Chapter 39

Software Quality of Service (QoS)

Introducing QoS 39-4
Stages 39-4
Interfaces for Software QoS 39-5
When to Use Software QoS 39-5
Separate Traffic—Separate Needs 39-6
Applying QoS in a Network 39-6
Local Level 39-6
Domain Level: DiffServ, TOS and 802.1p Priority 39-6
The Hierarchy 39-9
Traffic Class Trees 39-10
Traffic Classes 39-12
Policies 39-13
Order of Classifier Matching 39-13
Dynamic Application Recognition for Voice and Video 39-14
Processing Points 39-15
Ingress QoS 39-17
Egress QoS 39-17
Tunnel QoS 39-17
Software QoS Stages 39-18
Packet Flow 39-18
Classification: Identifying and Sorting Traffic 39-20
Bandwidth Class 39-20
Premarking: Labelling Packets Before Metering 39-20
Metering: Bandwidth Conformance 39-21
Packet Queuing 39-24
RED Curves 39-24
Dequeuing 39-26
Queue Scheduling 39-29
Re-Marking 39-31
Virtual Bandwidth 39-31
Configuring a Software QoS Hierarchy 39-32
The Total Software QoS Solution 39-32
Default Traffic Class 39-35
Configuring QoS Stages 39-36
Premarking 39-36
Metering 39-37
RED 39-40
Re-Marking 39-41
Queue Scheduling 39-43
Configuring DAR for Voice and Video Traffic 39-47
Configuring Software QoS on Specific Interfaces 39-49

PPP and PPPoE 39-49
Frame Relay 39-52
The Switch Instance 39-53
Configuring Software QoS on Tunnels 39-55
VPN 39-55
6 to 4 39-56
Generic Router Encapsulation (GRE) 39-57
Interaction with Other Modules 39-58
Network Address Translation (NAT) 39-58
Resource Reservation Protocol (RSVP) 39-59
Priority Filters 39-59
Policy Filters 39-59
Bandwidth Limiting on Ethernet Interfaces 39-59
Counters 39-60
Debugging 39-61
Configuration Examples 39-64
1: Guaranteeing VoIP Traffic 39-65
2: Guaranteeing VoIP Traffic using DAR 39-68
3: Guaranteeing VoIP Traffic While Maintaining File Server Traffic 39-71
4: Guaranteeing VoIP Traffic over a VPN Tunnel 39-74
5: VoIP, Critical Database, and File Server Traffic 39-79
6: Multiple Applications over Frame Relay 39-82
Command Reference 39-87
add sqos interface dar 39-87
add sqos policy trafficclass 39-88
add sqos trafficclass classifier 39-89
add sqos trafficclass dar 39-90
add sqos trafficclass subclass 39-91
create sqos dar 39-92
create sqos dscpmap 39-94
create sqos meter 39-95
create sqos policy 39-98
create sqos red 39-101
create sqos trafficclass 39-103
delete sqos interface dar 39-108
delete sqos policy trafficclass 39-109
delete sqos trafficclass classifier 39-110
delete sqos trafficclass dar 39-111
delete sqos trafficclass subclass 39-112
destroy sqos dar 39-113
destroy sqos dscpmap 39-113
destroy sqos meter 39-114
destroy sqos policy 39-114
destroy sqos red 39-115
destroy sqos trafficclass 39-115
disable sqos 39-116
disable sqos debug 39-116
enable sqos 39-117
enable sqos debug 39-117
purge sqos 39-118
reset sqos counters 39-119
set sqos dar 39-121
set sqos dscpmap 39-123
set sqos interface 39-125
set sqos meter 39-127
set sqos policy 39-130
set sqos red 39-133
set sqos trafficclass 39-135
show sqos 39-139

show sqos counters 39-140
show sqos dar 39-146
show sqos dscpmap 39-148
show sqos interface 39-150
show sqos meter 39-152
show sqos policy 39-154
show sqos red 39-158
show sqos trafficclass 39-161

Introducing QoS

Quality of Service refers to the ability to intelligently manage network traffic to allow stable and predictable end-to-end network performance. It helps you achieve either or both of the following fundamental aims:

- providing sensitive traffic with the network resources it needs even when the network is congested—including traffic that is sensitive to delay, jitter and packet loss. To achieve this, the router delays or drops non-sensitive traffic, or traffic that exceeds the bandwidth to which it is entitled.
- guaranteeing and/or limiting the resources available to a particular customer or traffic type.

The concept of QoS is a departure from the "best effort" approach to data networking, which treats all traffic on the Internet or within a LAN the same. Without QoS, the router is equally likely to drop every different traffic type when a link becomes oversubscribed. With QoS, the router can give preferential treatment to a subset of traffic. It does this by sorting packets according to criteria you set, measuring the bandwidth the packets are using, assigning them to an appropriate queue or dropping them, and then scheduling the transfer of packets from queues onto the wire. The router can also mark packets so that downstream routers or switches know how to process them, and act on the marking from an upstream router or switch.

Quality of service mechanisms allow:

- traditional voice and data carriers to effectively compete against aggressive competition from wireless, satellite, and cable providers through the ability to integrate and deliver voice, video, and data services over a single network
- network service providers to sell different levels of service to customers, based on what customers require, and be confident in their ability to guarantee the reliable delivery of these services
- enterprise and educational organisations to actively manage and provide many services across one network, for example live video streaming and standard data services, with preferential treatment given for mission-critical traffic
- network administrators to manage network congestion as network traffic levels increase and time-critical applications, such as streaming media, become more widely in demand by customers and organisations

Stages

Configuring Quality of Service involves separate stages that are described in different chapters of the Software Reference.

