QoS Packet Marking

QoS Packet Marking refers to changing a field within a packet either at Layer 2 (802.1Q/p CoS, MPLS EXP) or Layer 3 (IP Precedence, DSCP and/or IP ECN). It also refers to preserving any classification decision that was reached previously.

- About, on page 1
- Configuration Examples, on page 6
- Verifying QoS Packet Marking, on page 9
- Network-Level Configuration Examples, on page 13
- Command Reference, on page 19

About

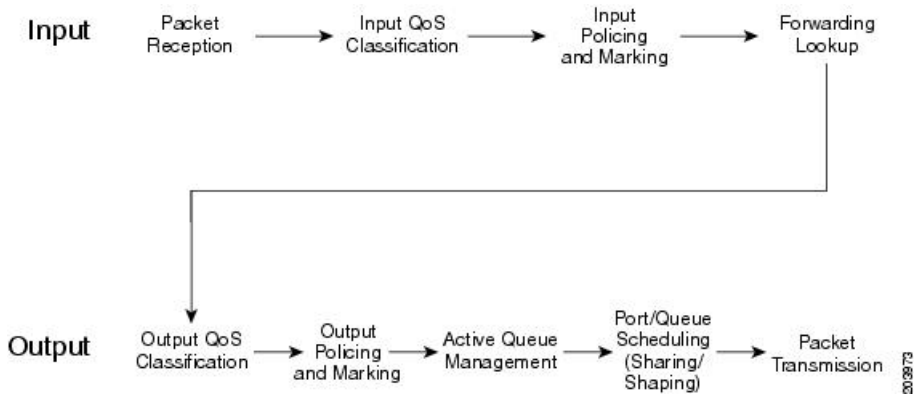
Marking Definition

Marking is similar conceptually to "service class" designation on an airplane ticket: first, business, or economy. This value reflects the level (quality) of service you should receive. Similarly, we mark a value in a packet to indicate the service class (henceforth termed service-class) for that packet as it traverses the network. By examining the marked value, network elements can decide how to treat your packet.

People in business-class may have used a variety of means to achieve that designation. They may have paid extra, used airmiles, or been lucky and booked at the normal rate when no other seat was available. Elsewhere, someone performed the complex task of classification - determining eligibility for a particular service-class then marked the ticket with a mere designation: first-class, business-class, or economy-class. Flight-attendants are unconcerned with how eligibility was determined; they simply look at the class marked on the ticket and provide that level of service.

This dynamic plays out in the networking world. One device may perform complex classification on the data in a flow, determining an appropriate service-class. Other network elements "trust" the value marked in packets they receive and provide service appropriate for that designation.

Figure 1: QoS Packet Processing



Within the context of QoS packet processing, marking occurs after classification and before queuing and is applicable on ingress or egress.

Typically, you would create a *trust boundary* at the edge of the network, then classify and mark packets on the edge device. Then, you would use that marked field for classification and determination of per-hop treatment throughout the network.

Note A trust boundary enables you to apply network-controlled marking on all packets as they enter the network and to remove or modify any non-default markings you did not apply.

Imagine that your system recognizes router ports with attached VoIP devices. You could mark the differentiated services code point (DSCP) value of voice packets as EF (at the edge of the network) and employ DSCP-based classification throughout the network to determine those packets that warrant low latency treatment.

Why Mark Packets

Reasons for marking packets include the following:

- Indicate the treatment you would like a packet to receive as it traverses the network.
- Perform complex classification once. By marking the service class, you can use simpler, less cpu-intensive classification elsewhere in the network.
- Perform classification at a point in the network where you have greater visibility into the flow. For example, if data is encrypted, you cannot perform complex classification such as determining the application carried within that flow. Instead, you could classify prior to encryption and mark a value in the unencrypted header that is visible to network elements along the path.

  As a packet traverses networks managed by different autonomous entities (e.g., the service provider network between two enterprise offices), you may need to re-mark if the markings to service-level designations are inconsistent across those networks.

  As a packet traverses different networking technologies the fields available to indicate service-class may differ. For example, you might carry service-class designation in the DSCP field of an IP packet but if this packet traverses an the multiprotocol label switching (MPLS) network only the MPLS experimental (EXP) field may be usable by network elements to determine service-class. As you enter that portion of the network, you may need to determine the appropriate marking of the MPLS EXP bits.

As a network operator you may contract to accept data from a user at a certain rate. Rather than dropping packets that exceed that rate, you can mark them as a lesser service-class.

Approaches to Marking Packets

You have two main approaches to marking packets: the **set** command and a policer marking action.

Note We only briefly touch upon "policing" actions within this chapter.

set Command

The simplest approach to marking packets on a router is to use the **set** command in a *policy-map* definition. (A policy-map is where you specify a QoS action for each class of traffic that you have defined).

You may decide to classify all RTP ports into a traffic class and mark each packet with AF41. If so, the policy-map may look something like this:

```
policy-map mark-rtp
  class rtp-traffic
    set dscp af41
```

Policer Marking Action

Recall that you can use a policer to drop packets within a traffic class above a defined rate. Alternatively, you could mark packets above that rate and allow them to receive a different per-hop treatment than packets below that rate.

For example, let's say that video traffic arrives at your router marked AF41. You may decide to consider user traffic up to 2 Mbps top *assured forwarding behavior* and to demote any traffic exceeding 2Mbps to AF42 (and considered *out of contract* - non-conforming).

The policy-map might appear as follows:

```
class-map video-traffic
  match dscp af41
!
policy-map enforce-contract
  class video-traffic
    police cir 2m conform-action transmit exceed-action set-dscp-transmit AF42
```

Scope of Marking Action

Similar to classification, marking cannot access every field within a data packet. For example, if an IP packet is encapsulated in multiprotocol label switching (MPLS), it cannot mark the DSCP within the IP header as that would require first de-capsulating from MPLS. However, you could mark the MPLS experimental (EXP) bits.

Note Only Layer 2 and outer Layer 3 headers are available for marking.

Multiple Set Statements

You can configure multiple marking rules within a single class (or policer action). This allows you to mark both Layer 2 and Layer 3 fields within the same packet, or if multiple traffic types are present in the same class, define marking values for each type.

For example consider the following egress policy attached to an Ethernet subinterface:

```
policy-map mark-rtp
  class rtp-traffic
    set cos 4
    set mpls exp topmost 4
    set dscp af41
```

If an MPLS packet were forwarded through this subinterface, the Layer 2 COS field and the EXP bits in the MPLS header would be marked. If an IP datagram were encapsulated in that packet, its DSCP value would remain unchanged. However, if an IP packet were forwarded through the subinterface, its Layer 2 COS value and Layer 3 DSCP values would be marked.

For details, refer to the command pages for set cos, on page 20, set mpls experimental topmost, on page 24, and set dscp, on page 21.

Marking Internal Designators

Cisco routers allow you to mark two internal values (qos-group and discard-class) that travel with the packet within the router but do not modify the packet's contents.

Typically, you mark these in an ingress policy and use them to classify to a traffic-class or WRED drop profile in an egress policy. For example, you may want to base your egress classification on a user's IP address but realize that encryption is configured and the user's IP address is invisible on an egress interface. You could classify their traffic on ingress (before encryption) and set an appropriate qos-group value. On egress, you could now classify based on the qos-group and choose the action accordingly.

