QoS Policies for VFI Pseudowires

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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.

Restrictions for QoS Policies for VFI Pseudowires

- A maximum of 32K pseudowires.
- A maximum of 4K unique policy maps.
- A maximum of 128 neighbors per VFI context.

Information About QoS Policies for VFI Pseudowires

QoS Policies for VFI Pseudowires

QoS policies are specified on individual pseudowire interfaces and are applied only to the corresponding pseudowires. It is possible to specify different QoS policies on different pseudowire members of the same virtual forwarding interface (VFI) or on the subset of the pseudowires. There may be one or more pseudowires configured per VFI. Both manually configured and auto discovered pseudowire configurations are supported.

QoS policies are specified using a pseudowire template. The template can be applied on multiple pseudowires of the same, or different, VFIs. All those pseudowires get the same QoS policy applied as specified in the template. For auto-discovered pseudowires, QoS policies can only be specified using a pseudowire template.

The QoS Policies for VFI Pseudowires feature supports both ingress and egress policies and traffic classification can be done based on different match criteria.

How to Configure QoS Policies for VFI Pseudowires

Configuring QoS Policies for Pseudowires

Perform this task to configure QoS policies for pseudowires.

Before You Begin

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** *class-map-name*
5. **priority** *bandwidth-kbps*
6. **exit**
7. **class** *class-map-name*
8. **bandwidth percent** *percentage*
9. **exit**
10. **class** *class-map-name*
11. **police cir** *bps*
12. **exit**
13. **class** *class-map-name*
14. **shape average** *bps*
15. **queue-limit** *queue-limit size* **packets**
16. **random-detect**
17. **exit**
18. **exit**
19. **policy-map** *policy-map-name*
20. **class** *class-map-name*
21. **shape average** *bps*
22. **service-policy** *policy-map*
23. **exit**
24. **exit**
25. **policy-map** *policy-map-name*
26. **class** *class-map-name*
27. **shape average** *bps*
28. **exit**
29. **exit**
30. **policy-map** *policy-map-name*
31. **class** *class-map-name*
32. **shape average** *bps*
33. **exit**
34. **exit**
35. **exit policy-map** *policy-map-name*
36. **class** *class-map-name*
37. **shape average** *bps*
38. **exit**
39. **exit**

40. **policy-map** *policy-map-name*
41. **class** *class-map-name*
42. **police** *bps*
43. **interface pseudowire** *number*
44. **encap mpls**
45. **neighbor** *peer-address vcid-value*
46. **service-policy input** *policy-map-name*
47. **service-policy output** *policy-map-name*
48. **interface gigabit ethernet** *number*
49. **service-policy output** *policy-map-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. **Note** Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map-name* **Example:** `Device# policy-map gold-policy-child`	Creates a policy map to specify a service policy.
Step 4	**class** *class-map-name* **Example:** `Device(config-pmap)# class priority-class`	Specifies the name of the class map.
Step 5	**priority** *bandwidth-kbps* **Example:** `Device(config-pmap-c)# priority 100`	Gives priority to a class of traffic belonging to a policy map.
Step 6	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode.

	Command or Action	**Purpose**
Step 7	**class** *class-map-name* **Example:** Device(config-pmap-c)# class guarantee-class	Specifies the name of the class map.
Step 8	**bandwidth percent** *percentage* **Example:** Device(config-pmap-c)# bandwidth percent 50	Specifies or modifies the bandwidth allocated for a class belonging to a policy map.
Step 9	**exit** **Example:** Device(config-pmap-c)# exit	Exits policy-map class configuration mode.
Step 10	**class** *class-map-name* **Example:** Device(config-pmap-c)# class limited-class	Specifies the name of the class map.
Step 11	**police cir** *bps* **Example:** Device(config-pmap-c)# police cir 8000	Creates a per-interface policer and configures the policy-map class to use it.
Step 12	**exit** **Example:** Device(config-pmap-c)# exit	Exits policy-map class configuration mode.
Step 13	**class** *class-map-name* **Example:** Device(config-pmap)# class class-default	Specifies the name of the class map.
Step 14	**shape average** *bps* **Example:** Device(config-pmap-c)# shape average 8000	Shapes traffic to the indicated bit rate.