The two stages are:

1. Classifying traffic into flows, according to a wide range of criteria.

   Classification is performed by the router's packet classifier and is not described in this chapter, but in Chapter 38, Generic Packet Classifier.

2. Acting on these traffic flows. The approaches, methods, and commands for this are described in this chapter.

Interfaces for Software QoS

Software QoS refers to QoS functions that the software in the router's CPU performs, rather than by a switching ASIC. It can apply to traffic over most WAN interfaces, plus IPv6, IPsec, and GRE tunnels. The following table lists interfaces for Software QoS.

Interface, tunnel or policy type	Example
ETH ports (but not individual switch ports)	eth0
PPP interfaces	ppp0
Frame Relay interfaces	fr0
ATM interfaces	atm0.0
the switch instance (all switch ports as a unit)	swi0
6 to 4 tunnels	virt0
IPsec tunnels	ipsec-CentralOffice
GRE tunnels	gre1

For the Layer 2 interfaces (eth, PPP, FR and ATM) most software QoS processing occurs as part of sending the traffic out. You can also use QoS to drop or prioritise traffic as soon as it arrives at the router, to reduce the probability of packet loss at a congested ingress interface.

When to Use Software QoS

Software QoS can benefit your network if:

- Traffic rates over an interface are too high, and therefore:
  - high-priority traffic is being dropped
  - delay-sensitive traffic, such as VoIP traffic, is being delayed
  - jitter-sensitive traffic, such as streaming video, is experiencing variable gaps between packets.
- Network congestion is occurring at other devices in your network which have no or minimal QoS capability. You can slow traffic down or mark it with priority information as it leaves the router so that downstream devices are not overwhelmed.
- You want to control the bandwidth available to particular users, depending on their required level of service.

The quality improvements are greatest for slower interface types.

Software QoS is not beneficial and may even reduce overall performance if:

- The network is not congested. Prioritising traffic is only useful if the target traffic is otherwise dropped or unacceptably delayed.
- All traffic has equal priority or is equally sensitive to delay, jitter, and loss.
- Your network is so congested that not all target traffic can be processed adequately even with QoS. In this case, the only solution is to upgrade the network infrastructure.

Separate Traffic—Separate Needs

Separate traffic has separate needs. Deciding which type of service is appropriate to deploy in the network depends on the service needs. For example, interactive voice and video requires high priority, low latency, low jitter, and controlled bandwidth. The following table describes different types of service and their requirements.

This service...	Requires...
Interactive voice and video conferencing	High priority, low latency, low jitter, controlled bandwidth
Client-server applications	High priority, low latency, low loss
Streaming audio and video	Medium priority, low jitter
Network control traffic	High priority, controlled bandwidth
Circuit emulation	Guaranteed, but controlled bandwidth
Everything else—Best effort	Low priority, long queues

Applying QoS in a Network

The major scenarios for applying QoS in a network are:

- **Local Level**
- **Domain Level: DiffServ, TOS and 802.1p Priority**

Local Level

You can configure QoS on the router as a local "action" which only affects only the flow of data from the router. This approach is suitable for many networks, for example those with a single router, or a single bottleneck, and which do not have to conform to a Service Level Agreement (SLA).

The QoS solution takes immediate action on the traffic passing through the router, directly affecting the flow of data. The profile of the traffic exiting the router reflects the QoS policy but the transmitted packets do not carry any QoS information to be used by the next-hop device.

Domain Level: DiffServ, TOS and 802.1p Priority

Alternatively, you can deploy QoS across an entire domain. The domain's QoS schema is designed for the whole domain so that the per-hop behaviour within the domain is consistent with the requirements that the schema serves. Packets that enter at the domain's edge may not carry any QoS information, but the edge device places such information into the packets before transmitting them to the next node in the domain. Thus, QoS information is preserved between nodes within the domain and the nodes treat the packets accordingly.

The following options are available for preserving QoS information:

- the 802.1p priority field within the VLAN tag of tagged Ethernet packets (see Figure 39-1)
- the IP Type of Service (TOS) field
- the Differentiated Services (DiffServ) Code Point (DSCP).

TOS and DSCP are mutually exclusive, and TOS is not available on IPv6 packets.

DiffServ is a method of dividing IP traffic into classes of service without requiring that every in a network remember detailed information about traffic flows. Routers within a DiffServ domain process traffic on the basis of the DSCP (DiffServ Code Point) value in the IP header's Differentiated Services (DS) field[1] (see Figure 39-2).