Ingress vs. Egress Marking Actions

Certain marking values are only relevant to ingress or egress policies. For example, marking the ATM CLP bit or Frame Relay DE bit in an ingress policy is meaningless as they are discarded when the packet is decapsulated. Similarly, marking qos-group or discard-class in an egress policy is unproductive as these leave the packet unchanged and are discarded when we enqueue the packet for forwarding to the next hop.

Imposition Marking

Under special circumstances, you can mark a header field that has not yet been added to a packet (we term this behavior *imposition marking*).

The most common example of imposition marking is the application ofthe **set mpls experimental imposition** command - you can use it on an ingress interface where a packet may arrive containing an IP datagram and no multiprotocol label switching (MPLS) header. When and if the router encapsulates the datagram with a MPLS header, the EXP bits will be marked accordingly as specified by this command.

Application of the **set dscp tunnel** and **set precedence tunnel** commands (for IPv4 only) represent another example of imposition marking. If an egress policy is applied on a tunnel interface, no tunnel header exists when the policy executes. This means that any marking would apply to the original (eventually inner) IP

header. Using either command, you can mark the tunnel (outer) IP header and leave the original header unchanged.

The following table lists the tunnel types and encapsulation variants that support these commands:

Table 1: Supported DSCP and Precedence Tunnel Marking Configurations

Name	Outer Header (encapsulating)	Inner Header (payload)	Comments
GRE (4 over 4)	IPv4/GRE	IPv4	Supported
GRE (6 over 4)	IPv4/GRE	IPv6	Encapsulation not supported
GREv6 (4 over 6)	IPv6/GRE	IPv4	Encapsulation not supported
GREv6 (6 over 6)	IPv6/GRE	IPv6	Encapsulation not supported
IP-IP	IPv4	IPv4	Supported
IPv6-IP	IPv4	IPv6	Supported
IPv6 (4 over 6)	IPv6	IPv4	Encapsulation not supported
IPv6 (6 over 6)	IPv6	IPv6	Not supported
IPSEC (4 over 4)	IPv4/IPSEC	IPv4	Not supported
IPSEC (6 over 4)	IPv4/IPSEC	IPv6	Not supported
IPSECv6 (4 over 6)	IPv6/IPSEC	IPv4	Encapsulation not supported
IPSECv6 (6 over 6)	IPv6/IPSEC	IPv6	Not supported
mVPN(Multicast VPN)	IPv4/GRE	IPv4	Supported
DMVPN(dynamic multipoint VPN)			Supported
Multipoint GRE			Supported
MPLSoGREv4	IPv4/GRE	MPLS	Not supported
MPLSoGREv6	IPv6/GRE	MPLS	Not supported
L2TP	IPv4/L2TP	PPPoX	Not supported

When a new header is added (encapsulated), any QoS marking in the inner header is copied to the outer header. For example, when an IP datagram is encapsulated with an MPLS header, the default behavior is to copy the IP Precedence bits from the IP header to the MPLS EXP bits in the newly-imposed header.

Regarding header disposition, we typically do not copy any outer marking(s) to the inner header. For example, at the endpoint for a GRE tunnel, let's say that we receive a packet with different DSCP values in the outer and inner IP headers. When we remove the outer header we do not copy its DSCP value to the inner header.

For examples of configuring Imposition Marking, see Example 4: Configuring Tunnel Imposition Marking, on page 7 and Example 5: Using Tunnel Imposition Marking to Remark for an SP Network, on page 18.

For command details, please refer to set mpls experimental imposition, on page 24, set dscp tunnel, on page 22, and set precedence tunnel, on page 25.

Configuration Examples

Example 1: Configuring Ingress Marking

You can set up a trust boundary at the edge of a network (where marking is used) to indicate service-class for some traffic and to bleach all other traffic (see *** below). Enforcing a trust boundary at all ingress ports to the network allows you to maintain control of which applications are mapped to each service-class within the network:

```
policy-map ingress-marking
  class voice
    set dscp ef
  class video
    set dscp af41
  class scavenger
    set dscp cs1
  class class-default                                    ***
    set dscp 0                                           ***
!
interface gigabitethernet1/0/0
  Service-policy in ingress-marking
```

For details, refer to the page set dscp, on page 21.

Example 2: Configuring Egress Marking

If a different administrator controls a portion of a network path and uses a different DSCP to service-class mapping, egress marking may be necessary (e.g., within your enterprise, you classify 12 distinct classes of traffic as described in RFC4594). However, your service provider only provides a three-class model.

You may also need egress marking to indicate treatment for certain classes in a Layer 2 network (like Ethernet, frame-relay, or ATM switched networks):

```
policy-map egress-marking
  class scavenger
    set atm-clp
```

For command details, refer to the page set atm-clp, on page 20.

Example 3: Configuring MPLS EXP Imposition

With MPLS, a provider edge (PE) router encapsulates datagrams or frames with MPLS headers. Switching decisions within the core are based on the MPLS headers without visibility into the encapsulated data.

Consider a Layer 3 MPLS network where IPv4 datagrams are encapsulated in MPLS headers. On the customer edge (CE) facing interface we have visibility into the IPv4 header of the packet. On the core-facing interface, we have encapsulated datagrams with MPLS headers and we cannot see beyond those headers.

By default, we copy the IP precedence to the MPLS EXP bits. What if we want to override this behavior? We can't parse the IPv4 type of service byte on the core-facing interface. We can, however, parse the IP header on ingress and store the EXP value we plan to set when MPLS headers are added. Although MPLS headers

are absent when we execute the command, the router retrieves the instruction and marks the EXP bits on the egress interface:

```
policy-map mpls-exp-remark
  class voice
    set mpls experimental imposition 5
  class video
    set mpls experimental imposition 4
  class scavenger
    set mpls experimental imposition 0
!
interface gigabitethernet1/0/0
  policy-map input mpls-exp-remark
```

For command details, refer to the page set mpls experimental imposition, on page 24.

Example 4: Configuring Tunnel Imposition Marking

Conceptually, tunnel and MPLS EXP imposition marking are similar. We want to mark a value in a header that has not yet been added to the packet and with a Layer 3 tunneling technology like GRE or IPinIP, a Layer 3 datagram may be encapsulated with an outer IP header. (Refer to Imposition Marking, on page 4.)

Let's say that we have a DMVPN network where a branch location encrypts data and encapsulates it with a GRE header before sending it over a public IP network. An administrator may attach a policy-map to the tunnel interface to prioritize applications within that tunnel and may also need to mark the DSCP of the outer IP header to indicate service-class within the provider's network. When the policy is executed, the outer header has not yet been added and commands like **set dscp** or **set precedence** would mark the inner IP header.

To solve the problem, we use the **set dscp tunnel** and **set precedence tunnel** commands, as they allow you to set the value in an outer header that has not yet been added.

In the following example, voice and video traffic are classified and queued separately within the enterprise network. The service provider has a smaller number of service-classes and we have decided to put both voice and video into the priority class within the provider's network.

By marking the DSCP in the outer tunnel header we achieve this yet preserve original markings in the inner header:

```
policy-map mark-outer-gre-header
  class voice
    priority level1 percent 20
    set dscp tunnel ef
  class video
    priority level 2 percent 20
    set dscp tunnel ef
!
interface tunnel100
  service-policy out mark-outer-gre-header
```

For command details, refer to the page set dscp tunnel, on page 22.

