53-2 Reserved Layer 2 MAC Addresses for an S-VLAN Component

Assignment	Value
IEEE Std 802.3 Full Duplex PAUSE operation	01-80-c2-00-00-01
IEEE Std. 802.3 Slow_Protocols_Multicast address	01-80-c2-00-00-02
IEEE Std. 802.1X PAE address	01-80-c2-00-00-03
Reserved for future standardization - media access method specific	01-80-c2-00-00-04
Reserved for future standardization - media access method specific	01-80-c2-00-00-05
Reserved for future standardization	01-80-c2-00-00-06
Reserved for future standardization	01-80-c2-00-00-07
Provider Bridge Group Address	01-80-c2-00-00-08
Reserved for future standardization	01-80-c2-00-00-09
Reserved for future standardization	01-80-c2-00-00-0a

Guidelines for Handling BPDU

The general BPDU guidelines are listed here:

UNI-C Ports

The guidelines pertaining to UNI-C ports are:

- VLAN-aware L2 protocols can be peered, tunneled, or dropped.
- Port L2 protocols can either be peered or dropped. They cannot be tunneled.

Table 53-3 shows the Layer 2 PDU destination MAC addresses for customer-facing C-bridge UNI ports, and how frames are processed.

Table 53-3 Layer 2 PDU Destination MAC Addresses for Customer-Facing C-Bridge UNI Ports

Assignment	Protocol	Significance on C-UNI Port	Default Action
01-80-C2-00-00-00	Bridge Group Address (End-to-End BPDUs)	BPDU	Peer
01-80-C2-00-00-01	802.3X Pause Protocol	BPDU	Drop
01-80-C2-00-00-02	Slow Protocol address: 802.3ad LACP, 802.3ah OAM, CDP Pagp, VTP, DTP, UDLD	BPDU	Peer

Table 53-3 Layer 2 PDU Destination MAC Addresses for Customer-Facing C-Bridge UNI Ports

Assignment	Protocol	Significance on C-UNI Port	Default Action
01-80-C2-00-00-03	802.1X	BPDU	May peer
01-80-C2-00-00-04	Reserved for future media access method	None	Drop
01-80-C2-00-00-05	Reserved for future media access method	None	Drop
01-80-C2-00-00-06	Reserved for future bridge use	None	Drop
01-80-C2-00-00-07	Reserved for future bridge use	None	Drop
01-80-C2-00-00-08	Provider STP (BPDU)	None	Drop
01-80-C2-00-00-09	Reserved for future bridge use	None	Drop
01-80-C2-00-00-0A	Reserved for future bridge use	None	Drop
01-80-C2-00-000-0B	Reserved for future S-bridge purpose	None	Drop
01-80-C2-00-00-0C	Reserved for future S-bridge purpose	None	Drop
01-80-C2-00-00-0D	Provider Bridge GVRP address	None	Drop
01-80-C2-00-00-0E	802.1ab-LLDP	BPDU	May peer
01-80-C2-00-00-0F	Reserved for future C-bridge or Q-bridge use	None	Drop
01-80-C2-00-00-10	All bridge addresses	Read Data	Snoop if implemented. Else, discard
01-80-C2-00-00-20	GMRP	Data/BPDU	May peer
01-80-C2-00-00-21	GVRP	Data/BPDU	May peer
01-80-C2-00-00-22 – 2F	Other GARP addresses	Data/BPDU	May peer
01-00-0C-CC-CC-CC	Cisco’s CDP DTP VTP PagP UDLD (End-to-End)	BPDU	Peer
01-00-0C-CC-CC-CD	Cisco’s PVST(End-to-End)	BPDU	May peer

UNI-S Ports

The guidelines pertaining to UNI-S ports are:

- Packets with C-Bridge addresses (00 - 0F) that are not part of S-Bridge addresses (01 - 0A) are treated as data packet (tunneled).
- VLAN-aware L2 protocols cannot be peered because the port is not C-VLAN aware. They can only be tunneled or dropped.
- Port L2 protocols can be peered, tunneled, or dropped.

Table 53-4 shows the Layer 2 PDU destination MAC addresses for customer-facing S-bridge UNI ports, and how frames are processed.

Table 53-4 Layer 2 PDU Destination MAC Addresses for Customer-Facing S-Bridge UNI Ports

Assignment	Protocol	Significance on S-UNI Port	Default Action
01-80-C2-00-00-00	Bridge Group Address (BPDUs)	Data	Data
01-80-C2-00-00-01	802.3X Pause Protocol	BPDU	Drop
01-80-C2-00-00-02	Slow Protocol address: 802.3ad LACP, 802.3ah	BPDU	Peer
01-80-C2-00-00-03	802.1X	BPDU	Peer
01-80-C2-00-00-04	Reserved for future media access method	BPDU	Drop
01-80-C2-00-00-05	Reserved for future media access method	BPDU	Drop
01-80-C2-00-00-06	Reserved for future bridge use	BPDU	Drop
01-80-C2-00-00-07	Reserved for future bridge use	BPDU	Drop
01-80-C2-00-00-08	Provider STP (BPDU)	BPDU	Drop (peer on NNI)
01-80-C2-00-00-09	Reserved for future bridge use	BPDU	Drop
01-80-C2-00-00-0A	Reserved for future bridge use	BPDU	Drop
01-80-C2-00-00-0B	Reserved for future bridge use	Data if not implemented	Drop
01-80-C2-00-00-0C	Reserved for future bridge use	Data if not implemented	Treat as data until implemented
01-80-C2-00-00-0D	Reserved for future GVRP address	Data if not implemented	Treat as data until implemented
01-80-C2-00-00-0E	802.1ab-LLDP	BPDU	May peer
01-80-C2-00-00-0F	Reserved for future C-bridge or Q-bridge use	Data	Data
01-80-C2-00-00-10	All bridge addresses	Data	Data
01-80-C2-00-00-20	GMRP	Data	Data
01-80-C2-00-00-21	GVRP	Data	Data
01-80-C2-00-00-22 – 2F	Other GARP addresses	Data	Data
01-00-0C-CC-CC-CC	Cisco’s CDP DTP VTP PagP UDLD	Data	Data
01-00-0C-CC-CC-CD	Cisco’s PVST	Data	Data

NNI Ports

The Dot1add NNI ports behave in the same way as the customer facing S-bridge ports, with the following exceptions:

- On NNI ports, frames received with DA 01-80-C2-00-00-08 contain STP BPDU. The frames are received and transmitted. On S-UNI ports, any such frames that are received are dropped, and none are sent.

- On NNI ports, frames received with DA 01-80-C2-00-00-02 include CDP Pagp, VTP, DTP, and UDLD protocols.