Figure 39-1: VLAN tag field in Ethernet packets

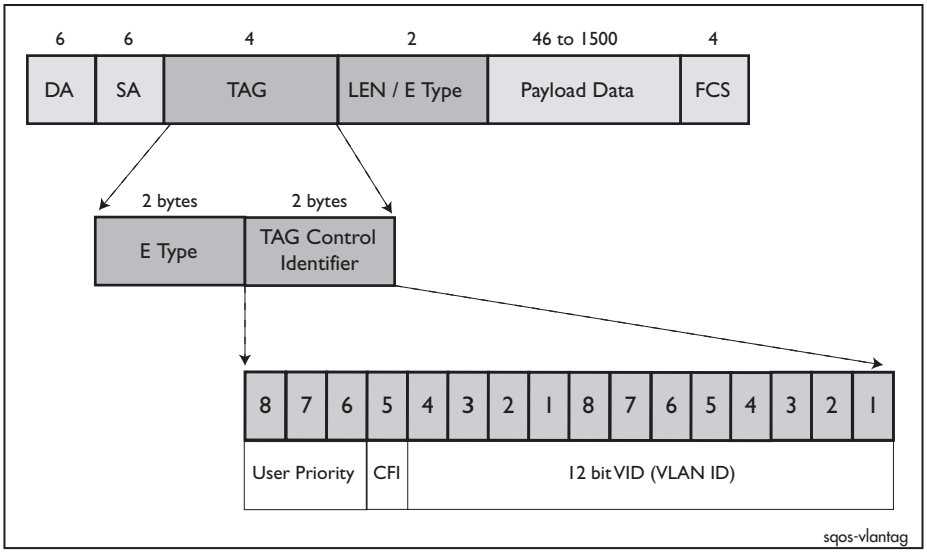


Figure 39-2: DSCP bits of the DS field in the IPv4 header.

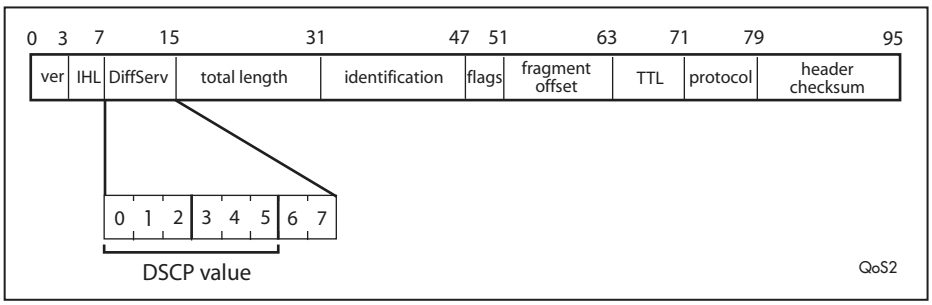


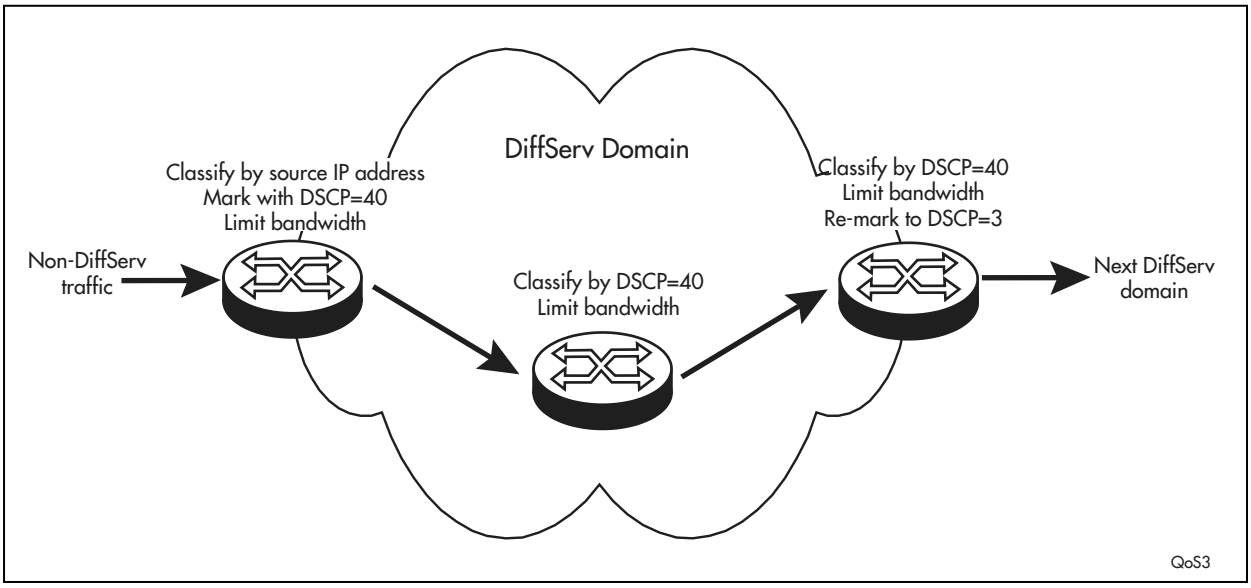
Figure 39-3 shows a simple example of a DiffServ domain. Packets that originate from a particular IP address have their IP headers marked with a DSCP value of 40 when they arrive at the edge of the domain. This DSCP value is preserved in the packets as they are sent to the next node. The next node classifies packets into flows according to their incoming DSCP and applies appropriate QOS functionality to them.
As the packets exit the domain they are re-marked with a different DSCP value, which are read and acted upon at the edge of the next QoS domain.

In this example the DSCP is determined by the nature of the packet (its source IP address). The DSCP can instead be determined by bandwidth conformance, so that packets which exceed their flow's allowed bandwidth at one node are treated differently by later nodes.

For more information about DiffServ, see RFC 2474, *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*.