Example 5: Configuring QoS-Group Marking

Occasionally, you may want to base egress queuing on ingress classification. For example, let's say you want more than 8 egress queues on a MPLS-enabled interface. Using egress classification, you are limited to MPLS EXP bits and therefore 8 classes. As a solution, you could perform classification on the ingress interface and

set a QoS group for packets that match that classification. QoS group has relevance only within the current router; it doesn't alter anything in the packet header. Instead, it's a value associated with the packet as it passes through the router.

In the following example we use Network Based Application Recognition (NBAR) classification on ingress and mark both telepresence and jabber video with qos-group 4. In the egress policy we classify based on the qos-group we marked on ingress (see "***"):

```
class-map telepresence-video
  match protocol telepresence-media
class-map jabber-video
  match protocol cisco-jabber-video
class-map egress-video-traffic                          ***
  match qos-group 4                                     ***
!
policy-map mark-qos-group
  class telepresence-video
    set qos-group 4
  class jabber-video
    set qos-group 4
!
policy-map egress-queuing
  class egress-video-traffic
    bandwidth remaining percent 50
!
interface gig 1/0/0
  service-policy in mark-qos-group
!
interface serial1/1/0
  service-policy out egress-queuing
```

For command details, refer to the page set qos-group, on page 25.

Example 6: Configuring Discard-Class Marking

In Example 5: Configuring QoS-Group Marking, on page 7, we marked both telepresence video and jabber video with qos-group 4 and placed both of these applications into the same egress queue.

What if we want to run Weighted Random Early Detection (WRED) on the egress queue and drop the jabber video first during congestion. Typically, WRED examines the precedence or DSCP value to determine drop thresholds for a flow. However, as indicated in Example 3: Configuring MPLS EXP Imposition, on page 6, we do not have visibility into the IP header. A solution is to mark a second internal value named discard-class. Then, we could use the qos-group to select the egress class (and queue) and the discard-class to select the WRED drop profile within that class.

```
class-map telepresence-video
  match protocol telepresence-media
class-map jabber-video
  match protocol cisco-jabber-video
class-map egress-video-traffic
  match qos-group 4
!
policy-map mark-qos-group
  class telepresence-video
    set qos-group 4
    set discard-class 1
  class jabber-video
    set qos-group 4
    set discard-class 2
```

```
!
policy-map egress-queuing
  class egress-video-traffic
    bandwidth remaining percent 50
    random-detect discard-class-based
    random-detect discard-class 1 24 40
    random-detect discard-class 2 22 30
!
interface gig 1/0/0
  service-policy in mark-qos-group
!
interface serial1/1/0
  service-policy out egress-queuing
```

For command details, refer to the page set discard-class, on page 21.

Verifying QoS Packet Marking

The **show policy-map interface** command is the primary means of verifying any QoS behavior on IOS XE platforms. Although the packet forwarding path (dataplane) is separated from the IOS instance (control plane), statistics are still reported through this well-known IOS command. This functionality is enabled by default.

This table describes the fields we employ in the following sections.

Table 2: show policy-map interface Field Descriptions (those useful for verifying marking)

Field	Description
Service-policy input	Denotes the name of the input service policy applied to the specified interface or VC
Class-map	Specifies the class of traffic being displayed. Output is displayed for each configured class in the policy. The choice for implementing class matches (e.g., match-all or match-any) can also appear adjacent to the traffic class
packets, bytes	Specifies the number of packets (shown in bytes) identified as belonging to the class of traffic being displayed
offered rate	Specifies the rate in bits per second of the packets entering the class
Match	Specifies the match criteria for the traffic class
QoS Set	Details the QoS marking actions configured for the particular class
Packets marked	If enabled, denotes the total number of packets marked for the particular class. If not enabled, you see "Marker statistics: Disabled."

Verifying with the show policy-map interface Command

The **show policy-map interface** command is the primary means of verifying any QoS behavior on IOS XE platforms. Ordinarily, knowing how many packets match a particular class ("class match statistics, " which is enabled by default) and what (if any) marking action is configured suffices to know how many packets were marked by that action.

Note You should understand how *class match statistics* (enabled by default) and *marking statistics* (disabled by default) differ. Typically, the former is sufficient. When a packet "hits" a class, you can assume it is marked. However, if you configure multiple, mutually exclusive marking values, and need to know how many packets were marked with each **set** command, you can enable marking statistics with all its caveats.

Here is an example of ingress marking with a policy attached to a physical interface. In this example, let's say that jabber-video is configured on ports 2000-3000:

```
class-map match-all jabber-video
  match ip rtp 2000 3000
!
policy-map mark-traffic
  class jabber-video
    set dscp af41

show policy-map int g1/0/0
GigabitEthernet1/0/0

  Service-policy input: mark-traffic

    Class-map: jabber-video (match-all)
      850 packets, 51000 bytes                                    note 1
      5 minute offered rate 2000 bps, drop rate 0000 bps
      Match: ip rtp 2000 3000
      QoS Set                                                     note 2
        dscp af41
          Marker statistics: Disabled

    Class-map: class-default (match-any)                          note 3
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any
```

Footnotes

note 1	Statistics for the class match
note 2	Packet matching section
note 3	Class-Default statistics section

Observe "`Marker statistics: Disabled`" in the output of ingress marking. If you are invoking multiple statistics and find the information provided in the previous output insufficient, you can enable "Packet Marker Statistics."

Verifying with QoS Packet Marking Statistics

Before you begin

Either

- Remove all policy-maps, issue the command, and re-attach all policy-maps.
- Issue the command, save the configuration, and reload the router.

Note Enabling QoS: Packet Marking Statistics may increase CPU utilization on a scaled configuration. Weigh the benefits of displaying statistics information against the increased CPU utilization for your system.

Enabling QoS Packet Marking Statistics

To enable Packet Marking Statistics, issue the **platform qos marker-statistics** command in configuration mode.

Displaying QoS Packet Marking Statistics

To display the packet statistics of all classes that are configured for all service policies either on the specified interface (or subinterface) or on a specific Permanent Virtual Circuit (PVC), use the **show policy-map interface** command.

When we singularly-configure marking in a policy-map, the output from an ASR 1000 Series Aggregation Services Router would appear as follows:

```
policy-map remark-af41
  class af41-traffic
    set dscp tunnel ef
```

Let's place this map on a tunnel interface with traffic marked af41 in the user's IP header and DSCP marked EF in the GRE IP header. The output of the **show policy-map interface** will appear as follows:

```
show policy-map interface tunnel1

  Service-policy output: remark-af41

    Class-map: af41-traffic (match-all)
      978 packets, 68460 bytes                                  note 1
      5 minute offered rate 2000 bps, drop rate 0000 bps
      Match:  dscp af41 (34)
      QoS Set                                                   note 2
        dscp tunnel ef
          Marker statistics: Disabled                           note 3

    Class-map: class-default (match-any)
      365 packets, 25550 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any
```

Footnotes

note 1	Displays the class match statistics (assume all "observed" packets are marked AF41).
note 2	Marking is the only action configured.

note 3	Per-set action statistics are disabled by default.