7600 Action Table

Table 53-5 lists the actions performed on a packet when the packet is received with a specified destination MAC address.

Table 53-5 7600 Action Table

MAC Address	Protocol	C-UNI Action	S-UNI Action	NNI Action
01-80-C2-00-00-00	Bridge Group Address (BPDUs)	Peer	Data	Data
01-80-C2-00-00-01	802.3X Pause Protocol	Drop	Drop	Drop
01-80-C2-00-00-02	Slow Protocol address: 802.3ad LACP, 802.3ah	Peer	Peer	Peer
01-80-C2-00-00-03	802.1X	May peer	May peer	May peer
01-80-C2-00-00-04	Reserved	Drop	Drop	Drop
01-80-C2-00-00-05	Reserved	Drop	Drop	Drop
01-80-C2-00-00-06	Reserved	Drop	Drop	Drop
01-80-C2-00-00-07	Reserved	Drop	Drop	Drop
01-80-C2-00-00-08	Provider STP (BPDU)	Drop	Drop	Peer
01-80-C2-00-00-09	Reserved for future bridge use	Drop	Drop	Drop
01-80-C2-00-00-0A	Reserved for future bridge use	Drop	Drop	Drop
01-80-C2-00-00-0B	Reserved for future bridge use	Drop	Data	Data
01-80-C2-00-00-0C	Reserved for future bridge use	Drop	Data	Data
01-80-C2-00-00-0D	Reserved for future GVRP address	Drop	Data	Data
01-80-C2-00-00-0E	802.1ab-LLDP	May peer	Data	Data
01-80-C2-00-00-0F	Reserved for future C-bridge or Q-bridge use	Drop	Data	Data
01-80-C2-00-00-10	All bridge addresses	Snoop if implemented. Else drop	Data	Data
01-80-C2-00-00-20	GMRP	May peer	Data	Data
01-80-C2-00-00-21	GVRP	May peer	Data	Data
01-80-C2-00-00-22 – 2F	Other GARP addresses	May peer	Data	Data

Table 53-5 7600 Action Table

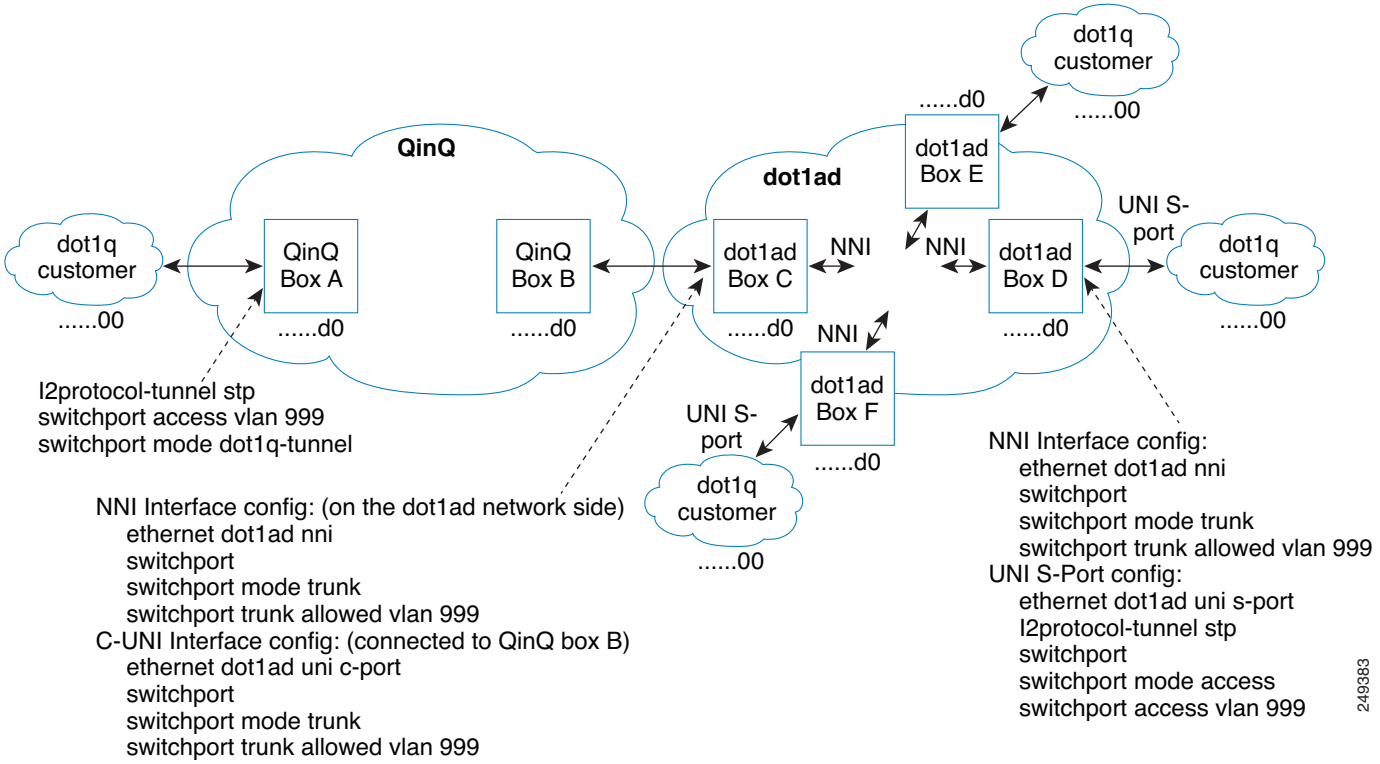
MAC Address	Protocol	C-UNI Action	S-UNI Action	NNI Action
01-00-0C-CC-CC-CC	Cisco's CDP DTP VTP PagP UDLD	Peer	Data	Data
01-00-0C-CC-CC-CD	Cisco's PVST	May peer	Data	Data

Interoperability of QinQ and Dot1ad

The interoperability of QinQ and Dot1ad network enables the exchange of data frames between the networks. The 802.1Q network outer tag VLANs are mapped to the provider S-VLANs of the 802.1ad network.

Figure 53-2 illustrates the interoperability of a Dot1ad network and a QinQ network.

Figure 53-2 Interoperability of Dot1ad Network and a QinQ Network



How to Configure IEEE 802.1ad

This section contains the information about following procedures:

- Configuring a Switchport
- Configuring a Layer 2 Protocol Forward
- Configuring a Switchport for Translating QinQ to 802.1ad

- Configuring a Switchport (L2PT)
- Configuring a Customer-Facing UNI-C Port with EVC
- Configuring a Customer-Facing UNI-C Port and Switchport on NNI with EVC
- Configuring a Customer-Facing UNI-S Port with EVC
- Configuring a Layer 3 Termination
- Displaying a Dot1ad Configuration

Configuring a Switchport

A switchport can be configured as a UNI-C port, UNI-S port, or NNI port.

