Modular QoS Deployment Scenarios

This module provides deployment scenarios use cases for specific QoS features or for QoS implementations of features that are described in other technology guides such as L2VPN or MPLS.

Feature History for QoS Deployment Scenarios on Cisco IOS XR Software

Release	Modification
Release 3.2.2	The QoS with SRP feature was introduced.
Release 3.8.0	The Qos on Multicast VPN feature was introduced. The VPLS QoS feature was introduced.
Release 4.0.0	The QinQ QoS feature was introduced(for Layer 2 only). The Hierarchical VPLS QoS feature was introduced.
Release 4.1.0	QoS on Pseudowire Headend (PWHE) Interfaces feature was introduced. Information regarding match criteria for inner VLAN was added to this document.
Release 5.3.2	VPLS(QoS) CRS fetaure was introduced on CRS-X.

Hierarchical VPLS QoS

Hierarchical VPLS (H-VPLS) is an extension of basic VPLS to provide scaling and operational benefits. H-VPLS partitions a network into several edge domains that are interconnected using an MPLS core. H-VPLS

provides a solution to deliver Ethernet multipoint services over MPLS. The use of Ethernet switches at the edge offers significant technical and economic advantages. H-VPLS also allows Ethernet point-to-point and multipoint Layer 2 VPN services, as well as Ethernet access to high-speed Internet and IP VPN services.

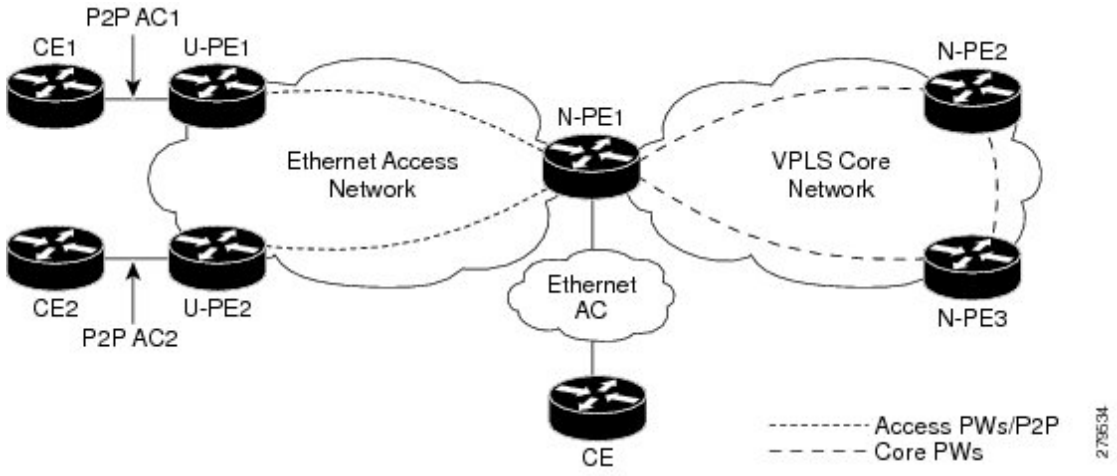
Note The Cisco CRS Series Modular Services Card 140G (CRS-MSC-140G) does not support Hierarchical VPLS QoS.

Hierarchical VPLS with Pseudowire Access

This figure shows the pseudowire (PW) access for H-VPLS. The edge domain can be an MPLS access network. In an H-VPLS configuration, the N-PE router forwards L2VPN packets from an Access PW to a Core PW and also forwards packets from a Core PW to an Access PW.

When forwarding packets, the N-PE router removes the virtual circuit (VC) label and also adds a VC label. The **set mpls experimental topmost** command, the **set mpls experimental imposition** command, or both of these commands can be applied to a QoS policy in a PW <-> PW configuration. Standard VPLS QoS actions, such as **match vlan inner**, **match cos inner**, and **set cos inner**, apply to H-VPLS QoS as well.

Figure 1: Pseudowire Access for H-VPLS



This table summarizes the actions taken with respect to the MPLS experimental value in a PW<-> PW configuration.