Now, if we enable marking statistics, output from the **show policy-map interface** command would appears as follows:

```
show policy-map interface tunnel1

  Service-policy output: remark-af41

    Class-map: af41-traffic (match-all)
      575 packets, 40250 bytes
      5 minute offered rate 1000 bps, drop rate 0000 bps
      Match:  dscp af41 (34)
      QoS Set
        dscp tunnel ef
          Packets marked 575                    note

    Class-map: class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any
```

Footnote

note	We have now enabled marking statistics but in this example the information is redundant.

For command details, refer to the page set dscp tunnel, on page 22.

Validating the Dataplane Configuration

To verify that the dataplane configuration reflects the IOS control plane configuration, use the **show platform hardware qfp active feature qos interface [input|output]** command, which engages only if issued before you attach any policy-map to an interface. So, you must do one of the following:

- Remove all policy-maps, issue the command and re-attach all policy-maps.
- Issue the command, save the configuration and reload the router.

In the following output, notice that we have configured the actions and set the values on the dataplane:

```
show platform hardware qfp active feature qos interface g1/0/0 input

Interface: GigabitEthernet1/0/0, QFP interface: 12
  Direction: Input
    Hierarchy level: 0
    Policy name: mark-traffic
      Class name: jabber-video, Policy name: mark-traffic
        QOS Set:
            dscp 34                                                    note
      Class name: class-default, Policy name: mark-traffic
```

Footnote

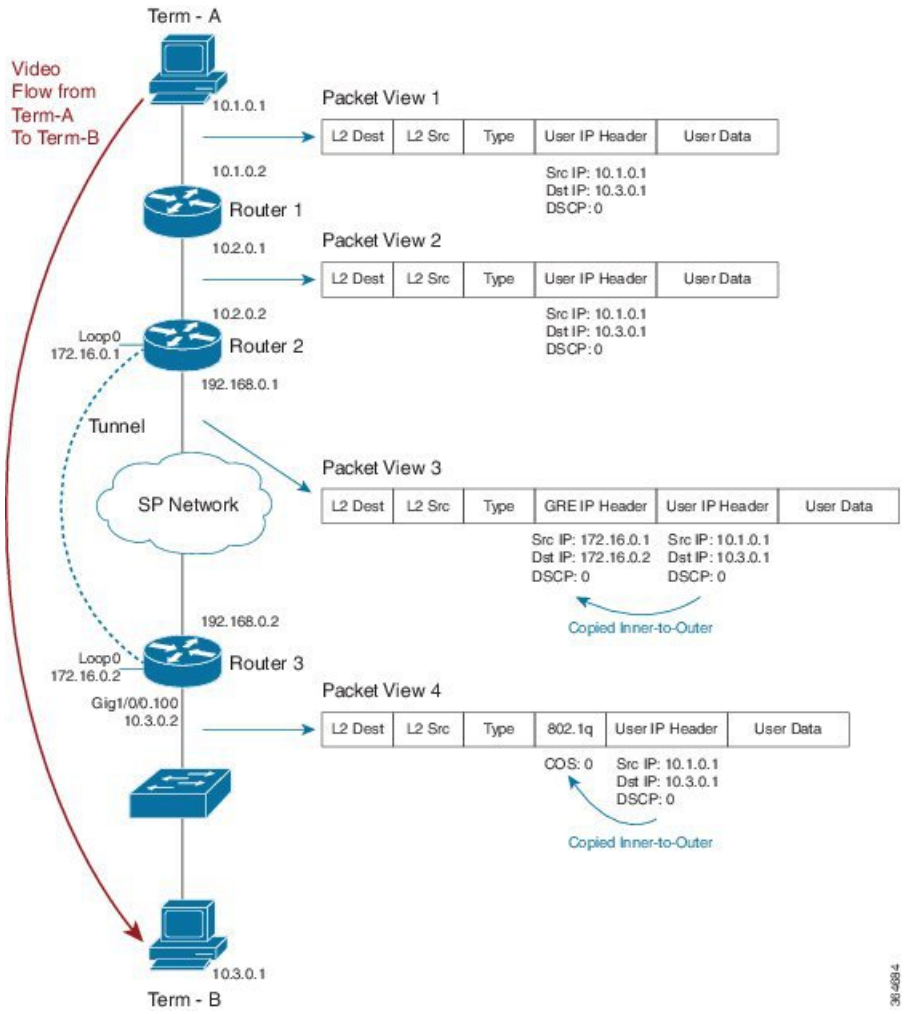
note	Dataplane is programmed to mark.

Network-Level Configuration Examples

In the scenarios that follow, a video-flow transits from Terminal-A to Terminal-B.