UNI-C Port

A UNI-C port can be configured as either a trunk port or an access port. Perform the following tasks to configure a UNI-C port as an access port for 802.1ad.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ethernet dot1ad {nni | uni {c-port | s-port}}**
5. **switchport**
6. **switchport mode {access | trunk}**
7. **switchport access vlan** *vlan-id*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** Example: Router> enable	Enables privileged EXEC mode.
Step 2	**configure terminal** Example: Router# configure terminal	Enters global configuration mode.
Step 3	**interface** *type number* Example: Router# interface gigabitethernet 2/1	Configures an interface.
Step 4	**ethernet dot1ad {nni | uni {c-port | s-port}}** Example: Router(config-if)# ethernet dot1ad uni c-port	Configures a dot1ad NNI port or UNI port. In this example, it is a UNI-C port.

	Command or Action	Purpose
Step 5	**switchport** Example: Router(config-if)# switchport	Put the interface into Layer 2 mode.
Step 6	**switchport mode {access | trunk}** Example: Router(config-if)# switchport mode access	Sets the interface type. In this example, it is Access.
Step 7	**switchport access vlan** *vlan-id* Example: Router(config-if)# switchport access 1000	Sets the VLAN when an interface is in access mode. In this example, the VLAN is set to 1000.
Step 8	**end** Example: Router(config-if)# end	Returns the CLI to privileged EXEC mode.

Perform the following tasks to configure a UNI-C port as a trunk port for 802.1ad.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ethernet dot1ad {nni | uni {c-port | s-port}}**
5. **switchport**
6. **switchport mode {access | trunk}**
7. **switchport trunk allowed vlan** *vlan-list*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** Example: Router> enable	Enables privileged EXEC mode.
Step 2	**configure terminal** Example: Router# configure terminal	Enters global configuration mode.
Step 3	**interface** *type number* Example: Router# interface gigabitethernet 2/1	Configures an interface.
Step 4	**ethernet dot1ad {nni | uni {c-port | s-port}}** Example: Router(config-if)# ethernet dot1ad uni c-port	Configures a dot1ad NNI port or UNI port. In this example, it is a UNI-C port.

	Command or Action	Purpose
Step 5	**switchport** Example: Router(config-if)# switchport	Put the interface into Layer 2 mode.
Step 6	**switchport mode {access | trunk}** Example: Router(config-if)# switchport mode trunk	Sets the interface type. In this example, it is Trunk.
Step 7	**switchport trunk allowed vlan** *vlan-list* Example: Router(config-if)# switchport trunk allowed vlan 1000, 2000	Sets the list of allowed VLANs that transmit traffic from this interface in tagged format when in trunking mode.
Step 8	**end** Example: Router(config-if)# end	Returns the CLI to privileged EXEC mode.

UNI-S Port

On a UNI-S port, all the customer VLANs that enter are provided with the same service. The port allows only access configuration. In this mode, the customer's port is configured as a trunk port. Therefore, the traffic entering the UNI-S port is tagged traffic.

Perform the following tasks to configure a UNI-S port as an access port for 802.1ad.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **switchport**
5. **switchport mode {access | trunk}**
6. **ethernet dot1ad {nni | uni {c-port | s-port}}**
7. **switchport access vlan** *vlan-id*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** Example: Router> enable	Enables privileged EXEC mode.
Step 2	**configure terminal** Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* Example: Router# interface gigabitethernet 2/1	Configures an interface.
Step 4	**switchport** Example: Router(config-if)# switchport	Put the interface into Layer 2 mode.
Step 5	**switchport mode {access | trunk}** Example: Router(config-if)# switchport mode access	Sets the interface type. In this example, it is Access.
Step 6	**ethernet dot1ad {nni | uni {c-port | s-port}}** Example: Router(config-if)# ethernet dot1ad uni s-port	Configures a dot1ad NNI port or UNI port. In this example, it is a UNI-S port.
Step 7	**switchport access vlan** *vlan-id* Example: Router(config-if)# switchport access 999	Sets the VLAN when an interface is in access mode. In this example, the VLAN is set to 999.
Step 8	**end** Example: Router(config-if)# end	Returns the CLI to privileged EXEC mode.

NNI Port

NNI port allows only trunk configuration. On an NNI port, the frames received on all the allowed VLANs are bridged to the respective internal VLANs.

Perform the following tasks to configure an NNI port as a trunk port for 802.1ad.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **switchport**
5. **switchport mode {access | trunk}**
6. **ethernet dot1ad {nni | uni {c-port | s-port}}**
7. **switchport trunk allowed vlan** *vlan-list*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** Example: Router> enable	Enables privileged EXEC mode.
Step 2	**configure terminal** Example: Router# configure terminal	Enters global configuration mode.
Step 3	**interface** *type number* Example: Router# interface gigabitethernet 2/1	Configures an interface.
Step 4	**switchport** Example: Router(config-if)# switchport	Put the interface into Layer 2 mode.
Step 5	**switchport mode {access | trunk}** Example: Router(config-if)# switchport mode trunk	Sets the interface type. In this example, it is Trunk.
Step 6	**ethernet dot1ad {nni | uni {c-port | s-port}}** Example: Router(config-if)# ethernet dot1ad nni	Configures a dot1ad NNI port or UNI port. In this example, it is an NNI.
Step 7	**switchport trunk allowed vlan** *vlan-list* Example: Router(config-if)# switchport trunk allowed vlan 999	Sets the list of allowed VLANs that transmit traffic from this interface in tagged format when in trunking mode.
Step 8	**end** Example: Router(config-if)# end	Returns the CLI to privileged EXEC mode.