	Ingress/Egress	
Default	Ingress Egress	MPLS experimental value from the VC is copied to all imposed labels
set mpls experimental topmost	Ingress	MPLS experimental value from the QoS policy is copied to all imposed labels
set mpls experimental topmost	Egress	MPLS experimental value from the QoS policy is copied to the outermost labels

	Ingress/Egress	
set mpls experimental imposition	Ingress	MPLS experimental value from the QoS policy is copied to all imposed labels
set mpls experimental topmost and **set mpls experimental imposition**	Ingress	MPLS experimental value from the **set mpls experimental imposition** command in the QoS policy is copied to all imposed labels

Hierarchical VPLS QoS: Example

In this example, the N-PE router forwards L2VPN packets from an Access PW to a Core PW and also forwards packets from a Core PW to an Access PW. The example defines classes and specifies experimental values in the topmost MPLS label as match criteria for a class map to match the MPLS label. A QOS policy is applied to the ingress Core interfaces in the MPLS Access Network or the MPLS Core Network.

In this H-VPLS QoS configuration, the U-PE (customer) router and the N-PE (service provider) router have different QoS policies. The policy on the U-PE router specifies a high experimental value so traffic can receive more bandwidth and is sent as fast as possible to minimize jitter and delay. (topmost values of 4, 5, and 6.)

The N-PE router needs to balance traffic from various sources. The N-PE router remarks the packets by adding a label with a lower experimental value (imposition values of 1 and topmost values of 2).

```
class-map match-any exp1-top
 match mpls experimental topmost 4
 end-class-map
!
class-map match-any exp2-top
 match mpls experimental topmost 5
 end-class-map
!
class-map match-any exp3-top
 match mpls experimental topmost 6
 end-class-map

policy-map RX-Core
 class exp1-top
  set mpls experimental imposition 1
  shape average percent 20
 !
 class exp2-top
  set mpls experimental imposition 1
  set mpls experimental topmost 2
    shape average percent 20

 class exp3-top
  set mpls experimental topmost 2
  shape average percent 20

 class class-default
  set mpls experimental imposition 1
  shape average percent 20
 !
 end-policy-map
```

QinQ QoS

IEEE 802.1Q-in-Q VLAN Tagging expands the VLAN space by tagging the already tagged packets, thus producing a “double-tagged” frame. Double-tagging is useful for service providers, because it allows them to use VLANs internally while mixing traffic from clients that are already VLAN-tagged.

Q-in-Q allows service providers to use a single VLAN to transport most or all of a single customer's VLANs (inner VLAN tag) across their MAN/WAN backbone. The service provider adds an extra 802.1Q tag (outer VLAN tag) to customer traffic in the switches or routers at the edge of the service provider network. The outer VLAN tag assigns a unique VLAN ID to each customer in the service provider network. The outer VLAN tag keeps each customer's VLAN traffic segregated and private.

In the below figure, Customer ABC and Customer XYZ use the same VLAN ID of 27. However, these customers do not receive each other’s traffic, because the service provider tags each customer’s incoming frame with a unique outer VLAN tag. Customer ABC is assigned outer VLAN Tag 9 and Customer XYZ is assigned outer VLAN tag 10.

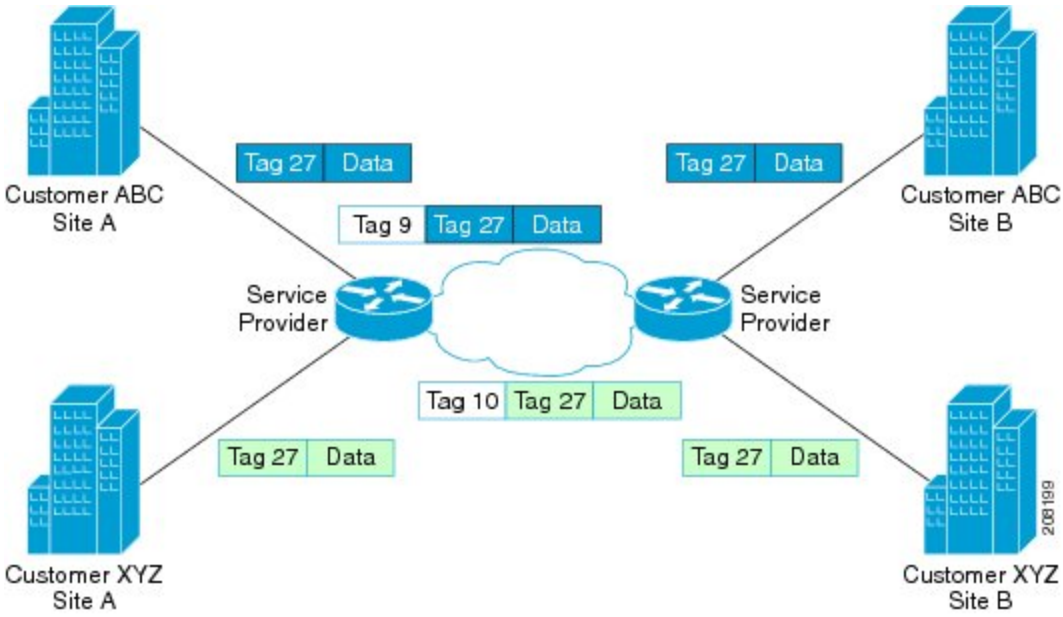
Classify and mark traffic for a specific customer, such as Customer XYZ, based on the inner (customer) VLAN tag using the **inner** keyword with the **match cos**, **match vlan**, and **set cos** commands.