Example 1: Propagating Service-Class Information Throughout the Network

Figure 2: Propagating Service-Class Information Throughout the Network

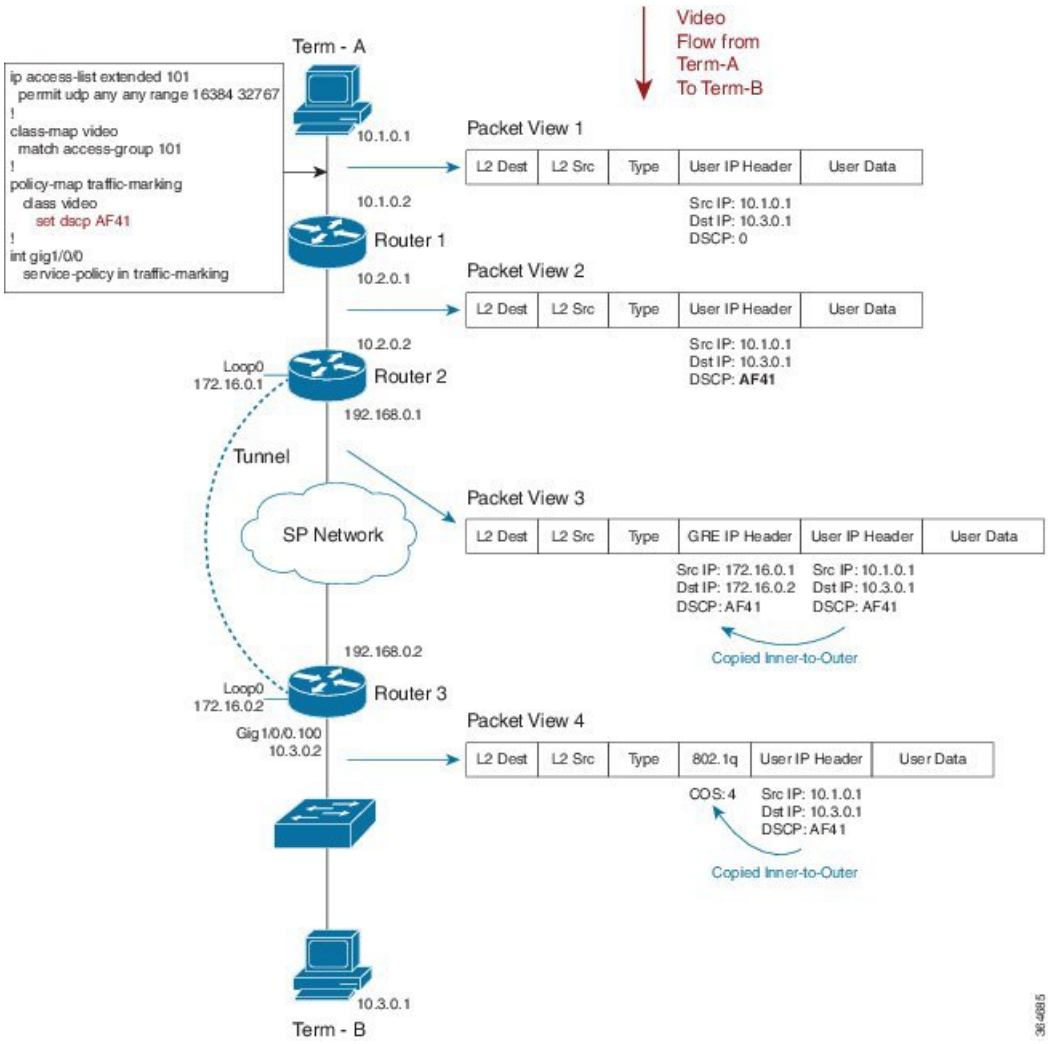


Imagine that an application marks the video stream with DSCP codepoint 0 (see Packet View 1). To cross the provider's network, we send the stream through a GRE tunnel (possibly encrypted). Packet View 3 shows that we have encapsulated the users' IP datagram in a GRE packet. Notice how the DSCP codepoint is copied by default to the imposed GRE header.

With the last hop at the final destination, Router 3 sends a VLAN tagged packet to a switch (see Packet View 4). Observe that the GRE header was stripped and a Dot1Q header was added due to the VLAN configuration. The precedence portion of the user's DSCP 0 (000 000) is copied by default to the COS bits of the VLAN header. The COS value set is 0 (000).

Example 2: Indicating Service-Class by Marking at the Network's Edge

Figure 3: Indicating Service-Class by Marking at the Network's Edge



In this example, we modify the default behavior by remarking the DSCP of users' traffic in an ingress policy as it enters Router 1. The following code shows how we do this:

```
ip access-list extended 101
 permit udp any any range 16384 32767
!
class-map video
 match access-group 101
!
policy-map traffic-marking
  class video
    set dscp AF41
!
int gig1/0/0
```

```
service-policy in traffic-marking
```

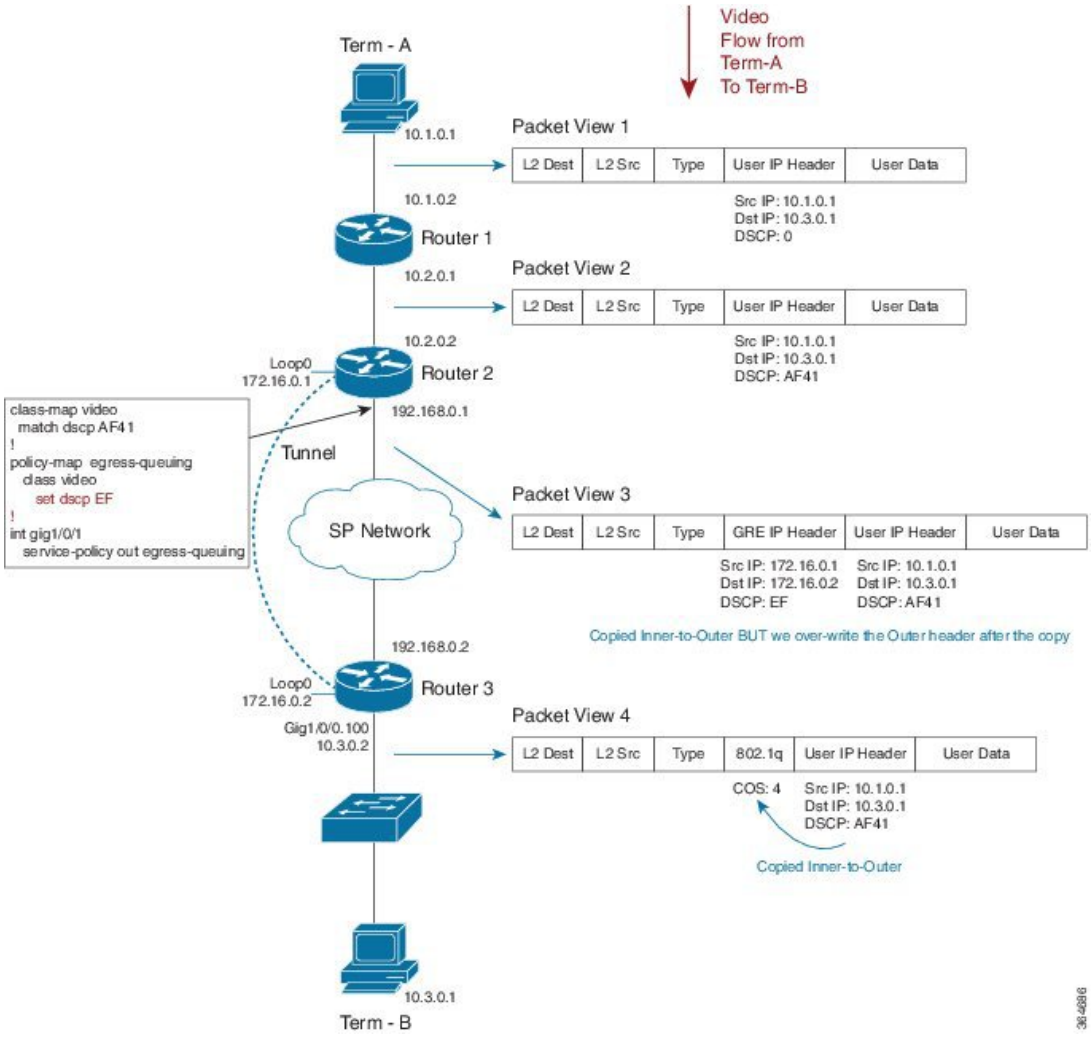
Let's say that we designate video traffic as DSCP AF41 throughout the network. When the packet reaches the GRE interface on egress, its DSCP value has already been changed to AF41 and its behavior matches that in Example 1. We send the stream through a GRE tunnel (possibly encrypted) as it traverses the providers network. Notice how the newly-marked DSCP codepoint (AF41) is copied by default to the imposed GRE header.

When we arrive at our destination, the router sends a VLAN-tagged packet to the last hop (a switch). The precedence portion of the users' DSCP value is copied by default into the COS bits of the VLAN header. As our DSCP is now AF41 (100 010), the COS value will be 4 (100).

For command details, refer to the command page set dscp, on page 21.

Example 3: Remarking Traffic to Match Service Provider Requirements

Figure 4: Remarking Traffic to Match Service Provider Requirements



In this example, we mark the DCSP value within the network while the service provider anticipates a different marking. The following code shows how we handle this:

```
class-map video
  match dscp AF41
!
policy-map  egress-queuing
  class video
    set dscp EF
!
int gig1/0/1
  service-policy out egress-queuing
```

We mark DCSP as AF41 for video within out network while the service provider expects video packets to be marked EF. On the egress Gig interface of Router 2, we add a policy that contains queuing commands (recall that we are only focusing on the marking portion of the configuration in this example).