Examples

The following example shows how to configure a UNI-C port as an access port. In this example, all the frames that are received are bridged to one internal VLAN 1000. The transmitted frames do not have the access VLAN Dot1q tag.

```
Router# configure terminal
Router(config)# interface gig2/1
Router(config-if# ethernet dot1ad uni c-port
Router(config-if)# switchport
Router(config-if)# switchport mode access
Router(config-if)# switchport access vlan 1000
```

The following example shows how to configure a UNI-C port as a trunk port. In this example, all the frames that are received on all allowed VLANs (1000 and 2000) are bridged to the respective internal VLANs. The transmitted frames have the respective internal VLAN Dot1q tag.

```
Router# configure terminal
outer(config)# interface gig2/1
Router(config-if)# ethernet dot1ad uni c-port
```

```
Router(config-if)# switchport
Router(config-if)# switchport mode trunk
Router(config-if)# switchport access vlan 1000, 2000
```

The following example shows how to configure a UNI-S port. In this example, all the frames that are received are bridged to one internal VLAN (999). The transmitted frames do not have the access VLAN Dot1q tag.

```
Router# configure terminal
Router(config)# interface gig2/1
Router(config-if)# switchport
Router(config-if)# switchport mode access
Router(config-if)# ethernet dot1ad uni s-port
Router(config-if)# switchport access vlan 999
```

The following example shows how to configure an NNI port. Only trunk configuration is allowed on an NNI port. In this example, all the frames that are received on all the allowed VLANs (999) are bridged to the respective internal VLANs. The transmitted frames have the respective internal VLAN Dot1q tag.

```
Router# configure terminal
Router(config)# interface gig2/1
Router(config-if)# switchport
Router(config-if)# switchport mode trunk
Router(config-if)# ethernet dot1ad nni
Router(config-if)# switchport trunk allowed vlan 999
```

The following example shows how to configure Dot1ad on an SVI:

```
Router# configure terminal
Router(config)# interface gig2/1
Router(config-if)# ethernet dot1ad nni
Router(config-if)# switchport
Router(config-if)# switchport mode trunk
Router(config-if)# switchport trunk allowed vlan 999
Router(config)# interface vlan 999
Router(config-if)# ip address 1.2.3.4 255.255.0.0
```

Configuring a Layer 2 Protocol Forward

Perform the following tasks to configure the Layer 2 protocol forward:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **switchport access vlan** *vlan-id*
5. **ethernet dot1ad {nni | uni {c-port | s-port}}**
6. **l2protocol [forward] [***protocol***]**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** Example: Router> enable	Enables privileged EXEC mode.
Step 2	**configure terminal** Example: Router# configure terminal	Enters global configuration mode.
Step 3	**interface** *type number* Example: Router(config)# interface gigabitethernet 3/0	Configures an interface.
Step 4	**switchport access vlan** *vlan-id* Example: Router(config)# switchport access vlan 500	Sets the VLAN when an interface is in access mode.
Step 5	**ethernet dot1ad {nni | uni {c-port | s-port}}** Example: Router(config-if)# ethernet dot1ad uni s-port	Configures a dot1ad NNI port or UNI port. In this example, it is a UNI S-port.
Step 6	**l2 protocol [forward] [***protocol***]** Example: Router(config-if)# l2 protocol forward vtp	Processes or forwards the Layer 2 BPDUs. In this example, all the BPDUs are forwarded except VTP PDUs.
Step 7	**end** Example: Router(config-if)# end	Returns the CLI to privileged EXEC mode.

Examples

The following example shows how to configure a Layer 2 protocol forward:

```
Router# configure terminal
Router(config)# interface gig3/0
Router(config-if)# switchport access vlan 500
Router(config-if)# ethernet dot1ad uni s-port
Router(config-if)# l2protocol forward vtp
```

Configuring a Switchport for Translating QinQ to 802.1ad

Translating a QinQ port to 802.1ad involves configuring the port connecting to QinQ port and NNI port.

Perform the following tasks to configure a port connecting to the QinQ port.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*

4. **switchport mode {access | trunk}**
5. **switchport trunk allowed vlan** *vlan-list*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** Example: Router> enable	Enables privileged EXEC mode.
Step 2	**configure terminal** Example: Router# configure terminal	Enters global configuration mode.
Step 3	**interface** *type number* Example: Router# interface gigabitethernet 1/1	Configures an interface.
Step 4	**switchport mode {access | trunk}** Example: Router(config-if)# switchport mode trunk	Sets the interface type. In this example, it is Trunk.
Step 5	**switchport trunk allowed vlan** *vlan-list* Example: Router(config-if)# switchport trunk allowed vlan 1000	Sets the list of allowed VLANs that transmit traffic from this interface in tagged format when in trunking mode.
Step 6	**end** Example: Router(config-if)# end	Returns the CLI to privileged EXEC mode.

Perform the following tasks to configure an NNI port.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ethernet dot1ad {nni | uni {c-port | s-port}}**
5. **switchport**
6. **switchport mode {access | trunk}**
7. **switchport trunk allowed vlan** *vlan-list*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** Example: Router> enable	Enables privileged EXEC mode.
Step 2	**configure terminal** Example: Router# configure terminal	Enters global configuration mode.
Step 3	**interface** *type number* Example: Router# interface gigabitethernet 4/1	Configures an interface.
Step 4	**ethernet dot1ad {nni | uni {c-port | s-port}}** Example: Router(config-if)# ethernet dot1ad nni	Configures a dot1ad NNI port or UNI port. In this example, it is an NNI.
Step 5	**switchport** Example: Router(config-if)# switchport	Put the interface into Layer 2 mode.
Step 6	**switchport mode {access | trunk}** Example: Router(config-if)# switchport mode trunk	Sets the interface type. In this example, it is Trunk.
Step 7	**switchport trunk allowed vlan** *vlan-list* Example: Router(config-if)# switchport trunk allowed vlan 999-1199	Sets the list of allowed VLANs that transmit traffic from this interface in tagged format when in trunking mode.
Step 8	**end** Example: Router(config-if)# end	Returns the CLI to privileged EXEC mode.

Examples

The following example shows how to translate a QinQ port to 802.1ad. In this example, the peer router to gig1/1 multiplexes various customer VLANs into VLAN 1000.

```
Router# configure terminal
Router(config)# interface gig1/1
Router(config-if)# switchport mode trunk
Router(config-if)# switchport trunk allowed vlan 1000


Router# configure terminal
Router(config)# interface gig4/0
Router(config-if)# ethernet dot1ad nni
Router(config-if)# switchport
Router(config-if)# switchport mode trunk
Router(config-if)# switchport trunk allowed vlan 1000,1199
```

Configuring a Switchport (L2PT)

Configuring the switchport for L2PT is required to tunnel the STP packets from a customer on the dot1ad network to a customer on the QinQ network.

Perform the following tasks to configure the port connecting to the customer.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **switchport**
5. **ethernet dot1ad {nni | uni {c-port | s-port}}**
6. **no l2 protocol [peer | forward]** [*protocol*]
7. **l2protocol-tunnel [cdp | stp | vtp]**
8. **switchport mode {access | trunk}**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** Example: Router> enable	Enables privileged EXEC mode.
Step 2	**configure terminal** Example: Router# configure terminal	Enters global configuration mode.
Step 3	**interface** *type number* Example: Router(config)# interface gigabitethernet 2/1	Configures an interface.
Step 4	**switchport** Example: Router(config-if)# switchport	Put the interface into Layer 2 mode.
Step 5	**ethernet dot1ad {nni | uni {c-port | s-port}}** Example: Router(config-if)# ethernet dot1ad uni s-port	Configures a dot1ad NNI port or UNI port. In this example, it is a UNI S-port.
Step 6	**no l2 protocol [peer | forward]** [*protocol*] Example: Router(config-if)# no l2 protocol forward	Disables L2 protocol forwarding.
Step 7	**l2protocol-tunnel [cdp | stp | vtp]** Example: Router(config-if)# l2protocol-tunnel stp	Enables protocol tunneling for STP.