	Command or Action	**Purpose**
Step 15	**queue-limit** *queue-limit size* **packets** **Example:** Device(config-pmap-c)# queue-limit 150 packets	Specifies the queue limit size for a class.
Step 16	**random-detect** **Example:** Device(config-pmap-c)# andom-detect	Configures Weighted Random Early Detection (WRED) for a class in a policy map.
Step 17	**exit** **Example:** Device(config-pmap-c)# exit	Exits policy-map class configuration mode.
Step 18	**exit** **Example:** Device(config-pmap)# exit	Exits policy-map configuration mode.
Step 19	**policy-map** *policy-map-name* **Example:** Device(config)# policy-map gold-policy-hqos	Creates a policy map to specify a service policy.
Step 20	**class** *class-map-name* **Example:** Device(config-pmap)# class class-default	Specifies the name of the class map.
Step 21	**shape** **average** *bps* **Example:** Device(config-pmap-c)# shape average 10000	Shapes traffic to the indicated bit rate.
Step 22	**service-policy** *policy-map* **Example:** Device(config-pmap-c)# service-policy gold-policy-child	Attaches a policy map to a class.

	Command or Action	**Purpose**
Step 23	**exit**	Exits policy-map class configuration mode.
	Example:	
	`Device(config-pmap-c)# exit`	
Step 24	**exit**	Exits policy-map configuration mode.
	Example:	
	`Device(config-pmap)# exit`	
Step 25	**policy-map** *policy-map-name*	Creates a policy map to specify a service policy.
	Example:	
	`Device(config)# policy-map pw-shaper`	
Step 26	**class** *class-map-name*	Specifies the name of the class map.
	Example:	
	`Device(config-pmap)#class class-default`	
Step 27	**shape average** *bps*	Shapes traffic to the indicated bit rate.
	Example:	
	`Device(config-pmap-c)#shape average 20000`	
Step 28	**exit**	Exits policy-map class configuration mode.
	Example:	
	`Device(config-pmap-c)#exit`	
Step 29	**exit**	Exits policy-map configuration mode.
	Example:	
	`Device(config-pmap)#exit`	
Step 30	**policy-map** *policy-map-name*	Creates a policy map to specify a service policy.
	Example:	
	`Device(config)# policy-map sub-ifc-shaper`	

	Command or Action	**Purpose**
Step 31	**class** *class-map-name* **Example:** Device(config-pmap)#class class-default	Specifies the name of the class map.
Step 32	**shape average** *bps* **Example:** Device(config-pmap-c)#shape average 40000	Shapes traffic to the indicated bit rate.
Step 33	**exit** **Example:** Device(config-pmap-c)#exit	Exits policy-map class configuration mode.
Step 34	**exit** **Example:** Device(config-pmap)#exit	Exits policy-map configuration mode.
Step 35	**exit policy-map** *policy-map-name* **Example:** Device(config)# policy-map port-shaper	Creates a policy map to specify a service policy.
Step 36	**class** *class-map-name* **Example:** Device(config-pmap)#class class-default	Specifies the name of the class map.
Step 37	**shape average** *bps* **Example:** Device(config-pmap-c)#shape average 60000	Shapes traffic to the indicated bit rate.
Step 38	**exit** **Example:** Device(config-pmap-c)#exit	Exits policy-map class configuration mode.

	Command or Action	Purpose
Step 39	**exit** **Example:** Device(config-pmap)#exit	Exits policy-map configuration mode.
Step 40	**policy-map** *policy-map-name* **Example:** Device(config)# policy-map ingress-police	Creates a policy map to specify a service policy.
Step 41	**class** *class-map-name* **Example:** Device(config-pmap)# class class-default	
Step 42	**police** *bps* **Example:** Device(config-pmap-c)# police 10000	Creates a per-interface policer and configures the policy-map class to use it.
Step 43	**interface pseudowire** *number* **Example:** Device(config-pmap-c-police)# interface pseudowire 1	Configures an interface type and enters interface configuration mode.
Step 44	**encap mpls** **Example:** Device(config-if)# encap mpls	Configures MPLS encapsulation.
Step 45	**neighbor** *peer-address* *vcid-value* **Example:** Device(config-if)# neighbor 10.0.0.1 100	Specifies the peer IP address and virtual circuit (VC) ID value of an L2VPN pseudowire.
Step 46	**service-policy input** *policy-map-name* **Example:** Device(config-if)# service-policy input ingress-policy	Attaches a policy map to an input interface.