The **match cos**, **match vlan**, and **set cos** commands classify and mark any traffic that matches the outer (service provider) VLAN tag; this traffic includes Customer XYZ traffic and customer ABC traffic.

The service provider removes the outer VLAN tag on traffic that exits the service provider network before it is sent to Customer ABC and Customer XYZ.

Note Some customers use three layers of VLAN tags. In this case, using the **inner** keyword with the **match cos** and **match vlan** commands classifies traffic traffic on the VLAN tag, which is right next to the outermost VLAN tag. Using the **inner** keyword with the **set cos** command marks CoS bits of the VLAN tag which is right next to the outermoset VLAN tag (what this document refers to as the inner (customer) VLAN tag).

Figure 2: QinQ Tagging



Note The CRS-MSC-140G does not support QinQ QoS.

Q-in-Any QoS

As in a QinQ configuration, a Q-in-Any interface receives “double-tagged” packets but only the outer VLAN ID is specified and inner VLAN ID can be any number in the range of 1-4094. Traffic on a Q-in-Any interface can be classified and marked based on the inner (customer) VLAN tag using the **inner** keyword with the **match cos**, **match vlan** and **set cos** commands.

Marking and Classifying Packets

To classify and mark packets in QinQ QoS or Q-in-Any QoS configurations based on the inner (customer) VLAN tag, use the **inner** keyword in the following commands:

	Supported On	On Layer 2	On Layer 3
match cos inner	ingress egress	Main interfaces Subinterfaces	Main interfaces

	Supported On	On Layer 2	On Layer 3
match vlan inner	ingress egress	Main interfaces Subinterfaces	Main interfaces
set cos inner	**egress: conditional and unconditional marking**	Main interfaces Subinterfaces	Main interfaces

Match on Inner CoS: Example

In this example, traffic with an outer VLAN ID of 2 and an inner VLAN ID of 3 enters the QinQ attachment circuit (AC). If traffic has a CoS value of 1, 3, or 5 based on the inner VLAN tag, it matches class ic.

```
policy-map p2
 class ic
  police rate percent 30
  !
  bandwidth remaining percent 40
 !
 class class-default
 !
 end-policy-map
!
class-map match-any ic
 match cos inner 1 3 5
 end-class-map
!
interface GigabitEthernet0/5/0/0.2 l2transport
 dot1q vlan 2 3
 service-policy input p2
!
```

Match Criteria for Inner VLANs: Example

This section provides guidance for configuring match criteria for inner VLANs on Q-in-Any and QinQ attachment circuits (ACs).

Match Criteria for Inner VLAN on Q-in-Any AC

In this example, all traffic with an outer VLAN ID of 1 enters the Q-in-Any AC, but the inner VLAN ID can be any value. However, only the traffic with an inner VLAN ID of 1, 2, or 3 matches the class provisioned in the policy-map (that is, class iv).

```
policy-map p1
 class iv
  shape average percent 30
  set qos-group 1
 !
 class class-default
 !
 end-policy-map
!
class-map match-any iv
 match vlan inner 1 2 3
 end-class-map
```

```
!
interface GigabitEthernet0/5/0/0.1 l2transport
 dot1q vlan 1 any
 service-policy input p1
!
```

Match Criteria for Inner VLAN on QinQ AC

The treatment of match criteria for inner VLAN on a QinQ attachment circuit (AC) does not function in the same manner as it does on a Q-in-Any AC (as described in the Match Criteria for Inner VLANs: Example). You should *not* attempt to configure match criteria for inner VLAN on QinQ ACs.