	Command or Action	Purpose
Step 8	**switchport mode {access | trunk}** Example: Router(config-if)# switchport mode trunk	Sets the interface type. In this example, it is Trunk.
Step 9	**end** Example: Router(config-if)# end	Returns the CLI to privileged EXEC mode.

Perform the following tasks to configure an NNI port.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **switchport**
5. **ethernet dot1ad {nni | uni {c-port | s-port}}**
6. **switchport mode {access | trunk}**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** Example: Router> enable	Enables privileged EXEC mode.
Step 2	**configure terminal** Example: Router# configure terminal	Enters global configuration mode.
Step 3	**interface** *type number* Example: Router(config)# interface gigabitethernet 2/1	Configures an interface.
Step 4	**switchport** Example: Router(config-if)# switchport	Put the interface into Layer 2 mode.
Step 5	**ethernet dot1ad {nni | uni {c-port | s-port}}** Example: Router(config-if)# ethernet dot1ad nni	Configures a dot1ad NNI or UNI port. In this example, it is an NNI.

	Command or Action	Purpose
Step 6	**switchport mode {access | trunk}** Example: Router(config-if)# switchport mode trunk	Sets the interface type. In this example, it is Trunk.
Step 7	**end** Example: Router(config-if)# end	Returns the CLI to privileged EXEC mode.

Examples

The following example shows how to tunnel the STP packets from a customer on the Dot1ad network to a customer on a QinQ network:

```
Router# configure terminal
Router(config)# interface gig1/0
Router(config-if)# switchport
Router(config-if)# ethernet dot1ad uni s-port
Router(config-if)# no l2protocol forward
Router(config-if)# l2protocol-tunnel stp
Router(config-if)# switchport mode access


Router# configure terminal
Router(config)# interface gig4/0
Router(config-if)# switchport
Router(config-if)# ethernet dot1ad nni
Router(config-if)# switchport mode trunk
```

Configuring a Customer-Facing UNI-C Port with EVC

Perform the following tasks to configure a UNI-C port.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ethernet dot1ad {nni | uni {c-port | s-port}}**
5. **service instance** *id service-type*
6. **encapsulation dot1q** *vlan*-**id second-dot1q {any | *vlan*-id} [native]**
7. **bridge-domain** *vlan-id*
8. **service instance** *id service-type*
9. **encapsulation dot1q** *vlan*-**id second-dot1q {any | *vlan*-id} [native]**
10. **bridge-domain** *vlan-id*
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** Example: Router> enable	Enables privileged EXEC mode.
Step 2	**configure terminal** Example: Router# configure terminal	Enters global configuration mode.
Step 3	**interface** *type number* Example: Router(config)# interface gigabitethernet 2/1	Configures an interface.
Step 4	**ethernet dot1ad {nni | uni {c-port | s-port}}** Example: Router(config-if)# ethernet dot1ad uni c-port	Configures a dot1ad NNI port or UNI port. In this example, it is a UNI C port.
Step 5	**service instance** *id service-type* Example: Router(config-if)# service instance 1 ethernet	Configures an Ethernet service instance. In this example, the service instance is 1.
Step 6	**encapsulation dot1q** *vlan-id* **second-dot1q {any |** *vlan-id***} [native]** Example: Router(config-if)# encapsulation dot1q 1-100	Enables IEEE 802.1Q encapsulation of traffic on a specified subinterface in a VLAN.
Step 7	**bridge-domain** *vlan-id* Example: Router(config-if)# bridge-domain 1000	Binds a service instance or a MAC tunnel to a bridge domain.
Step 8	**service instance** *id service-type* Example: Router(config-if)# service instance 2 ethernet	Configures an Ethernet service instance. In this example, the service instance is 2.
Step 9	**encapsulation dot1q** *vlan-id* **second-dot1q {any |** *vlan-id***} [native]** Example: Router(config-if)# encapsulation dot1q 102-4094	Enables IEEE 802.1Q encapsulation of traffic on a specified subinterface in a VLAN.
Step 10	**bridge-domain** *vlan-id* Example: Router(config-if)# bridge-domain 500	Binds a service instance or a MAC tunnel to a bridge domain.
Step 11	**end** Example: Router(config-if)# end	Returns the CLI to privileged EXEC mode.

Perform the following tasks to configure an NNI port.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **interface** *type number*
4. **ethernet dot1ad {nni | uni {c-port | s-port}}**
5. **service instance** *id service-type*
6. **encapsulation dot1q** *vlan*-**id second-dot1q {any |** *vlan*-**id} [native]**
7. **rewrite ingress tag pop 1 symmetric**
8. **bridge-domain** *vlan-id*
9. **service instance** *id service-type*
10. **encapsulation dot1q** *vlan*-**id second-dot1q {any |** *vlan*-**id} [native]**
11. **rewrite ingress tag pop 1 symmetric**
12. **bridge-domain** *vlan-id*
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** Example: Router> enable	Enables privileged EXEC mode.
Step 2	**configure terminal** Example: Router# configure terminal	Enters global configuration mode.
Step 3	**interface** *type number* Example: Router(config)# interface gigabitethernet 2/1	Configures an interface.
Step 4	**ethernet dot1ad {nni | uni {c-port | s-port}}** Example: Router(config-if)# ethernet dot1ad uni c-port	Configures a dot1ad NNI port or UNI port. In this example, it is a UNI C port.
Step 5	**service instance** *id service-type* Example: Router(config-if)# service instance 1 ethernet	Configures an Ethernet service instance. In this example, the service instance is 1.
Step 6	**encapsulation dot1q** *vlan-id* **second-dot1q {any |** *vlan-id***} [native]** Example: Router(config-if)# encapsulation dot1q 1000 second-dot1q 1-100	Enables IEEE 802.1Q encapsulation of traffic on a specified subinterface in a VLAN.
Step 7	**rewrite ingress tag pop 1 symmetric** Example: Router(config-if)# rewrite ingress tag pop 1 symmetric	Specifies the encapsulation adjustment that is to be performed on the frame ingress to the service instance.