	Command or Action	**Purpose**
Step 47	**service-policy output** *policy-map-name* **Example:** `Device(config-if)# service-policy output gold-policy-hqos`	Attaches a policy map to an output interface.
Step 48	**interface gigabit ethernet** *number* **Example:** `Device(config-if)# interface gigabitethernet 1/1/0`	Configures an interface type.
Step 49	**service-policy output** *policy-map-name* **Example:** `Device(config-if)# service-policy output port-shaper`	Attaches a policy map to an output interface.

Creating a Hierarchical Policy for VFI Pseudowires

Perform this task to create a hierarchical policy for VFI Pseudowires.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** *class-map-name*
5. **shape average** *bps*
6. **service-policy** *policy-map*
7. **exit**
8. **exit**
9. **policy-map** *policy-map-name*
10. **class** *class-map-name*
11. **shape average** *bps*
12. **exit**
13. **exit**
14. **policy-map** *policy-map-name*
15. **class** *class-map-name*
16. **shape average** *bps*
17. **exit**
18. **exit**
19. **exit policy-map** *policy-map-name*
20. **class** *class-map-name*
21. **shape average** *bps*
22. **exit**
23. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. **Note** Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	**Purpose**
Step 3	**policy-map** *policy-map-name* **Example:** Device(config)# policy-map gold-policy-hqos	Creates a policy map to specify a service policy.
Step 4	**class** *class-map-name* **Example:** Device(config-pmap)# class class-default	Specifies the name of the class map.
Step 5	**shape average** *bps* **Example:** Device(config-pmap-c)# shape average 10000	Shapes traffic to the indicated bit rate.
Step 6	**service-policy** *policy-map* **Example:** Device(config-pmap-c)# service-policy gold-policy-child	Attaches a policy map to a class.
Step 7	**exit** **Example:** Device(config-pmap-c)# exit	Exits policy-map class configuration mode.
Step 8	**exit** **Example:** Device(config-pmap)# exit	Exits policy-map configuration mode.
Step 9	**policy-map** *policy-map-name* **Example:** Device(config)# policy-map pw-shaper	Creates a policy map to specify a service policy.
Step 10	**class** *class-map-name* **Example:** Device(config-pmap)# class class-default	Specifies the name of the class map.

	Command or Action	Purpose
Step 11	**shape average** *bps* **Example:** `Device(config-pmap-c)# shape average 20000`	Shapes traffic to the indicated bit rate.
Step 12	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode.
Step 13	**exit** **Example:** `Device(config-pmap)# exit`	Exits policy-map configuration mode.
Step 14	**policy-map** *policy-map-name* **Example:** `Device(config)# policy-map sub-ifc-shaper`	Creates a policy map to specify a service policy.
Step 15	**class** *class-map-name* **Example:** `Device(config-pmap)# class class-default`	Specifies the name of the class map.
Step 16	**shape average** *bps* **Example:** `Device(config-pmap-c)# shape average 40000`	Shapes traffic to the indicated bit rate.
Step 17	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode.
Step 18	**exit** **Example:** `Device(config-pmap)# exit`	Exits policy-map configuration mode.

	Command or Action	**Purpose**
Step 19	**exit policy-map** *policy-map-name* **Example:** `Device(config)# policy-map port-shaper`	Creates a policy map to specify a service policy.
Step 20	**class** *class-map-name* **Example:** `Device(config-pmap)# class class-default`	Specifies the name of the class map.
Step 21	**shape average** *bps* **Example:** `Device(config-pmap-c)# shape average 60000`	Shapes traffic to the indicated bit rate.
Step 22	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode.
Step 23	**exit** **Example:** `Device(config-pmap)# exit`	Exits policy-map configuration mode.