1. The Differentiated Services field supersedes the IPv4 Type of Service (TOS) field and the IPv6 Traffic Class field.

Figure 39-3: Example of a DiffServ domain



Assured Forwarding and Expedited Forwarding

Assured Forwarding (AF) provides predefined DSCP values for four classes of service. Each class has DSCP values that correspond to a high drop probability, a medium drop probability and a low drop probability. You can use AF in a DiffServ domain to offer different classes of service, in which traffic which conforms with its bandwidth allocation has a lower drop probability than partially- or non-conformant traffic. AF is described in RFC 2597, *Assured Forwarding PHB Group*.

Expedited Forwarding (EF) also provides a predefined DSCP value, for a single "premium" service. You can use EF in a DiffServ domain to offer a low loss, low latency, low jitter, assured bandwidth, end-to-end service. EF is described in RFC 2598, *An Expedited Forwarding PHB*.

The DSCP values corresponding to each of these services are shown in Table 39-1, in binary and decimal.

Table 39-1: DSCP and TOS values for Assured Forwarding and Expedited Forwarding

Per Hop Behaviour (PHB)	Class	DSCP value (AF/EF name, then binary, then decimal)			IP TOS precedence value
Default		000000 0			0
Assured Forwarding		**Low drop prob**	**Medium drop prob**	**High drop prob**	
	Class 1	AF11 001010 10	AF12 001100 12	AF13 001110 14	1
	Class 2	AF21 010010 18	AF22 010100 20	AF23 010110 22	2
	Class 3	AF31 011010 26	AF32 011100 28	AF33 011110 30	3
	Class 4	AF41 100010 34	AF42 100100 36	AF43 100110 38	4
Expedited Forwarding		EF 101110 46			5

The Hierarchy

For each interface or tunnel, you need to build up your software QoS solution out of:

- A **policy**, which you attach to the interface. The policy defines a complete QoS solution for all traffic on the interface or group of interfaces.
- **Traffic classes** within the policy. Each traffic class defines the QoS processing for a group of traffic flows. The traffic flows for each traffic class are identified by classifiers or sub traffic classes attached to the traffic class. In software QoS, queues are contained within traffic classes.
- **Classifiers**, to sort traffic into the appropriate traffic classes. Classification is simply a method of dividing the incoming traffic into traffic flows so that packets of one type can be treated differently to packets of another type. The router supports two types of classifiers:
  - static classifiers, which can identify packets by a large number of characteristics. Available classification options depend on the interface and traffic type, and range from Layers 1 and 2 features (for example, destination VLAN), to Layer 3 features (for example, destination IP address) and Layer 4 features (for example, destination TCP port).
  - Dynamic Application Recognition objects, which identify and sort VoIP and video traffic.

Traffic Class Trees

Definition

Each software QoS policy contains a **traffic class tree**, which provides hierarchical queue scheduling. Depending on the policy settings and the types of traffic class in the tree, the policy empties queues using priority queuing (PQ), weighted round robin (WRR), deficit weighted round robin (DWRR) or mixed scheduling (PQ plus WRR or DWRR). For information about these queue scheduling methods, see "Queue Scheduling" on page 39-29.

First level

The first (top) level of the traffic class tree is made up of three weighted traffic classes:

- A **system** traffic class for important system traffic. On egress and tunnel policies, this includes ARP, RIP, RIPv2, BGP, OSPF, IPv6 control packets such as ND and NS, PPP control packets, ISAKMP, keepalive messages, and SNMP messages generated by the router. On ingress policies, it includes PPP control packets. The system class is a weighted traffic class with a configurable weighting, and a default weight of 20.
- A **root** traffic class. When you assign traffic classes to a policy, the router attaches them to the root traffic class. The root class is not configurable. Its weight is calculated using the formula:

  ```
  100 - SystemClassWeight - DefaultClassWeight
  ```

  By default, this gives a root traffic class weight of:

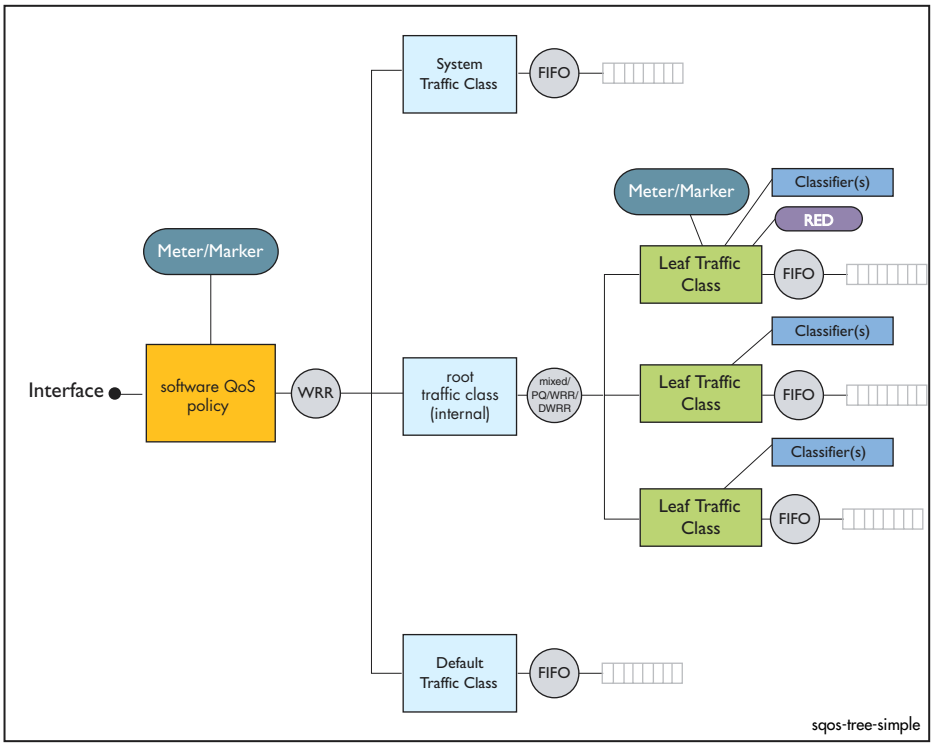
  ```
  100 - 20 - 0 = 80
  ```

- A **default** traffic class (DTC) for unclassified traffic. The DTC provides a catch-all for any traffic that does not match one of the traffic classes you assign to a policy. When you create a policy, the router creates a default traffic class with a weight of 0. A weight of 0 means that the DTC is only emptied if all other queues are empty. If you require other behaviour, you can specify another traffic class as the default instead.

Second level

Other levels of the traffic class tree are made up of traffic classes attached to the root traffic class. Figure 39-4 shows a simple two-level traffic class tree.

Figure 39-4: Simple traffic class tree



Multi-level trees

Figure 39-5 on page 39-12 shows a more complex multi-level traffic class tree, which is used in Configuration Example "6: Multiple Applications over Frame Relay" on page 39-82 to prioritise real-time traffic while controlling file server downloads. Multi-level trees offer hierarchical queuing and bandwidth management. You can attach up to three levels of traffic class to a policy.

This chapter uses the following terms for traffic classes within complex traffic class trees:

- **Intermediate**

  A traffic class that contains one or more other traffic classes is called an *intermediate* traffic class.

- **Sub**

  A traffic class that is part of an intermediate traffic class is called a *sub* traffic class.

- **Leaf**

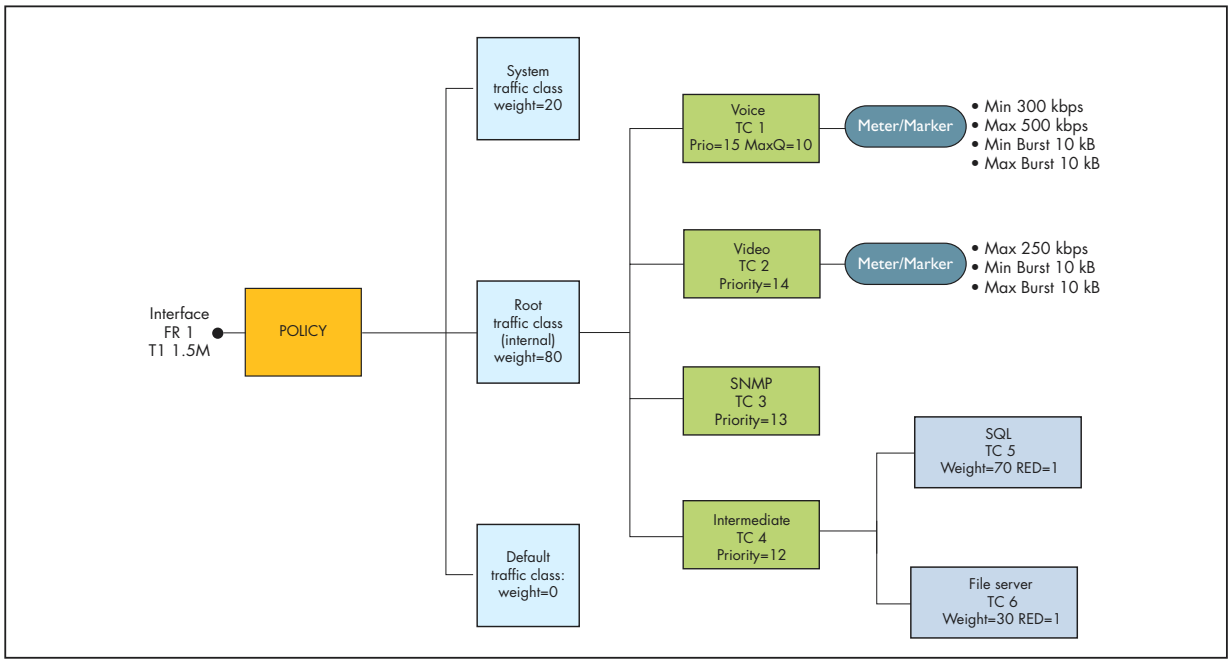
  A traffic class that contains no sub traffic classes, and that is therefore at the edge of the tree, is called a *leaf* traffic class.

- **Sibling**

  Traffic classes at the same position in the traffic class tree are called *sibling* traffic classes.

You should plan your traffic class tree so that traffic classes containing latency and jitter-sensitive traffic, such as VoIP, are not children of any weighted traffic classes (except for the root class).

Figure 39-5: Complex traffic class tree



Traffic Classes

For each traffic class, you can specify a combination of the following:

- the bandwidth class and DSCP that packets are given on admission to the traffic class
- the meter the traffic class uses and how the router responds to non-conformant packets
- if and how the router modifies the packet DSCP and/or VLAN priority when dequeuing packets. The router can assign the DSCP on the basis of metering results.
- for leaf traffic classes, the RED curve set.
- for leaf traffic classes, the maximum queue length, notification when queue length is exceeded, total internal bandwidth available to the traffic class, and whether packets are dropped from the head or tail of the queue.
- the priority or weight of this traffic class compared with others in the policy's traffic class tree
- the scheduling method used for weighted traffic classes that are attached to this traffic class (WRR or DWRR)

These settings specify the action the router takes at each of the software QoS processing stages described in "Software QoS Stages" on page 39-18. Settings in a traffic class override settings in the policy.