	Command or Action	Purpose
Step 8	**bridge-domain** *vlan-id* Example: Router(config-if)# bridge-domain 1000	Binds a service instance or a MAC tunnel to a bridge domain.
Step 9	**service instance** *id service-type* Example: Router(config-if)# service instance 2 ethernet	Configures an Ethernet service instance. In this example, the service instance is 2.
Step 10	**encapsulation dot1q** *vlan-id* **second-dot1q {any |** *vlan-id***} [native]** Example: Router(config-if)# encapsulation dot1q 500 second-dot1q 102-4904	Enables IEEE 802.1Q encapsulation of traffic on a specified subinterface in a VLAN.
Step 11	**rewrite ingress tag pop 1 symmetric** Example: Router(config-if)# rewrite ingress tag pop 1 symmetric	Specifies the encapsulation adjustment that is to be performed on the frame ingress to the service instance.
Step 12	**bridge-domain** *vlan-id* Example: Router(config-if)# bridge-domain 500	Binds a service instance or a MAC tunnel to a bridge domain.
Step 13	**end** Example: Router(config-if)# end	Returns the CLI to privileged EXEC mode.

Examples

The following example shows how to configure a customer-facing UNI port. In this example, a dot1q frame coming on VLAN 50 matches service instance 1, and on the ingress port, the rewrite command pushes the 1000 outer-vlan.

```
Router# configure terminal
Router(config)# interface gig1/1
Router(config-if)# ethernet dot1ad uni c-port
Router(config-if)# service instance 1 ethernet
Router(config-if)# encapsulation dot1q 1-100
Router(config-if)# bridge-domain 1000
Router(config-if)# service instance 2 ethernet
Router(config-if)# encapsulation dot1q 102-4904
Router(config-if)# bridge-domain 500


Router# configure terminal
Router(config)# interface gig4/1
Router(config-if)# ethernet dot1ad nni
Router(config-if)# service instance 1 ethernet
Router(config-if)# encapsulation dot1q 1000 second dot1q 1-100
Router(config-if)# rewrite ingress tag pop 1 symmetric
Router(config-if)# bridge-domain 1000
Router(config-if)# service instance 2ethernet
Router(config-if)# encapsulation dot1q 500 second dot1q 102-4904
Router(config-if)# rewrite ingress tag pop 1 symmetric
Router(config-if)# bridge-domain 500
```

Configuring a Customer-Facing UNI-C Port and Switchport on NNI with EVC

Perform the following tasks to configure a UNI-C port.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ethernet dot1ad {nni | uni {c-port | s-port}}**
5. **service instance** *id service-type*
6. **encapsulation dot1q** *vlan*-**id second-dot1q {any | *vlan*-id} [native]**
7. **bridge-domain** *vlan-id*
8. **service instance** *id service-type*
9. **encapsulation dot1q** *vlan*-**id second-dot1q {any | *vlan*-id} [native]**
10. **bridge-domain** *vlan-id*
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** Example: Router> enable	Enables privileged EXEC mode.
Step 2	**configure terminal** Example: Router# configure terminal	Enters global configuration mode.
Step 3	**interface** *type number* Example: Router(config)# interface gigabitethernet 2/1	Configures an interface.
Step 4	**ethernet dot1ad {nni | uni {c-port | s-port}}** Example: Router(config-if)# ethernet dot1ad uni c-port	Configures a dot1ad NNI port or UNI port. In this example, it is a UNI C port.
Step 5	**service instance** *id service-type* Example: Router(config-if)# service instance 1 ethernet	Configures an Ethernet service instance. In this example, the service instance is 1.
Step 6	**encapsulation dot1q** *vlan-id* **second-dot1q {any |** *vlan-id***} [native]** Example: Router(config-if)# encapsulation dot1q 1-100	Enables IEEE 802.1Q encapsulation of traffic on a specified subinterface in a VLAN.

	Command or Action	Purpose
Step 7	**bridge-domain** *vlan-id* Example: Router(config-if)# bridge-domain 1000	Binds a service instance or a MAC tunnel to a bridge domain.
Step 8	**service instance** *id service-type* Example: Router(config-if)# service instance 2 ethernet	Configures an Ethernet service instance. In this example, the service instance is 2.
Step 9	**encapsulation dot1q** *vlan-id* **second-dot1q {any |** *vlan-id***} [native]** Example: Router(config-if)# encapsulation dot1q 102-4094	Enables IEEE 802.1Q encapsulation of traffic on a specified subinterface in a VLAN.
Step 10	**bridge-domain** *vlan-id* Example: Router(config-if)# bridge-domain 500	Binds a service instance or a MAC tunnel to a bridge domain.
Step 11	**end** Example: Router(config-if)# end	Returns the CLI to privileged EXEC mode.

Perform the following tasks to configure an NNI port.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ethernet dot1ad {nni | uni {c-port | s-port}}**
5. **switchport**
6. **switchport mode {access | trunk}**
7. **switchport trunk allowed vlan** *vlan-list*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** Example: Router> enable	Enables privileged EXEC mode.
Step 2	**configure terminal** Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* Example: Router# interface gigabitethernet 4/1	Configures an interface.
Step 4	**ethernet dot1ad {nni | uni {c-port | s-port}}** Example: Router(config-if)# ethernet dot1ad nni	Configures a dot1ad NNI port or UNI port. In this example, it is an NNI.
Step 5	**switchport** Example: Router(config-if)# switchport	Put the interface into Layer 2 mode.
Step 6	**switchport mode {access | trunk}** Example: Router(config-if)# switchport mode trunk	Sets the interface type. In this example, it is Trunk.
Step 7	**switchport trunk allowed vlan** *vlan-list* Example: Router(config-if)# switchport trunk allowed vlan 1000-500	Sets the list of allowed VLANs that transmit traffic from this interface in tagged format when in trunking mode.
Step 8	**end** Example: Router(config-if)# end	Returns the CLI to privileged EXEC mode.