Attaching a Policy Map to a VFI Pseudowire

Perform this task to attach a policy map to a VFI Pseudowire.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** *class-map-name*
5. **police** *bps*
6. **interface pseudowire** *number*
7. **encap mpls**
8. **neighbor** *peer-address vcid-value*
9. **service-policy input** *policy-map-name*
10. **service-policy output** *policy-map-name*
11. **interface gigabit ethernet** *number*
12. **service-policy output** *policy-map-name*
13. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. **Note** Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map-name* **Example:** `Device# policy-map ingress-police`	Creates a policy map to specify a service policy.
Step 4	**class** *class-map-name* **Example:** `Device(config-pmap)# class class-default`	Specifies the name of the class map.
Step 5	**police** *bps* **Example:** `Device(config-pmap-c)# police 10000`	Creates a per-interface policer and configures the policy-map class to use it.

	Command or Action	**Purpose**
Step 6	**interface pseudowire** *number* **Example:** `Device(config-pmap-c-police)# interface pseudowire 1`	Configures an interface type and enters interface configuration mode.
Step 7	**encap mpls** **Example:** `Device(config-if)# encap mpls`	Configures MPLS encapsulation.
Step 8	**neighbor** *peer-address vcid-value* **Example:** `Device(config-if)# neighbor 10.0.0.1 100`	Specifies the peer IP address and virtual circuit (VC) ID value of an L2VPN pseudowire.
Step 9	**service-policy input** *policy-map-name* **Example:** `Device(config-if)# service-policy input ingress-policy`	Attaches a policy map to an input interface.
Step 10	**service-policy output** *policy-map-name* **Example:** `Device(config-if)# service-policy output gold-policy-hqos`	Attaches a policy map to an output interface.
Step 11	**interface gigabit ethernet** *number* **Example:** `Device(config-if)# interface gigabit ethernet 1/1/0`	Configures an interface type.
Step 12	**service-policy output** *policy-map-name* **Example:** `Device(config-if)# service-policy output port-shaper`	Attaches a policy map to an output interface.
Step 13	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.

Configuring VFI with Two Pseudowire Members with Different QoS Policies

Perform this task to configure VFI with two pseudowire members with different QoS policies.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface pseudowire** *number*
4. **encap mpls**
5. **neighbor** *peer-address vcid value*
6. **service-policy output** *policy-map-name*
7. **interface pseudowire** *number*
8. **encap mpls**
9. **neighbor** *peer-address vcid value*
10. **service-policy output** *policy-map-name*
11. **l2vpn vfi context** *name*
12. **vpn id** *vpn-id*
13. **member pseudowire** *pw-int-number*
14. **member pseudowire** *pw-int-number*
15. **bridge-domain** *bridge-domain-id*
16. **member** *interface-type-number*
17. **interface BDI** *number*
18. **ip vrf forwarding** *vrf-name*
19. **ip address** *ip-address mask*

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. **Note** Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	**Purpose**
Step 3	**interface pseudowire** *number* **Example:** `Device# interface pseudowire 1`	Configures an interface type and enters interface configuration mode.
Step 4	**encap mpls** **Example:** `Device(config-if)# encap mpls`	Configures MPLS encapsulation.
Step 5	**neighbor** *peer-address vcid value* **Example:** `Device(config-if)# neighbor 10.0.0.1 100`	Specifies the peer IP address and virtual circuit (VC) ID value of an L2VPN pseudowire.
Step 6	**service-policy output** *policy-map-name* **Example:** `Device(config-if)# service-policy output gold-policy`	Attaches a policy map to an output interface.
Step 7	**interface pseudowire** *number* **Example:** `Device(config-if)# interface pseudowire 2`	Configures an interface type.
Step 8	**encap mpls** **Example:** `Device(config-if)# encap mpls`	Configures MPLS encapsulation.
Step 9	**neighbor** *peer-address vcid value* **Example:** `Device(config-if)# neighbor 20.0.0.1 100`	Specifies the peer IP address and VCID of an L2VPN pseudowire.
Step 10	**service-policy output** *policy-map-name* **Example:** `Device(config-if)# service-policy output silver-policy`	Attaches a policy map to an output interface.