When the packet reaches the egress physical interface it already has the GRE header imposed and we copy the DSCP value of AF41 from the inner encapsulated datagram. The policy on the physical interface changes the DSCP value in the outer GRE header only.

Note Notice how the inner-user datagram IP header is unchanged.

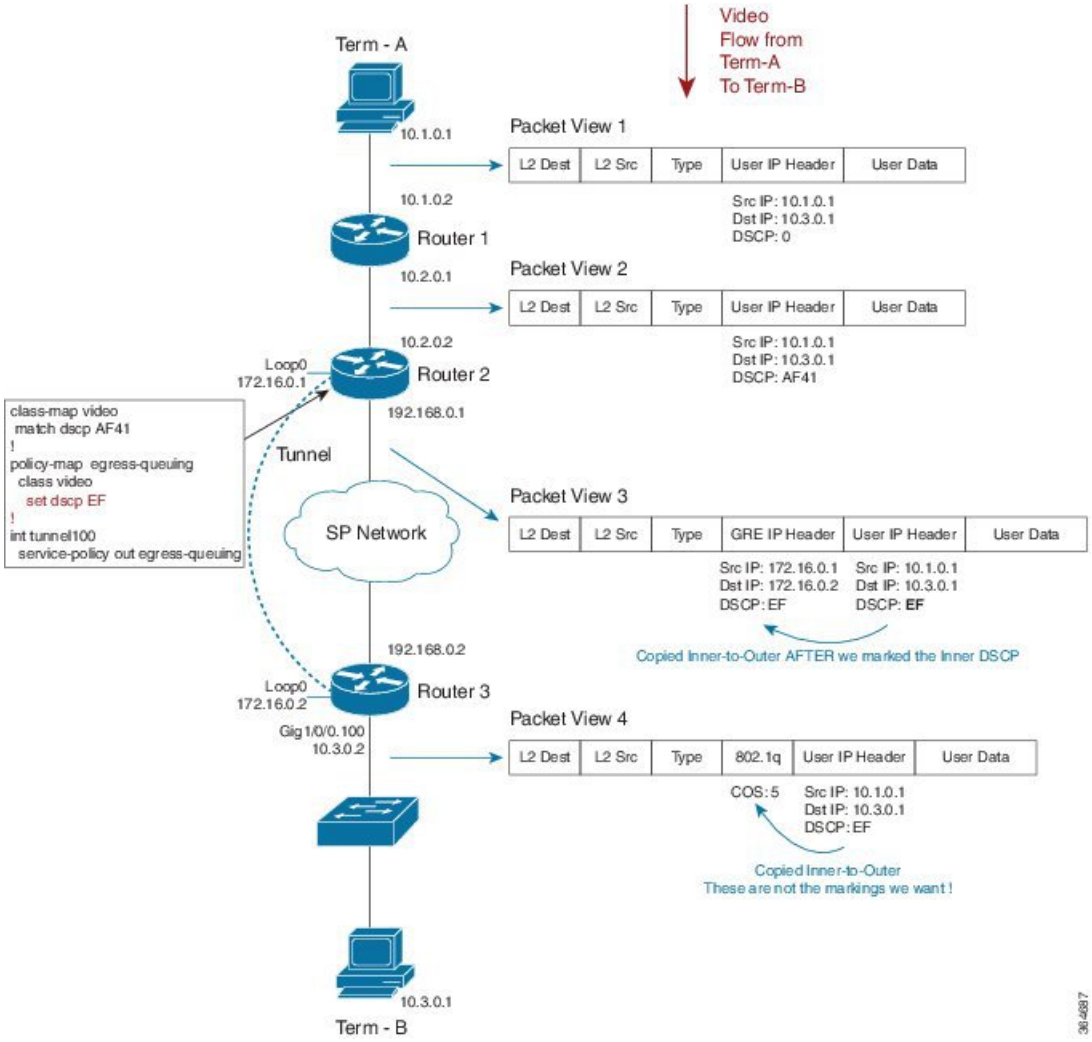
When we reach Router 3 and exit the tunnel, the tunnel GRE header is stripped. Henceforth, only the user datagram IP header is visible, still preserving the AF41 value we marked on ingress to the network.

As in previous examples, the router sends a VLAN-tagged packet to the last hop (a switch). By default, the precedence portion of the User IP Header's DSCP value is copied into the COS bits of the VLAN header (802.1q). As the DSCP value is currently af41 (100 010), the COS value will be 4 (100).

For command details, refer to the page set dscp, on page 21.

Example 4: Remarking on a Tunnel Interface for an SP Network - Potential Gotcha

Figure 5: Remarking on a Tunnel Interface for an SP Network - Potential Gotcha



In this example, we place the QoS policy on the tunnel interface of Router 1 rather than on the physical interface. (There are many advantages to configuring queuing per tunnel rather than as an aggregate policy on the physical interface.) The following code shows how we do this:

```
class-map video
  match dscp AF41
!
policy-map  egress-queuing
  class video
    set dscp EF
!
int tunnel100
```

```
service-policy out egress-queuing
```

We focus solely on the marking portion of the policy. The key point is that marking on the tunnel interface is performed before the tunnel headers are added.

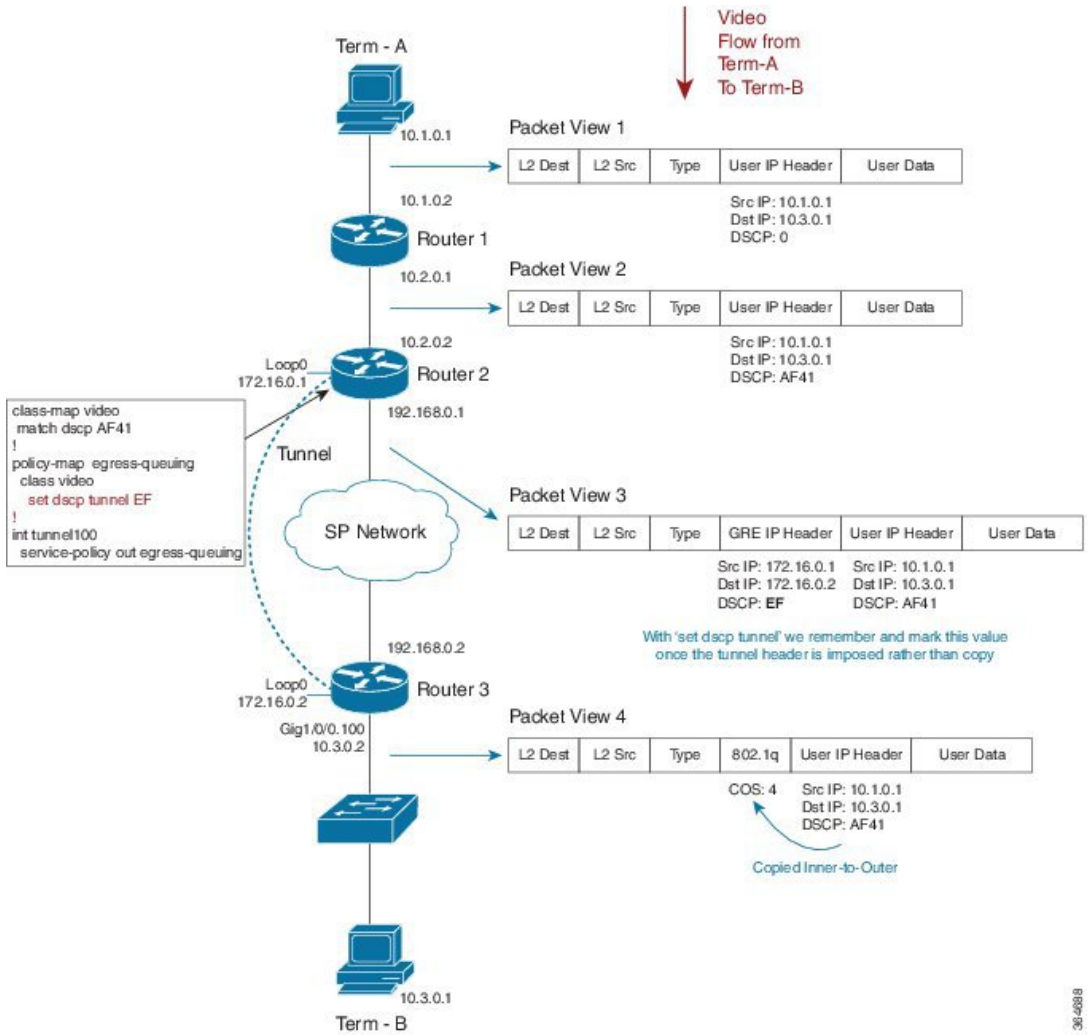
Notice how our policy has over-written the DSCP in the user datagram IP header. Because this happened before GRE encapsulation, we copy the newly-marked value to the outer header.