Examples

The following example shows how to configure a customer-facing UNI-C port and switchport on NNI with EVC:

```
Router# configure terminal
Router(config)# interface gig1/1
Router(config-if)# ethernet dot1ad uni c-port
Router(config-if)# service instance 1 ethernet
Router(config-if)# encapsulation dot1q 1-100
Router(config-if)# bridge-domain 1000
Router(config-if)# service instance 2 ethernet
Router(config-if)# encapsulation dot1q 102-4904
Router(config-if)# bridge-domain 500


Router# configure terminal
Router(config)# interface gig4/0
Router(config-if)# switchport
Router(config-if)# ethernet dot1ad uni
Router(config-if)# switchport mode trunk
Router(config-if)# switchport allowed vlan 1000,500
```

Configuring a Customer-Facing UNI-S Port with EVC

Perform the following tasks to configure a UNI-S port.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **service instance** *id service-type*
5. **ethernet dot1ad {nni | uni {c-port | s-port}}**
6. **encapsulation default**
7. **bridge-domain** *vlan-id*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** Example: Router> enable	Enables privileged EXEC mode.
Step 2	**configure terminal** Example: Router# configure terminal	Enters global configuration mode.
Step 3	**interface** *type number* Example: Router(config)# interface gigabitethernet 2/1	Configures an interface.
Step 4	**service instance** *id service-type* Example: Router(config-if)# service instance 1 ethernet	Configures an Ethernet service instance. In this example, the service instance is 1.
Step 5	**ethernet dot1ad {nni | uni {c-port | s-port}}** Example: Router(config-if)# ethernet dot1ad uni s-port	Configures a dot1ad NNI port or UNI port. In this example, it is a UNI-S port.
Step 6	**encapsulation default** Example: Router(config-if)# encapsulation default	Configures the default service instance on a port. Anything that does not meet the criteria of other service instances on the same physical interface falls into this service instance.
Step 7	**bridge-domain** *vlan-id* Example: Router(config-if)# bridge-domain 1000	Binds a service instance or a MAC tunnel to a bridge domain.
Step 8	**end** Example: Router(config-if)# end	Returns the CLI to privileged EXEC mode.

Perform the following tasks to configure an NNI port.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **service instance** *id service-type*
5. **ethernet dot1ad {nni | uni {c-port | s-port}}**
6. **encapsulation dot1q** *vlan*-**id second-dot1q {any |** *vlan*-**id} [native]**
7. **rewrite ingress tag pop 1 symmetric**
8. **bridge-domain** *vlan-id*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** Example: Router> enable	Enables privileged EXEC mode.
Step 2	**configure terminal** Example: Router# configure terminal	Enters global configuration mode.
Step 3	**interface** *type number* Example: Router(config)# interface gigabitethernet 2/1	Configures an interface.
Step 4	**service instance** *id service-type* Example: Router(config-if)# service instance 1 ethernet	Configures an Ethernet service instance. In this example, the service instance is 1.
Step 5	**ethernet dot1ad {nni | uni {c-port | s-port}}** Example: Router(config-if)# ethernet dot1ad uni c-port	Configures a dot1ad NNI or UNI port. In this example, it is a UNI C port.
Step 6	**encapsulation dot1q** *vlan-id* **second-dot1q {any |** *vlan-id***} [native]** Example: Router(config-if)# encapsulation dot1q 1000 second-dot1q 1-100	Enables IEEE 802.1Q encapsulation of traffic on a specified subinterface in a VLAN.
Step 7	**rewrite ingress tag pop 1 symmetric** Example: Router(config-if)# rewrite ingress tag pop 1 symmetric	Specifies the encapsulation adjustment that is to be performed on the frame ingress to the service instance.

	Command or Action	Purpose
Step 8	**bridge-domain** *vlan-id* Example: Router(config-if)# bridge-domain 1000	Binds a service instance or a MAC tunnel to a bridge domain.
Step 9	**end** Example: Router(config-if)# end	Returns the CLI to privileged EXEC mode.

Examples

The following example shows how to configure an NNI port:

```
Router# configure terminal
Router(config)# interface gig1/1
Router(config-if)# service instance 1 ethernet
Router(config-if)# ethernet dot1ad nni
Router(config-if)# encapsulation dot1q 1000
Router(config-if)# rewrite ingress tag pop 1 symmetric
Router(config-if)# bridge-domain 1000
```

Configuring a Layer 3 Termination

Perform the following tasks to configure a Layer 3 termination.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ethernet dot1ad {nni | uni {c-port | s-port}}**
5. **interface** *type number*
6. **encapsulation dot1q** *vlan*-**id second-dot1q {any |** *vlan*-**id} [native]**
7. **ip address** *ip-address mask*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** Example: Router> enable	Enables privileged EXEC mode.
Step 2	**configure terminal** Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* Example: Router(config)# interface gigabitethernet 3/0	Configures an interface.
Step 4	**ethernet dot1ad {nni | uni {c-port | s-port}}** Example: Router(config-if)# ethernet dot1ad nni	Configures a dot1ad NNI or UNI port. In this example, it is an NNI port.
Step 5	**interface** *type number* Example: Router(config)# interface gigabitethernet 3/0/.1	Configures an interface.
Step 6	**encapsulation dot1q** *vlan-id* **second-dot1q {any |** *vlan-id***} [native]** Example: Router(config-if)# encapsulation dot1q 10 second-dot1q 10	Enables IEEE 802.1Q encapsulation of traffic on a specified subinterface in a VLAN.
Step 7	**ip address** Example: Router(config-if)# ip address 1.2.3.4 255.255.0.0	Sets a primary or secondary IP address for an interface.
Step 8	**end** Example: Router(config-if)# end	Returns the CLI to privileged EXEC mode.

Examples

The following example shows how to configure a Layer 3 termination. Note that Layer 3 is supported only on trunk interfaces.

```
Router# configure terminal
Router(config)# interface gig3/0
Router(config-if)# ethernet dot1ad nni
Router(config)# interface gig3/0/0.1
Router(config-if)# encapsulation dot1q 10 second dot1q 10
Router(config-if)# ip address 1.2.3.4 255.255.0.0
```

The following example shows how to configure a Layer 3 termination on an SVI:

```
Router# configure terminal
Router(config)# interface gig4/1
Router(config-if)# ethernet dot1ad nni
Router(config-if)# service instance 1 ethernet
Router(config-if)# encapsulation dot1q 200 second dot1q 300
Router(config-if)# rewrite ingress tag pop 2 symmetric
Router(config-if)# bridge-domain 50
Router(config-if)# service instance 2 ethernet
Router(config-if)# encapsulation dot1q 300
Router(config-if)# rewrite ingress tag pop 1 symmetric
Router(config-if)# bridge-domain 60

Router(config)# interface vlan 50
Router(config-if)# ip address 2.3.4.5 255.255.0.0
Router(config)# interface vlan 60
Router(config-if)# ip address 3.4.5.6 255.255.0.0
```

Displaying a Dot1ad Configuration

You can display a Dot1ad configuration using the **show ethernet dot1ad** command. This command displays the Dot1ad configuration for all interfaces. To display the configuration on a particular interface, use the **show ethernet dot1ad interface** command.

The following example shows how to display a Dot1ad configuration on all interfaces:

```
Router# show ethernet dot1ad
Interface: GigabitEthernet4/0/1
DOT1AD C-Bridge Port
L2protocol pass cdp stp vtp dtp pagp dot1x lacp

Interface: GigabitEthernet4/0/2
DOT1AD C-Bridge Port
L2protocol pass cdp stp vtp dtp pagp dot1x lacp
```

Troubleshooting Dot1ad

The following section describes how to troubleshoot Dot1ad.