	Command or Action	**Purpose**
Step 11	**l2vpn vfi context** *name* **Example:** Device(config-if)# l2vpn vfi context my-vfi	Establishes a Layer 2 VPN (L2VPN) virtual forwarding interface (VFI) between two or more separate networks.
Step 12	**vpn id** *vpn-id* **Example:** Device(config-vfi)# vpn id 100	Sets a VPN ID on a Virtual Private LAN Services (VPLS) instance.
Step 13	**member pseudowire** *pw-int-number* **Example:** Device(config-vfi)# member pseudowire 1	Specifies the devices that form a point-to-point Layer 2 VPN (L2VPN) virtual forwarding interface (VFI) connection.
Step 14	**member pseudowire** *pw-int-number* **Example:** Device(config-vfi)# member pseudowire 2	Specifies the devices that form a point-to-point Layer 2 VPN (L2VPN) virtual forwarding interface (VFI) connection.
Step 15	**bridge-domain** *bridge-domain-id* **Example:** Device(config-vfi)# bridge-domain 100	Configures components on a bridge domain.
Step 16	**member** *interface-type-number* **Example:** Device(config-bdomain)# member vfi my-vfi	Binds a service instance to a bridge domain instance.
Step 17	**interface BDI** *number* **Example:** Device(config-bdomain)# interface BDI 100	Configures an interface type and enters interface configuration mode.
Step 18	**ip vrf forwarding** *vrf-name* **Example:** Device(config-if)# ip vrf forwarding MY-VRF	Associates a Virtual Private Network (VPN) routing and forwarding (VRF) instance with an interface or subinterface.
Step 19	**ip address** *ip-address mask* **Example:** Device(config-if)# ip address 30.0.0.1 255.255.255.0	Sets a primary or secondary IP address for an interface.

Configuring VFI with Two Pseudowire Members with the Same QoS Policy

Perform this task to configure VFI with two pseudowire members with the same QoS policy.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **template type pseudowire** *name*
4. **encap mpls**
5. **service-policy output** *policy-map-name*
6. **interface pseudowire** *number*
7. **encap mpls**
8. **neighbor** *peer-address vcid value*
9. **source template type pseudowire** *template-name*
10. **interface pseudowire** *number*
11. **encap mpls**
12. **neighbor** *peer-address vcid value*
13. **source template type pseudowire** *template-name*
14. **l2vpn vfi context** *name*
15. **vpn id** *vpn-id*
16. **member pseudowire** *pw-int-number*
17. **member pseudowire** *pw-int-number*
18. **bridge-domain** *bridge-domain-id*
19. **member** *interface-type-number*
20. **interface BDI** *number*
21. **ip vrf forwarding** *vrf-name*
22. **ip address** *ip-address mask*

DETAILED STEPS

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

	Command or Action	**Purpose**
Step 2	**configure terminal** **Example:** Device# configure terminal	Enters global configuration mode.
Step 3	**template type pseudowire** *name* **Example:** Device(config)# template type pseudowire my_template	Configures a template.
Step 4	**encap mpls** **Example:** Device(config-if)# encap mpls	Configures MPLS encapsulation.
Step 5	**service-policy output** *policy-map-name* **Example:** Device(config-template)# service-policy output common-policy	Attaches a policy map to a output interface.
Step 6	**interface pseudowire** *number* **Example:** Device(config-if)# interface pseudowire 1	Configures an interface type.
Step 7	**encap mpls** **Example:** Device(config-if)# encap mpls	Configures MPLS encapsulation.
Step 8	**neighbor** *peer-address vcid value* **Example:** Device(config-if)# neighbor 10.0.0.1 100	Specifies the peer IP address and VCID of an L2VPN pseudowire.
Step 9	**source template type pseudowire** *template-name* **Example:** Device(config-if)# source template type pseudowire my_template	Configures the name of a source template of type pseudowire.