Policies

For each policy, you can specify a combination of the following:

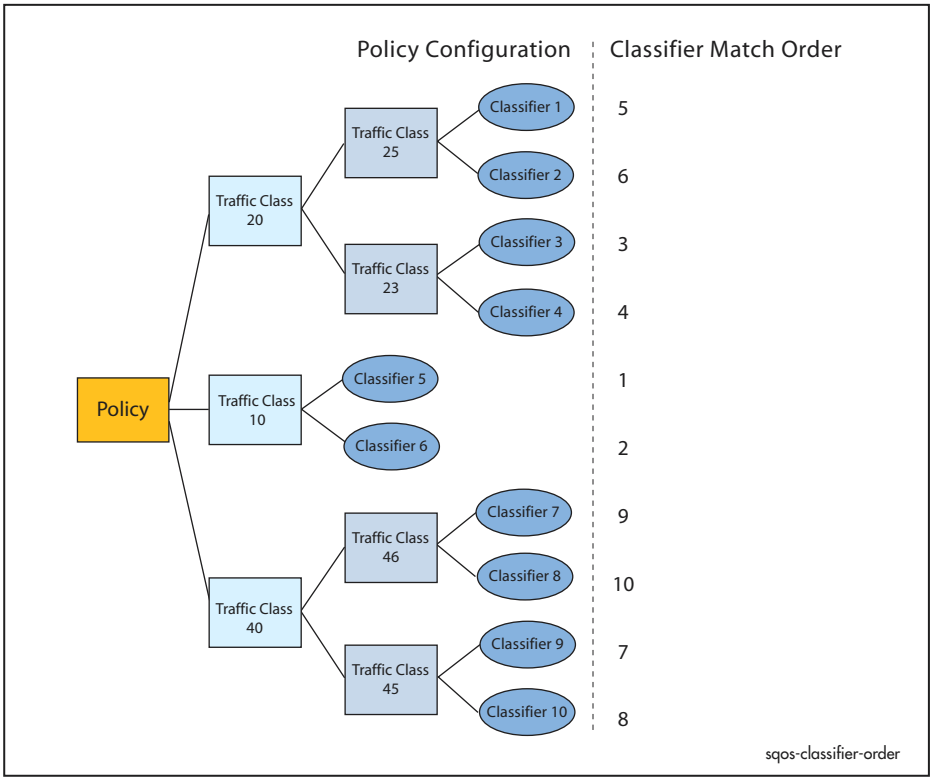
- the meter the policy uses and how the router responds to non-conformant packets
- if and how the router modifies the packet DSCP and/or VLAN priority when dequeuing packets
- total internal bandwidth available to the policy
- the scheduling method used for weighted traffic classes in the policy's traffic class tree
- the amount of resource given to system traffic

These settings specify the action the router takes at each of the software QoS processing stages described in "Software QoS Stages" on page 39-18. Settings in a traffic class override settings in the policy.

Order of Classifier Matching

Each packet is sorted according to the first classifier in the classifier list that matches it. Within each policy, classifiers are listed first by the ID number of the traffic class they are attached to, then by classifier ID number. Figure 39-6 shows the order in which classifiers are matched in an example of a traffic class tree. In this example, if a packet matched the criteria of classifier 2 and of classifier 6, then the packet would be associated to traffic class **10** because classifier 6 is above classifier 2 in the match order.