Explanation

During the configuration process, it is possible to configure an invalid match on the inner VLAN and commit the configuration. However, if you configure a match on inner VLAN with an ID different from the inner VLAN ID configured on the QinQ AC, the traffic on the AC might not be directed to the correct class.

Caution We recommend that you *not* configure match criteria for inner VLAN on QinQ ACs.

Example

In this example, the user has configured class iv1 with match on inner VLAN 1, but has configured an inner VLAN ID of 2 on the QinQ AC. Now, if traffic with an outer VLAN of 4 and inner VLAN of 2 enters the QinQ AC, the system matches the first class with inner VLAN ID (class iv1).

```
interface GigabitEthernet0/5/0/0.4 l2transport
 dot1q vlan 4 2
 service-policy input p1
!

policy-map p1
 class iv1
  shape average percent 30
  set qos-group 1
 !
 class iv2
  bandwidth remaining percent 30
 !
 class class-default
 !
 end-policy-map
!

class-map match-any iv1
 match vlan inner 1
 end-class-map
!

class-map match-any iv2
 match vlan inner 2
 end-class-map
!
```

Set Inner CoS: Example

In this example, traffic with outer VLAN ID of 3 and inner VLAN ID of 2 exits through the QinQ AC. If the traffic matches class qg1 or class qg2, it is marked with the inner CoS value specified in class qg1 or class qg2, respectively.

```
policy-map p3
 class qg1
  police rate percent 30 peak-rate percent 50
   conform-action set cos inner 1
   exceed-action set cos inner 2
   violate-action set cos inner 3
  !
 !
 class qg2
  set cos inner 4
 !
 class class-default
 !
 end-policy-map
!
class-map match-any qg1
 match qos-group 1
 end-class-map
!
class-map match-any qg2
 match qos-group 2
 end-class-map
!
interface GigabitEthernet0/5/0/0.3 l2transport
 dot1q vlan 3 2
 service-policy output p3
!
```

QoS on Multicast VPN

QoS on Multicast VPN: Example

Supporting QoS in an mVPN-enabled network requires conditional and unconditional marking of the DSCP or precedence bits onto the tunnel header. Unconditional marking marks the DSCP or precedence tunnel as a policy action. Conditional marking marks the DSCP or precedence values on the tunnel header as a policer action (conform, exceed, or violate).

Unconditional Marking

```
class-map c1
  match vlan 1-10

policy-map p1
 class c1
  set precedence tunnel 3
```

Conditional Marking

```
policy-map p2
 class c1

  police rate percent 50
  conform action set dscp tunnel af11
  exceed action set dscp tunnel af12
```

QoS with SRP

Spatial bandwidth reuse (SRP) is possible due to the packet destination-stripping property of SRP. Older technologies incorporate source stripping, where packets traverse the entire ring until they are removed by the source. Even if the source and destination nodes are next to each other on the ring, packets continue to traverse the entire ring until they return to the source to be removed. SRP provides more efficient use of available bandwidth by having the destination node remove the packet after it is read. This provides more bandwidth for other nodes on the SRP ring.

SRP rings consists of two counter rotating fibers, known as outer and inner rings, both concurrently used to carry data and control packets. SRP uses both explicit control packets and control information piggybacked inside data packets (control packets handle tasks such as keepalives, protection switching, and bandwidth control propagation). Control packets propagate in the opposite direction from the corresponding data packets, ensuring that the data takes the shortest path to its destination. The use of dual fiber-optic rings provides a high-level of packet survivability. In the event of a failed node or a fiber cut, data is transmitted over the alternate ring.

SRP rings are media independent and can operate over a variety of underlying technologies, including SONET/SDH, wavelength division multiplexing (WDM), and dark fiber. This ability to run SRP rings over any embedded fiber transport infrastructure provides a path to packet-optimized transport for high- bandwidth IP networks.

To distinguish between the two rings, one is referred to as the “inner” ring and the other as the “outer” ring. SRP operates by sending data packets in one direction (downstream) and sending the corresponding control packets in the opposite direction (upstream) on the other fiber. This allows SRP to use both fibers concurrently to maximize bandwidth for packet transport and to accelerate control signal propagation for adaptive bandwidth utilization, and for self-healing purposes.

Note The CRS-MSC-140G does not support QoS on SRP interfaces.