	Command or Action	**Purpose**
Step 10	**interface pseudowire** *number* **Example:** Device(config-if)# interface pseudowire 2	Configures an interface type.
Step 11	**encap mpls** **Example:** Device(config-if)# encap mpls	Configures MPLS encapsulation.
Step 12	**neighbor** *peer-address vcid value* **Example:** Device(config-if)# neighbor 20.0.0.1 100	Specifies the peer IP address and VCID of an L2VPN pseudowire.
Step 13	**source template type pseudowire** *template-name* **Example:** Device(config-if)# source template type pseudowire my_template	Configures the name of a source template of type pseudowire.
Step 14	**l2vpn vfi context** *name* **Example:** Device(config-if)# l2vpn vfi context my-vfi	Establishes a Layer 2 VPN (L2VPN) virtual forwarding interface (VFI) between two or more separate networks.
Step 15	**vpn id** *vpn-id* **Example:** Device(config-vfi)# vpn id 100	Sets a VPN ID on a Virtual Private LAN Services (VPLS) instance.
Step 16	**member pseudowire** *pw-int-number* **Example:** Device(config-vfi)# member pseudowire 1	Specifies the devices that form a point-to-point Layer 2 VPN (L2VPN) virtual forwarding interface (VFI) connection.
Step 17	**member pseudowire** *pw-int-number* **Example:** Device(config-vfi)# member pseudowire 2	Specifies the devices that form a point-to-point Layer 2 VPN (L2VPN) virtual forwarding interface (VFI) connection.
Step 18	**bridge-domain** *bridge-domain-id* **Example:** Device(config-vfi)# bridge-domain 100	Configures components on a bridge domain.

	Command or Action	Purpose
Step 19	**member** *interface-type-number* **Example:** Device(config-bdomain)# member vfi my-vfi	Binds a service instance to a bridge domain instance.
Step 20	**interface BDI** *number* **Example:** Device(config-bdomain)# interface BDI 100	Configures an interface type and enters interface configuration mode.
Step 21	**ip vrf forwarding** *vrf-name* **Example:** Device(config-if)# ip vrf forwarding MY-VRF	Associates a Virtual Private Network (VPN) routing and forwarding (VRF) instance with an interface or subinterface.
Step 22	**ip address** *ip-address mask* **Example:** Device(config-if)# ip address 30.0.0.1 255.255.255.0	Sets a primary or secondary IP address for an interface.

Configuring VFI with Auto Discovered Pseudowires

Perform this task to configure VFI with auto discovered pseudowires.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **template type pseudowire** *name*
4. **encap mpls**
5. **service-policy output** *policy-map-name*
6. **l2vpn vfi context** *name*
7. **vpn id** *vpn-id*
8. **autodiscovery bgp signaling ldp template** *template-name*
9. **bridge-domain** *bridge-domain-id*
10. **member** *interface-type-number*
11. **interface BDI** *number*
12. **ip vrf forwarding** *vrf-name*
13. **ip address** *ip-address mask*

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. **Note** Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**template type pseudowire** *name*
	
	Example:
	
	`Device(config)# template type pseudowire`
	`my_template`
		Configures a template.

Step 4	**encap mpls** **Example:** `Device(config-if)# encap mpls`	Configures MPLS encapsulation.

Step 5	**service-policy output** *policy-map-name*
	
	Example:
	
	`Device(config-template)# service-policy output`
	`common-policy`
		Attaches a policy map to a output interface.

Step 6	**l2vpn vfi context** *name* **Example:** `Device(config-if)# l2vpn vfi context my-vfi`	Establishes a Layer 2 VPN (L2VPN) virtual forwarding interface (VFI) between two or more separate networks.
Step 7	**vpn id** *vpn-id* **Example:** `Device(config-vfi)# vpn id 100`	Sets a VPN ID on a Virtual Private LAN Services (VPLS) instance.

Step 8	**autodiscovery bgp signaling ldp template** *template-name*
	
	Example:
	
	`Device(config-vfi)# autodiscovery bgp signaling`
	`ldp template my_template`
		Designates a Layer 2 virtual forwarding interface (VFI) as having Label Distribution Protocol (LDP) autodiscovered pseudowire members.

	Command or Action	Purpose
Step 9	**bridge-domain** *bridge-domain-id* **Example:** `Device(config-vfi)# bridge-domain 100`	Configures components on a bridge domain.
Step 10	**member** *interface-type-number* **Example:** `Device(config-bdomain)# member vfi my-vfi`	Binds a service instance to a bridge domain instance.
Step 11	**interface BDI** *number* **Example:** `Device(config-bdomain)# interface BDI 100`	Configures an interface type and enters interface configuration mode.
Step 12	**ip vrf forwarding** *vrf-name* **Example:** `Device(config-if)# ip vrf forwarding MY-VRF`	Associates a Virtual Private Network (VPN) routing and forwarding (VRF) instance with an interface or subinterface.
Step 13	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 30.0.0.1 255.255.255.0`	Sets a primary or secondary IP address for an interface.