Figure 39-6: Order of classifier matching in a traffic class tree



Dynamic Application Recognition for Voice and Video

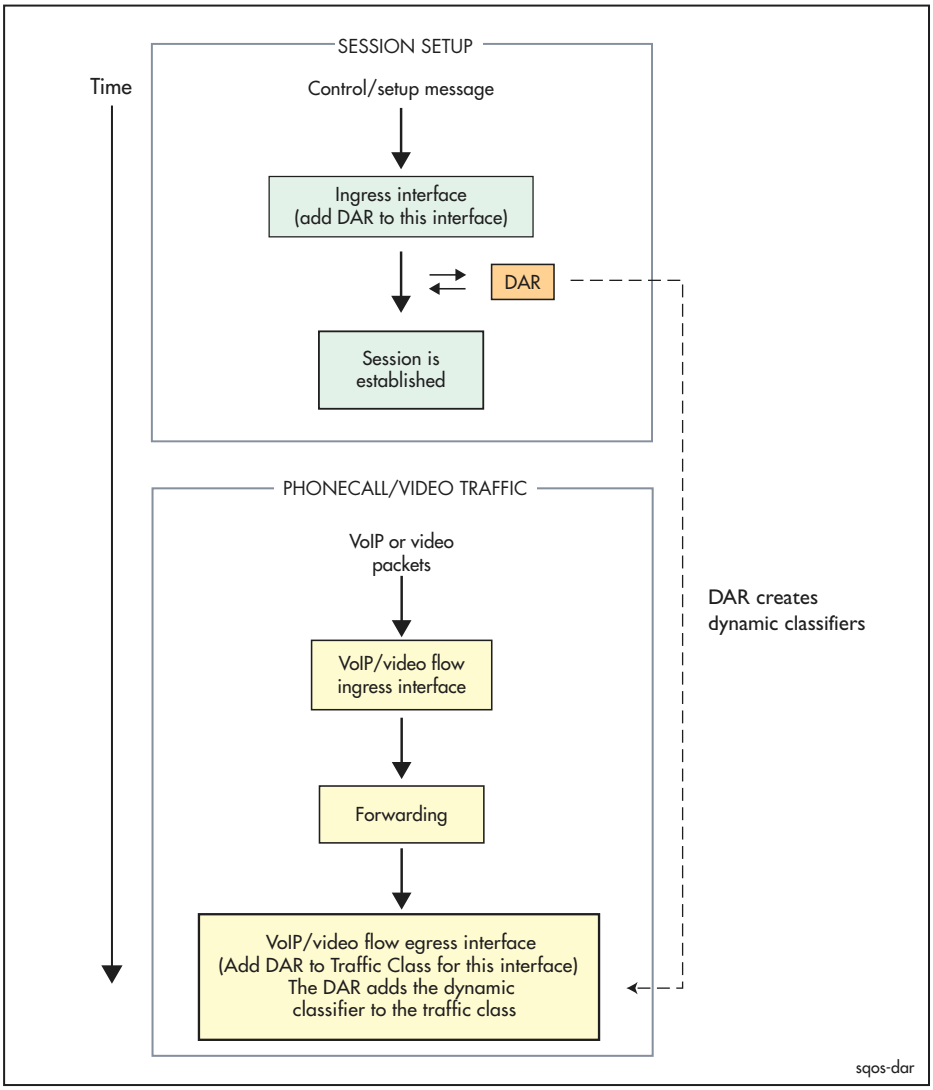
The Dynamic Application Recognition (DAR) system applies full software QoS functionality to voice and video packets, by creating dynamic classifiers.

The stages of the DAR process (Figure 39-7) are:.

1. The router examines incoming voice or video session initiation messages that arrive at an interface, and compares them against a DAR object. The DAR object tells the router what kind of session to match on that interface. The router creates a dynamic classifier to match the session, and applies it to the interface that uses a traffic class to which the DAR object belongs.
2. The router uses that dynamic classifier to sort voice or video packets into traffic classes and apply the configured QoS processing to them.

In most networks the control messages and VoIP/video traffic flow ingress the same interface, but the system does not require this.

Figure 39-7: Process flow for Dynamic Application Recognition (DAR)



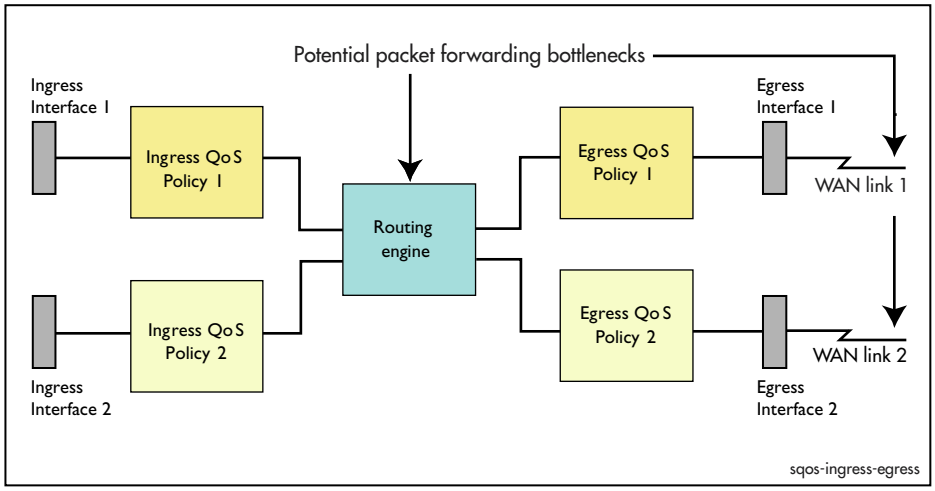
Processing Points

The router performs software QoS at the following points:

- the ingress interface, immediately after the packet arrives at the router, and/or
- the egress interface, before the packet leaves the router, or
- the entry to a tunnel, before the packet is encapsulated.