When we reach Router 3 and exit the tunnel the tunnel GRE header is stripped. Because we marked the user datagram header, the new value propagates through the rest of the network. This is not the behavior we wanted.

For command details, refer to the page for set dscp, on page 21.

Example 5: Using Tunnel Imposition Marking to Remark for an SP Network

Figure 6: Using Tunnel Imposition Marking to Remark for an SP Network



In this example, we use the **set dscp tunnel** *dscp-value* command to alter only the tunnel IP Header:

```
class-map video
  match dscp AF41
!
policy-map  egress-queuing
  class video
    set dscp tunnel EF
!
int tunnel100
  service-policy out egress-queuing
```

We have a QoS policy on the tunnel interface of Router 2 and we have used the **set dscp tunnel** command rather than **set dscp** command.

We have yet to impose the GRE header. The **set dscp tunnel** command dictates that we remember the DSCP value; during encapsulation we use this value instead of copying "inner to outer." Observe that the DSCP value in the users IP datagram header is unchanged. The **set dscp tunnel** command will alter only the tunnel IP header.

For command details, refer to the page for set dscp tunnel, on page 22.

Command Reference

platform qos marker-statistics

To enable individual statistics collection for each marking action in every policy configured on the router, use the **platform qos marker-statistics** command in global configuration mode. To disable packet marking statistics, use the **no** form of this command.

[no] platform qos marker-statistics

Syntax Description

This command has no arguments or keywords.

Command Default

Disabled (no packet marking statistics are displayed). The network operator relies on class match statistics.

Command Modes

policy-map (config-pmap)

Usage Guidelines

This command executes only if issued before any policy-map is attached to an interface. So, you must do one of the following:

- Remove all policy-maps, issue the command and re-attach all policy-maps.
- Issue the command, save the configuration and reload the router.

Note Enabling packet marking statistics may increase CPU utilization on a scaled configuration. So, weigh the benefits of the statistics information against the increased CPU utilization for your system.

set atm-clp

To set the ATM cell loss priority (CLP) bit, use the **set atm-clp** command in policy-map class configuration mode. To disable this setting, use the **no** form of this command.

[no] set atm-clp

Syntax Description

This command has no arguments or keywords.

Command Default

The ATM CLP bit is not set.

Command Modes

policy-map (config-pmap)

Usage Guidelines

On ATM interfaces, you can use the **set atm-clp** command in an outbound policy to set the ATM-CLP bit in ATM cell headers to 1.

This command is supported for ATM, PPPoA, PPPoEoA and L2TPv3 encapsulations. It is not supported if the policy is attached to a tunnel rather than directly to the VC.

You cannot attach a policy-map containing ATM set cell loss priority (CLP) bit QoS to PPP over X (PPPoX) sessions. The map is accepted only if you do not specify the **set atm-clp** command.

For an example using the **set atm-clp** command to configure egress marking, please refer to Example 2: Configuring Egress Marking, on page 6.

set cos

To set the Layer 2 class of service (CoS) value of an outgoing packet, use the **set cos** command in policy-map class configuration mode. To disable this setting, use the **no** form of this command.

[no] set cos *cos-value*

Syntax Description

cos-value	Specifies the IEEE 802.1Q CoS value of an outgoing packet ranging from 0 to 7

Command Default

Either IP Precedence or MPLS EXP bits are copied from the encapsulated datagram.

Command Modes

policy-map (config-pmap)

Usage Guidelines

You can use the **set cos** command to propagate service-class information to a Layer 2 switched network. Although a Layer 2 switch may not be able to parse embedded Layer 3 information (such as DSCP), it might be able to provide differentiated service based on CoS value. Switches can leverage Layer 2 header information, including the marking of a CoS value.

Traditionally the **set cos** command had meaning only in service policies that are attached in the egress direction of an interface because routers discard Layer 2 information from received frames. With the introduction of features like EoMPLS and EVC, the setting of CoS on ingress has meaning, such that you can preserve Layer 2 information throughout the routed network.

set cos-inner

To set the Layer 2 CoS value in the inner VLAN tag of a QinQ packet, use the **set cos-inner** command in policy-map class configuration mode. To disable this setting, use the **no** form of this command.

[no] set cos-inner *cos-value*

Syntax Description

cos-value	Specifies a IEEE 802.1q CoS value ranging from 0-7

Command Default

Either IP Precedence or MPLS EXP bits are copied from the encapsulated datagram.

Command Modes

policy-map (config-pmap)

Usage Guidelines

Traditionally, because routers discard Layer 2 information from received frames, the **set cos-inner** command had meaning only in service policies that are attached in the egress direction of an interface. With the introduction of features like EoMPLS and EVC, the setting of CoS on ingress has significance as you can preserve Layer 2 information throughout the routed network.

set discard-class

To set the QoS discard class for a packet, use the **set discard-class** command in policy-map configuration mode. To disable this setting, use the **no** form of this command.

[no] set discard-class *discard-class-value*

Syntax Description

discard-class-value	Specifies a Discard Class value ranging from 0 to 7

Command Default

The discard-class value associated with a packet is set to 0.

Command Modes

policy-map (config-pmap)

Usage Guidelines

The **set discard-class** command allows you to associate a discard class value with a packet while processed by the router. Setting this value leaves the packet unchanged.

You can use the discard class and discard-class based WRED in egress policies to control which packets are dropped during congestion.

set dscp

To set the DSCP value in the IP header, use the **set dscp** command in policy-map class configuration mode. To disable this setting, use the **no** form of this command.

[no] set dscp *dscp-value*

Syntax Description

dscp-value	Sets the DSCP value in an IP header ranging from 0 to 63. You can specify the value numerically or by using it's well known DiffServe name (e.g., EF)

Command Default

Retain the existing DSCP value in the received packet.