QoS with SRP: Example

This example shows how to configure two quality-of-service (QoS) classes. One is for voice traffic and is identified by an MPLS experimental bit value of 4; the second is control traffic that is identified by an IP precedence value of 6. Both classes of traffic are sent to the SRP high priority queue and are marked with high SRP priority (4 and 6).

```
Last configuration change at 04:56:06 UTC Tue Sep 06 2005 by lab
!
hostname router
```

```
class-map match-any ctrl
 match precedence internet
!
class-map match-any voice
 match mpls experimental topmost 4
!
policy-map srp-policy
 class voice
  police cir 2000000
  set cos 4
  priority
 !
 class ctrl
  priority
  set cos 6
 !
!
interface SRP0/7/0/0
 description "Connected to 3-nodes ring"
 service-policy output srp-policy
 ipv4 address 30.30.30.2 255.255.255.0
```

VPLS and VPWS QoS

To support QoS on a virtual private LAN service (VPLS)-enabled network, packets can be classified based on these VPLS-specific match criteria:

- Match on vpls broadcast
- Match on vpls known
- Match on vpls unknown
- Match on vpls multicast
-

Note VPLS-specific classification is performed only in the ingress direction. VPLS-specific and VPWS-specific classification are performed only in the ingress direction.

This example illustrates a typical VPLS topology with this configuration (in PE class c1):

```
class c1
  match vpls known
!
class c2
  match vpls unknown
!
class c3
  match vpls multicast
!
class c4
  match vpls broadcast
!
policy-map p1
  class c1
   set qos-group2
```

```
!
class c2
  set qos-group3
!
class c3
  set discard-class4
!
class c4
  set discard-class 5
!
```

The following figure illustrates a typical VPLS topology. The VPLS network is a mesh of pseudowires (PWs) interconnected to bridge domains in the routers. Each of the provider edge (PE) routers has a bridge domain. Each PW is a bridge port into the bridge domain. The customer edge (CE) connection into each PE router is an attachment circuit (AC) bridge port into the same bridge domain. QoS configuration commands are applied to the AC that connects to the CE router on the one end and the bridge domain of the PE router on the other.

VPLS and VPWS QoS: Example

In the VPLS-enabled network:

- If a unicast packet arrives on the ingress interface of the PE router with a known MAC address (which means the destination MAC address of the packet is found in the MAC forwarding table), it matches class c1.
- If a unicast packet arrives on the ingress interface of the PE router with an unknown MAC address (which means the destination MAC address of the packet is not found in the MAC forwarding table), it matches class c2.
- If a VPLS multicast packet arrives on the ingress interface of the PE router, it matches class c3.
- If a VLPS broadcast packet arrives on the ingress interface of the PE router, it matches class c4.

The packets that meet the VPLS-specific match criteria receive QoS treatment according to the policy actions defined in the policy.

Note Packets with unknown destination MAC address, multicast packets, and broadcast packets are flooded.

Note The CRS-MSC-140G does not support VPLS QoS.

Note The CRS-X supports VPLS QoS.

VPLS QoS: Example

In this example, the packets that meet the VPLS-specific match criteria receive QoS treatment according to the policy actions defined in the policy. Apply the policy to the VPLS AC interface.

```
class-map match-any c1
 match vpls known
 end-class-map
!

class-map match-any c2
 match vpls unknown
 end-class-map
!

class-map match-any c3
 match vpls broadcast
 end-class-map
!

class-map match-any c4
 match vpls multicast
 end-class-map
!

policy-map p2
 class c1
  set qos-group 3
  set mpls experimental imposition 4
  shape average percent 40
 !

 class c2
  bandwidth remaining percent 10
  set mpls experimental imposition 5
 !

 class c3
  police rate percent 30
  set mpls experimental imposition 6
  !

 class c4
  bandwidth remaining percent 10
  set mpls experimental imposition 7
  !

class class-default
 !
 end-policy-map
!

class-map match-any c1
 match vpls known
 end-class-map
!

class-map match-any c2
 match vpls unknown
 end-class-map
!

class-map match-any c3
 match vpls multicast
 end-class-map
!

policy-map p2
```

```
 class c1
  set qos-group 3
  set mpls experimental imposition 4
  shape average percent 40
 !

 class c2
  bandwidth remaining percent 10
  set mpls experimental imposition 5
 !

 class c3
  bandwidth remaining percent 10
  set mpls experimental imposition 7
  !

class class-default
 !
 end-policy-map
!
```