Note The show commands in these examples should be run from a line card console.

- How do I verify the Dot1ad configuration on a switchport on an X40G card?

  Run the following command to verify the Dot1ad configuration:

```
XYZ-PE1-dfc1# show platform npc switchport interface gi 1/2
 [GigabitEthernet1/2]
        status [valid, -, applied, enabled]
        src_index [0x1]
        rpcb [0x178BB9C4]
        xlif_id [4097]
        xlif_handle [type:[3] hwidb:[0x20E97F08] if_number:[1121]]
        ft_bits [0x2]
        ing_ctrl_ft_bits [0x2]
        egr_ctrl_ft_bits [0x2]
        port vlan [1]
        mode ingress [NORMAL] egress [NORMAL]
        dot1q_tunnel [No]
        native tagging [No]
        PVLAN isolated or community [No] promiscuous [No]
        ingress vlan-translation [No] BPDU [No]
        egress  vlan-translation [No] BPDU [No]
        dot1ad [Yes] <<<<<<<<<<<<
        ethertype [0x88A8] <<<<<<<<<<<
        Ingress Stat ID: 778698
        Egress Stat ID: 778700
        VLAN List:
        1
        num of vlans [1]
       XYZ-PE1-dfc1#
```

- How do I verify the Dot1ad configuration on the ports with EVCs on an X40G card?

  Run the following command to verify the Dot1ad configuration:

```
XYZ-PE1-dfc1# show platform npc xlif interface gi 1/2 efp 1
EFP XLIF(GigabitEthernet1/2, efp1)[np0] = 4136
```

```
 Ingress XLIF table fields

 Feature common enable:  0x1
 Feature enable:         0x1
 Feature bits:           0x1
 Control common bits:    0x0
 Control feature bits:   0x0
 Control rewrite opcode: 0x0
 Reserved 1:             0x0
 Match cond              0x1
 Entry valid:            0x1
 Dbus VLAN:              30
 QoS policy ID:          0
 ACL ID:                 0
 Statistics ID:          450976
 Inner rewrite VLAN:     0
 Outer rewrite VLAN:     0
 QoS flow ID:            0
 Feature data: 00000000 40000000 AAA80000 E0000829
 EFP admin down state  0x0
----- Bridge data ------
 layer2_acl_index:           0x00000000
 evc_feat_data.ip_src_guard     : 0x0
 evc_feat_data.mst_evc          : 0x1
 evc_feat_data.layer2_acl       : 0x0
 EVC - Mac Security:     0x0
 evc_feat_data.sacl       : 0x0
 evc_feat_data.layer2_acl_statid: 0
 PDT: 0xAAA8
 ipsg_label: 0
 block_data: 0x0
 block_l2bpdu: 0x0
 split_h: 0x0
 imp_ltl: 0x0829
 EFP dot1ad port type 0x3       <<<<<<<<
 EFP CDP forward 0x1  <<<<<<<<
 EFP DTP forward 0x0
 EFP VTP forward 0x0
 EFP STP forward 0x0
 EFP DOT1X forward 0x0

 Egress XLIF table fields

 Feature common enable:  0x1
 Feature enable:         0x1
 Feature bits:           0x01
 Control common bits:    0x00
 Control feature bits:   0x00
 Control rewrite opcode: 0x00
 Port:                   0x1
 Match cond              0x1
 Entry valid:            0x1
 Dbus VLAN:              30
 QoS policy ID:          0
 ACL ID:                 0
 Statistics ID:          450980
 Inner rewrite VLAN:     0
 Outer rewrite VLAN:     0
 QoS flow ID:            0
 IP Session en :         0
 Multicast  en :         0
 Feature data 0          0x00000000
 Intf etype:             0x00008064
```

```
 Post Filter Opcode      0x00000008
 Pre Filter Opcode       0x00000000
 Pre Tag Outer           0x00000000
 Pre Tag Inner           0x00000000
 Post Filter Vlan high   0x00000064
 Post Filter Vlan low    0x00000064
 Post Filter Vlan outer  0x00000000
 EVC - MST:              0x1
 EVC etype               0x8100
 CFM MEP Level           0x00000008
 CFM MIP Level           0x00000008
 CFM disable             0x0
 MIP filtering           0x0
 block_data:            0x0
 block_l2bpdu:           0x0
 sacl:                   0x0
 sacl index:             0x0000
 sacl statid:            0x00000
XYZ-PE1-dfc1#
XYZ-PE1-dfc1#
```

- How do I verify the L2protocol forwarding on a regular L3 switchports?

  Run the following command to verify the L2protocol forwarding:

```
XYZ-PE1-dfc1# show platform npc xlif 0 port_sram 1

.......................

 dot1ad port type:      0x0002  <<<<<<<<<
 l2proto cdp fwd:       0x0001  <<<<<<<<<
 l2proto dtp fwd:       0x0000
 l2proto vtp fwd:       0x0000
 l2proto stp fwd:       0x0000
 l2proto dot1x fwd:     0x0000

          ........................................
```

- How do I verify the Dot1ad configuration on ES20 cards?

  For switchports, run the following command:

```
XYZ-PE1-dfc1# show platform hardware dot1ad l2protocfg port <port-num>
```

  For EVCs, run the following command:

```
XYZ-PE1-dfc1# show platform soft efp-client interface gi x/0/y efp-id l2protocfg
```

  To display the default values, run the following commands:

```
XYZ-PE1-dfc1# show platform hardware dot1ad l2protocfg defaults ?
  <0-2>  0=c-uni, 1=s-uni, 2=nni

XYZ-PE1-dfc1# show platform hardware dot1ad l2protocfg defaults 0 ?
  <0-2>  0=L3, 1=BD, 2=XCON

XYZ-PE1-dfc1# show platform hardware dot1ad l2protocfg defaults 0 2
Raw Data :000FFF77 FFFCFF51
 L2 Proto Configs :
  Protocol         IEEE       CISCO
 -------------------------------------
    CDP       :    FRWD       FRWD
    VTP       :    FRWD       FRWD
    DTP       :    FRWD       FRWD
```

```
  Others     :    PEER       PEER

802.1d protocols : 01:80:C2:00:00:XX

XX | Config    XX | Config    XX | Config    XX | Config
-----------    -----------    -----------    -----------
00 : PEER      01 : DROP      02 : PEER      03 : PEER
04 : FRWD      05 : FRWD      06 : FRWD      07 : FRWD
08 : DROP      09 : FRWD      0A : FRWD      0B : FRWD
0C : FRWD      0D : FRWD      0E : FRWD      0F : FRWD

All Bridge (0180C2000010)= FRWD
Group = PEER
PVST = FRWD
```