Command Modes

policy-map (config-pmap)

Usage Guidelines

The command may be used in ingress or egress policies.

You can use the DSCP value to indicate the QoS treatment a packet should receive as it traverses a network.

Note The differentiated services architecture using DSCP supersedes use of precedence.

This command marks packets where the outermost Layer 3 header is either IPv4 or IPv6.

If issued in an egress policy-map, this command will not alter the class or queue selection but might influence the WRED drop profile selection.

The **set dscp** and **set ip dscp** commands behave identically, marking both IPv4 and IPv6 packets.

Note This differs from the process of classification wherein the **match ip dscp** command classifies only IPv4 packets while the **match dscp** command classifies both IPv4 and IPv6 packets.

set dscp tunnel

To set the DSCP value in a tunnel header that has not yet been added to a packet, use the **set dscp tunnel** command in policy-map class configuration mode. To disable this setting, use the **no** form of this command.

[no] set dscp tunnel *dscp-value*

Syntax Description

dscp-value	Specifies the DSCP value in a tunnel header ranging from 0 to 63. You can either specify the value numerically or use its well known DiffServe name (e.g. EF).

Command Default

DSCP value from an encapsulated datagram is copied to the newly-imposed tunnel header.

Command Modes

policy-map (config-pmap)

Usage Guidelines

This command only makes sense before a tunnel header is added.

Note You can use this command in either an ingress or egress policy that is attached to a tunnel interface. However, if the latter is attached, the command has no meaning because all headers would be added when the policy is evaluated.

On the Cisco ASR Series Aggregation Services Router, the **set dscp tunnel** command is supported for IPv4 only. See Imposition Marking, on page 4 for a table that lists the supported DSCP tunnel marking configurations.

For an example using this command to encapsulate a Layer 3 datagram with an outer IP header, please refer to Example 4: Configuring Tunnel Imposition Marking, on page 7.

set fr-de

To set the frame-relay (FR) discard eligible (DE) bit, use the **set fr-de** command in policy-map class configuration mode. To disable the setting, use the **no** form of this command.

[no] set fr-de

Syntax Description This command has no arguments or keywords.

Command Default The DE bit is not set when datagrams are encapsulated with frame relay.

Usage Guidelines On serial interfaces configured with Frame Relay encapsulation, you can use the **set fr-de** command in an outbound policy to set the Discard Eligible bit in the Frame Relay header to 1.

set ip dscp

To preserve backwards compatibility, we support two command variants that perform identical functions: **set ip dscp** and **set dscp**. You can use either to mark the DSCP value in the IP header. Please refer to the **set dscp** command page (set dscp, on page 21) for more information.

set ip dscp tunnel

To preserve backwards compatibility, we support two command variants that perform identical functions: **set ip dscp tunnel** and **set dscp tunnel**. Please refer to the **set dscp tunnel** command page (set dscp tunnel, on page 22) for details.

set ip precedence

To preserve backwards compatibility, we support two command variants that perform identical functions: **set ip precedence** and **set precedence**. You can use either to mark the precedence value in the IP header. Please refer to the **set precedence** command page (set precedence, on page 24) for more information.

set ip precedence tunnel

To preserve backwards compatibility, we support two command variants that perform identical functions: **set ip precedence tunnel** and **set precedence tunnel**. Please refer to the **set precedence tunnel** command page (set precedence tunnel, on page 25) for more information.

set mpls experimental imposition

To set the value of the MPLS EXP field on all imposed label entries, use the **set mpls experimental imposition** command in policy-map class configuration mode. To disable this setting, use the **no** form of this command.

[no] set mpls experimental imposition *mpls-exp-value*

Syntax Description

mpls-exp-value	Specifies the MPLS EXP value, which ranges from 0 to 7

Command Default

MPLS value is copied from the appropriate field (usually precedence) in the encapsulated packet.

Command Modes

policy-map (config-pmap)

Usage Guidelines

The **set mpls experimental imposition** command is supported only on input interfaces. Use this command during label imposition to set the MPLS EXP field on all imposed label entries.

For an example of using this command to set the EXP bits in an MPLS header that we use to encapsulate the datagram or frame, please refer to Example 3: Configuring MPLS EXP Imposition, on page 6.

set mpls experimental topmost

To set the MPLS EXP field value in the topmost label, use the **set mpls experimental topmost** command in policy-map class configuration mode. To disable this setting, use the **no** form of this command.

[no]set mpls experimental topmost *mpls-exp-value*

Syntax Description

mpls-exp-value	Specifies the MPLS EXP value ranging from 0 to 7

Command Default

The MPLS EXP value is either copied from the innermost header on encapsulation or remains unchanged.

Command Modes

policy-map (config-pmap)

Usage Guidelines

This command marks packets provided the outermost Layer 3 header is an MPLS label when the command is evaluated.

This command sets the MPLS EXP value in the topmost label only. If multiple labels exist in a stack, the MPLS EXP value in labels other than the topmost remain unchanged.

set precedence

To set the IP Precedence value in the packet header, use the **set precedence** command in policy-map class configuration mode. To disable this setting, use the **no** form of this command.

[no] set precedence *precedence-value*

Syntax Description

precedence-value	Sets the precedence bit in the packet header, which ranges from 0 to 7

Command Default

Retain the precedence value in the received packet.

Command Modes

policy-map (config-pmap)

Usage Guidelines

The command may be used in ingress or egress policies. However, if you issue the command in an egress policy-map, it will not alter the class or queue selection but it may influence the WRED drop profile selection.

By setting a precedence value, you indicate the QoS treatment a packet should receive as it traverses a network.

Note The differentiated services architecture using DSCP largely supersedes the use of precedence.

The **set precedence** and **set ip precedence** commands behave identically, marking packets where the outermost Layer 3 header is IPv4 or IPv6. In contrast, the **match ip precedence** command classifies only IPv4 packets while the **match precedence** command classifies both IPv4 and IPv6.

set precedence tunnel

To set the IP precedence value in a tunnel header that has not yet been added to a packet, use the **set precedence tunnel** command in policy-map class configuration mode. To disable this setting, use the **no** form of this command.

[no] set precedence tunnel *precedence-value*

Syntax Description

precedence-value	Sets the precedence bit in the tunnel header ranging from 0 to 7

Command Default

DSCP (and the precedence portion) are copied from the encapsulated to the newly-imposed header.

Command Modes

policy-map (config-pmap)

Usage Guidelines

On the Cisco ASR Series Aggregation Services Router, the **set precedence tunnel** command is supported for IPv4 only. See Imposition Marking, on page 4 for a table that lists the supported DSCP tunnel marking configurations.

set qos-group

To set the QoS group identifier (ID) for a packet, use the **set qos-group** command in policy-map class configuration mode. To disable this setting, use the **no** form of this command.

[no] set qos-group *group-id*

Syntax Description

group-id	Specifies a QoS group ID ranging from 0 to 99

Command Default

QoS group-id defaults to 0.

Command Modes

policy-map (config-pmap)

Usage Guidelines

The **set qos-group** command allows you to associate a group ID with a packet as it is processed by the router.

You can use the group ID in egress policies to classify packets to service-classes. Historically, this action had no meaning because we chose the service-class before egress marking occured. With color-aware policing, however, setting the QoS group ID in an egress policy can have meaning.