QoS on Pseudowire Headend

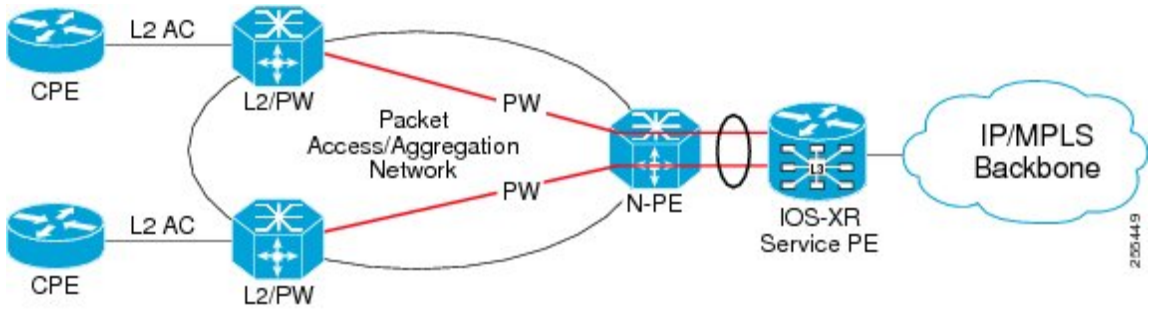
A pseudowire (PW) headend (PWHE) virtual interface originates as a PW on an access node (the Layer 2 PW feeder node) and terminates on a Layer 3 service instance, such as a VRF instance, on the service provider router (Cisco CRS Router). All ingress and egress QoS functions can be configured on the interface, including policing, shaping, queuing, and hierarchical policies.

Note The system supports only hierarchical QoS policy on the PWHE interface, a single default class in the parent policy, and policing or shaping on the parent. Nonhierarchical policies are not supported.

Note The Cisco CRS Series Modular Services Card 140G (CRS-MSC-140G) does not support QoS on PWHE interfaces.

This figure illustrates PWs originating on access node (L2/PW feeder node) and terminating on the service provider edge (Service PE). The composite Layer 2 attachment circuit (AC) plus PW segment forms a point-to-point virtual link that functions and behaves the same as traditional Layer 3 links. Although the PWHE interface is virtual, the traffic corresponding to that interface is sent and received over a single physical interface.

Figure 3: Example of Pseudowire Headend Virtual Interface



Note For information on creating the PWHE interface, see *Cisco IOS XR Virtual Private Network Configuration Guide for the Cisco CRS Router*.

The QoS policy on PWHE virtual interface can coexist with the QoS policy on its underlying physical interfaces. These two figures (one each for the ingress and egress directions) show the application of QoS for physical and PWHE traffic.

Figure 4: Ingress QoS Policy Application for Physical and PWHE Traffic

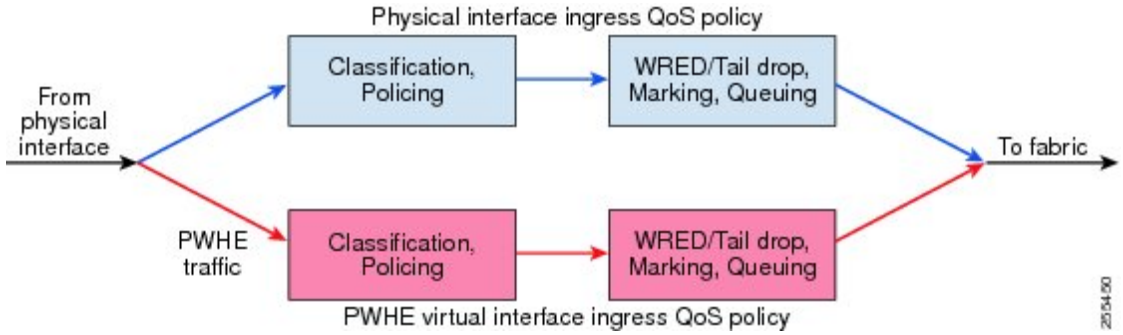
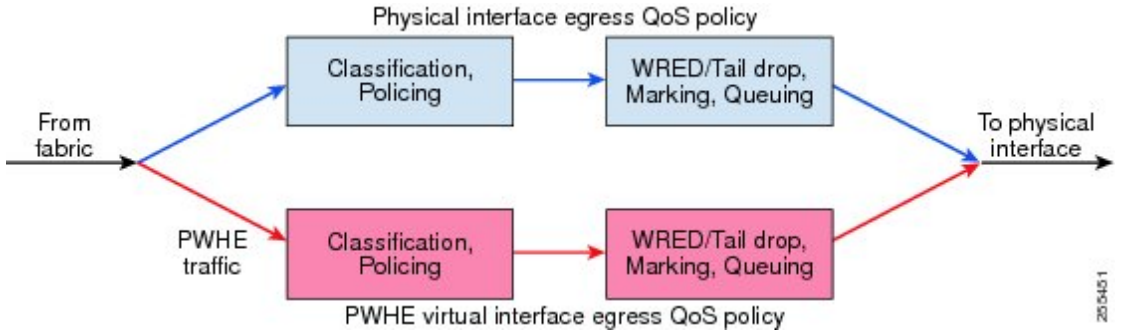