Configuration Examples for QoS Policies for VFI Pseudowires

Example: Configuring QoS Policies for Pseudowires

The following example shows how to QoS policies for pseudowires:

```
Device(config)# policy-map GOLD-POLICY-CHILD
Device(config-pmap)# class PRIORITY-CLASS
Device(config-pmap-c)# priority 100
Device(config-pmap-c)# exit
Device(config-pmap)# class GUARANTEE-CLASS
Device(config-pmap-c)# bandwidth 1000
Device(config-pmap-c)# exit
Device(config-pmap)# class LIMITED-CLASS
Device(config-pmap-c)# police cir 8000
Device(config-pmap-c-police)# class class-default
Device(config-pmap-c)# shape average 8000
Device(config-pmap-c)# queue-limit 150
Device(config-pmap-c)# random-detect
Device(config-pmap-c)# exit
```

```
Device(config-pmap)# exit
Device(config)# policy-map GOLD-POLICY-HQOS
Device(config-pmap)# class class-default
Device(config-pmap-c)# shape average 8000
Device(config-pmap-c)# service-policy GOLD-POLICY-CHILD
Device(config-pmap-c)# exit
Device(config-pmap)# exit
Device(config)# policy-map PW-SHAPER
Device(config-pmap)# class class-default
Device(config-pmap-c)# shape average 8000
Device(config-pmap-c)# exit
Device(config-pmap)# exit
Device(config)# policy-map SUB-IFC-SHAPER
Device(config-pmap)# class class-default
Device(config-pmap-c)# shape average 10000
Device(config-pmap-c)# exit
Device(config-pmap)# exit
Device(config)# policy-map PORT-SHAPER
Device(config-pmap)# class class-default
Device(config-pmap-c)# shape average 20000
Device(config-pmap-c)# exit
Device(config-pmap)# exit
Device(config)# policy-map INGRESS-POLICE
Device(config-pmap)# class class-default
Device(config-pmap-c)# police 10000
Device(config-pmap-c-police)# interface pseudowire 1
Line protocol on Interface pseudowire0, changed state to up
Device(config-if)# encap mpls
Device(config-if)# neighbor 10.0.0.1 100
Device(config-if)# service-policy input INGRESS-POLICY
Device(config-if)# service-policy output GOLD-POLICY-HQOS
Device(config-if)# interface GigabitEthernet 1/1/0
--- Pseudowire is going out through this interface
Device(config-if)# service-policy output PORT-SHAPER
```

Example: Configuring VFI with Two Pseudowire Members with Different QoS Policies

The following example shows how to configure VFI with two pseudowire members with different QoS policies:

```
Device(config)# interface pseudowire1
Line protocol on Interface pseudowire0, changed state to up
Device(config-if)# encap mpls
Device(config-if)# neighbor 10.0.0.1 100
Device(config-if)# service-policy output GOLD-POLICY
Device(config-if)# interface pseudowire2
Device(config-if)# encap mpls
Device(config-if)# neighbor 20.0.0.1 100
Device(config-if)# service-policy output SILVER-POLICY
Device(config-if)# l2vpn vfi context MY-VFI
Device(config-vfi)# vpn id 100
Device(config-vfi)# member pseudowire1
Device(config-vfi)# member pseudowire2
Device(config-vfi)# bridge-domain 100
Device(config-bdomain)# member vfi MY-VFI
STATUS_CHANGED: Status of VFI my-vfi changed from DOWN to UP
Device(config-bdomain)# interface BDI 100
Device(config-if)# ip vrf forwarding MY-VRF
Device(config-if)# ip address 30.0.0.1 255.255.255.0
```

Example: Configuring VFI with Two Pseudowire Members with the Same QoS Policy

The following example shows how to configure VFI with two pseudowire members with the same QoS policy:

```
Device(config)# template type pseudowire MY_TEMPLATE
Device(config-template)# encapsulation mpls
Device(config-template)# service-policy output COMMON-POLICY
Device(config-template)# interface pseudowire1
Line protocol on Interface pseudowire0, changed state to up
Device(config-if)# encap mpls
Device(config-if)# neighbor 10.0.0.1 100
Device(config-if)# source template type pseudowire MY_TEMPLATE
Device(config-if)# interface pseudowire2
Device(config-if)# encap mpls
Device(config-if)# neighbor 20.0.0.1 100
Device(config-if)# source template type pseudowire MY_TEMPLATE
Device(config-if)# l2vpn vfi context MY-VFI
Device(config-vfi)# vpn id 100
Device(config-vfi)# member pseudowire1
Device(config-vfi)# member pseudowire2
Device(config-vfi)# bridge-domain 100
Device(config-bdomain)# member vfi MY-VFI
Status of VFI my-vfi changed from DOWN to UP
Device(config-bdomain)# interface BDI 100
Device(config-if)# ip vrf forwarding MY-VRF
Device(config-if)# ip address 30.0.0.1 255.255.255.0
```

Example: Configuring VFI with Auto Discovered Pseudowires

The following example shows how to configure VFI with auto discovered pseudowires:

```
Device(config)# template type pseudowire MY_TEMPLATE
Device(config-template)# encapsulation mpls
Device(config-template)# service-policy output COMMON-POLICY
Device(config-template)# l2vpn vfi context MY-VFI
Device(config-vfi)# vpn id 100
Line protocol on Interface pseudowire0, changed state to up
Device(config-vfi)# autodiscovery bgp signaling ldp template MY_TEMPLATE
Device(config-vfi-autodiscovery)# bridge-domain 100
Device(config-bdomain)# member vfi MY-VFI
Status of VFI my-vfi changed from DOWN to UP
Device(config-bdomain)# interface BDI 100
Device(config-if)# ip vrf forwarding MY-VRF
Device(config-if)# ip address 30.0.0.1 255.255.255.0
```

Example: Displaying Pseudowire Policy Map Information

The following is sample output from the **show policy-map interface** command which shows class maps and policy maps configured for the pseudowire 2 interface:

```
Device#show policy-map interface pseudowire2
 pseudowire2

 Service-policy output: pw_brr

   Class-map: prec1 (match-all)
     0 packets, 0 bytes
     30 second offered rate 0000 bps, drop rate 0000 bps
```

```
      Match: ip precedence 1
      Queueing
      queue limit 4166 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 0/0
      bandwidth remaining ratio 1

    Class-map: prec2 (match-all)
      0 packets, 0 bytes
      30 second offered rate 0000 bps, drop rate 0000 bps
      Match: ip precedence 2
      Queueing
      queue limit 4166 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 0/0
      bandwidth remaining ratio 2

    Class-map: prec3 (match-all)
      0 packets, 0 bytes
      30 second offered rate 0000 bps, drop rate 0000 bps
      Match: ip precedence 3
      Queueing
      queue limit 4166 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 0/0
      bandwidth remaining ratio 3

    Class-map: class-default (match-any)
      0 packets, 0 bytes
      30 second offered rate 0000 bps, drop rate 0000 bps
      Match: any
      Queueing
      queue limit 4166 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 0/0
      bandwidth remaining ratio 4
Device#
```

Additional References for QoS Policies for VFI Pseudowires

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS commands	*Cisco IOS Multiprotocol Label Switching Command Reference*
QoS commands	*Cisco IOS Quality of Service Solutions Command Reference*
Configuring the pseudowire class	"Any Transport over MPLS"
Layer 2 VPN	• Any Transport over MPLS • L2VPN Pseudowire Switching • MPLS LSP Ping/Traceroute for LDP/TE, and LSP Ping for VCCV

Related Topic	Document Title
L2VPN pseudowires	• L2VPN Pseudowire Redundancy • MPLS Pseudowire Status Signaling

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 QoS Policies for VFI Pseudowires

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 QoS Policies for VFI Pseudowire

Feature Name	Releases	Feature Information
QoS Policies for VFI Pseudowires	Cisco IOS XE 3.8S	This features allows you to configure QoS classes and policies for use on VFI pseudowire members. The following commands were introduced or modified: **show policy-map interface**.