Figure 5: Egress QoS Policy Application for Physical and PWHE Traffic



The PWHE virtual interface is provisioned as a regular Layer 3 interface, and all QoS features that are supported on regular Layer 3 interfaces are supported on PWHE virtual interfaces. All ingress and egress QoS functions can be configured on the interface, including policing, shaping, queuing, and hierarchical policies. These details apply to header fields for traffic on PWHE interfaces:

- QoS classification and marking functions are based on the inner packet IP header and Layer 2 header fields. Marking of EXP bits for all imposed labels is supported in both ingress and egress QoS policies.
- If you configure Layer 2 overhead on the PWHE interface, the system adds this Layer 2 overhead to the packet length and then performs QoS functions.

The common MQC restrictions for QoS policies on Layer 3 interfaces apply to QoS policies for PWHE interfaces. In addition, these restrictions apply to QoS policies on PWHE interfaces:

- Configure the policy as a hierarchical policy with a single default class in the parent policy. Nonhierarchical policies are not supported.
- Configure either a police or shape command in the parent default class with an absolute rate. Percentage units are not allowed for police and shape commands in the parent default class.
- If you configure a police command without a shape command, you must configure a drop action within either the exceed action or the violate action.

QoS on Pseudowire Headend: Example

This example shows how to configure QoS policy on a PWHE interface. The **show** command at the end of the example displays the PWHE QoS statistics.

```
policy-map pw_parent_in
  class class-default
    police rate 100 mbps
    service-policy pw_child_in

policy-map pw_child_in
  class voip
    priority
    police rate 1 mbps
  class video
    police rate 10 mbps
    bandwidth remaining percent 20
  class data
    bandwidth remaining percent 70
  class class-default
    bandwidth remaining percent 10

policy-map pw_parent_out
  class class-default
    shape average 100 mbps
    service-policy pw_child_out

policy-map pw_child_out
  class voip
    priority
    police rate 1 mbps
  class video
    bandwidth 10 mbps
  class data
    bandwidth remaining percent 70
    random-detect discard-class 3 40 ms 50 ms
  class class-default
    random-detect discard-class 1 20 ms 30 ms

interface pw-ether 1
 service-policy input pw_parent_in
```

```
 service-policy output pw_parent_out

show policy-map interface pw-ether 1 member GigE 0/0/0/0 location 0/0/CPU0
```

Related Information

The information in this module focuses on the QoS implementation of features that are described in other technology guides. This table indicates the guides where you can find more information about these features.

Feature	Guide
Hierarchical VPLS QoS	*Cisco IOS XR MPLS Configuration Guide for the* Cisco CRS Router Cisco IOS XR *MPLS* Command Reference for the Cisco CRS Router
QinQ QoS	Cisco IOS XR Virtual Private Network Configuration Guide for the Cisco CRS Router Cisco IOS XR Virtual Private Network Command Reference for the Cisco CRS Router
QoS on Multicast VPN	Cisco IOS XR Multicast Configuration Guide for the Cisco CRS Router Cisco IOS XR Multicast Command Reference for the Cisco CRS Router
QoS with SRP	*Cisco IOS XR Interface and Hardware Component Configuration Guide* for the Cisco CRS Router *Cisco IOS XR Interface and Hardware Component Command Referenc*e for the Cisco CRS Router
VPLS QoS	*Cisco IOS XR MPLS Configuration Guide for the* Cisco CRS Router Cisco IOS XR *MPLS* Command Reference for the Cisco CRS Router