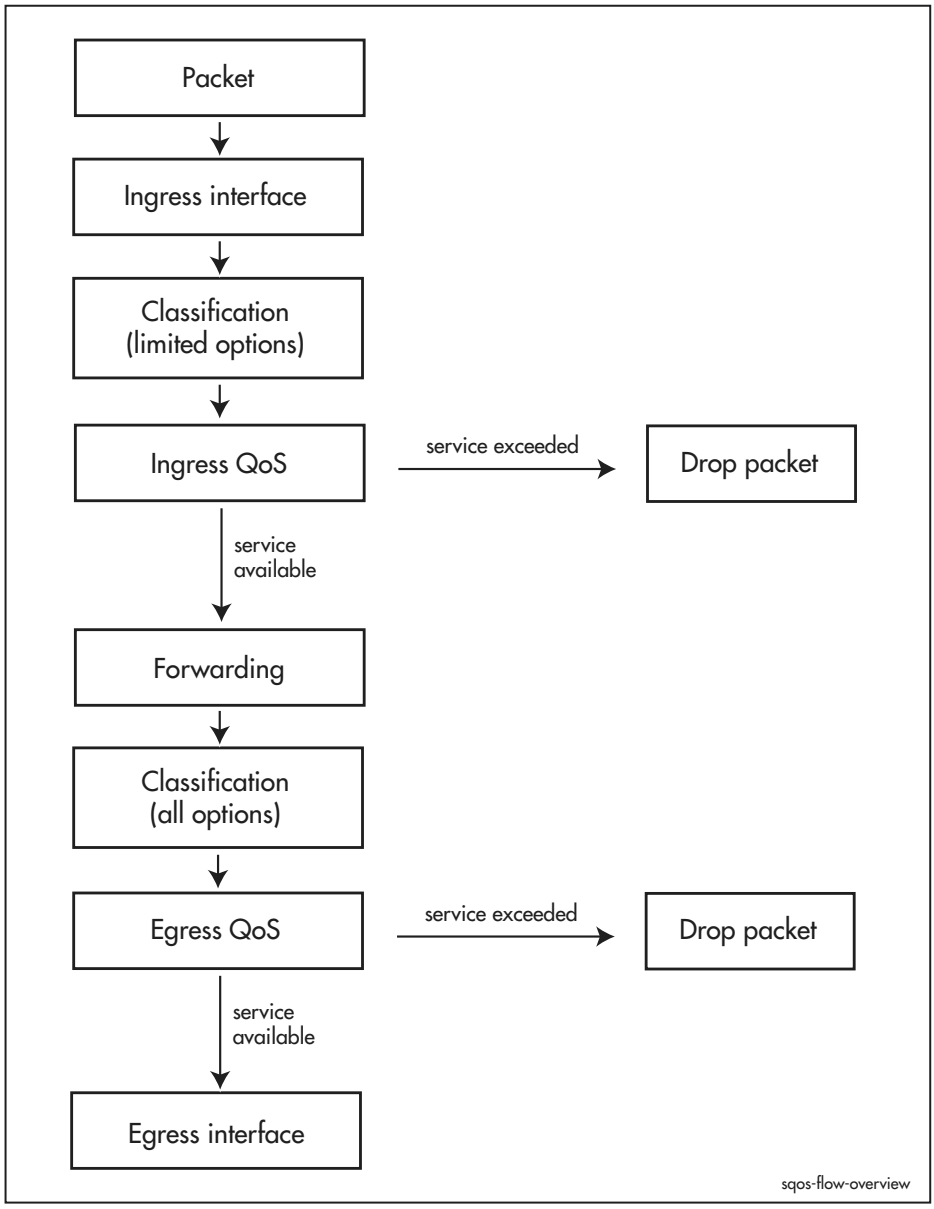
Typically you would apply ingress QoS to high-bandwidth interfaces because they can send enough traffic to oversubscribe the router's routing engine. You would apply egress QoS to low-bandwidth interfaces because the router can oversubscribe those links.

Figure 39-8: Points in the router that can be oversubscribed



The flow from ingress interface to egress interface is summarised in Figure 39-9.

Figure 39-9: Summary of packet flow through the router



Ingress QoS

In heavy traffic conditions, processing within the router's routing engine can itself be a bottleneck. Ingress QoS lets you give some packets preferential access to the routing engine. It identifies very high priority traffic and ensures the router processes it as quickly as possible, or identifies very low priority traffic and drops it if necessary.

Ingress QoS processing adds to the load on the CPU. To minimise the impact, an ingress QoS policy must:

- not use a QoS meter, on the policy or on any of its traffic classes
- not include traffic classes that use RED curves
- have all its traffic classes at the same level, rather than using a multi-level tree (see "Traffic Class Trees" on page 39-10)
- only include traffic classes with classifiers that use VLAN priority, IP DSCP and IP TOS.

To minimise the processing overhead, we recommend that you minimise the number of QoS entities you create, and in particular that you configure as few ingress classifiers as possible.

Egress QoS

The most significant QoS processing takes place at the egress interface, on WAN links with limited bandwidth. QoS at the egress interface can classify packets according to a wide range of characteristics in Layers 1 to 4. It is designed to affect the fate of the packet:

- while the packet is still within the router

  Its fate can be only one of two possibilities: it is either dropped or placed in an egress queue. Packets that belong to an "important" traffic flow are placed in a high priority egress queue to ensure their timely delivery. Less important packets are placed in a lower priority queue or may even be dropped if the router gets congested.

- as the packet crosses the QoS domain

  This involves permanently marking the packet in a way that downstream devices understand so they can determine the actions they will take on the packet.

These types of action are not mutually exclusive. A single action that is taken on a packet may well determine which egress queue it is assigned to and also determine how a downstream device will treat the packet.

Tunnel QoS

QoS at a tunnel is performed on packets before they are encapsulated and enter the tunnel. Tunnels are at Layer 3 in the OSI model, so the router can classify tunnelled traffic according to characteristics for Layers 3 and 4, plus ingress interface and port. The router can perform all QoS functions on the classified traffic.

Software QoS Stages

This section summarises the processes that make up a QoS solution, then explains them in detail.

Software QoS consists of the following overarching processes:

- Classification: putting packets into traffic classes

  As described in "The Hierarchy" on page 39-9, the router uses static or dynamic (DAR) classifiers to identify packets.

- Enqueuing: putting packets into queues, or dropping them if the queues are oversubscribed

  Premarking, metering, RED curves, and tail or head drop interact to determine which packets get put into queues.

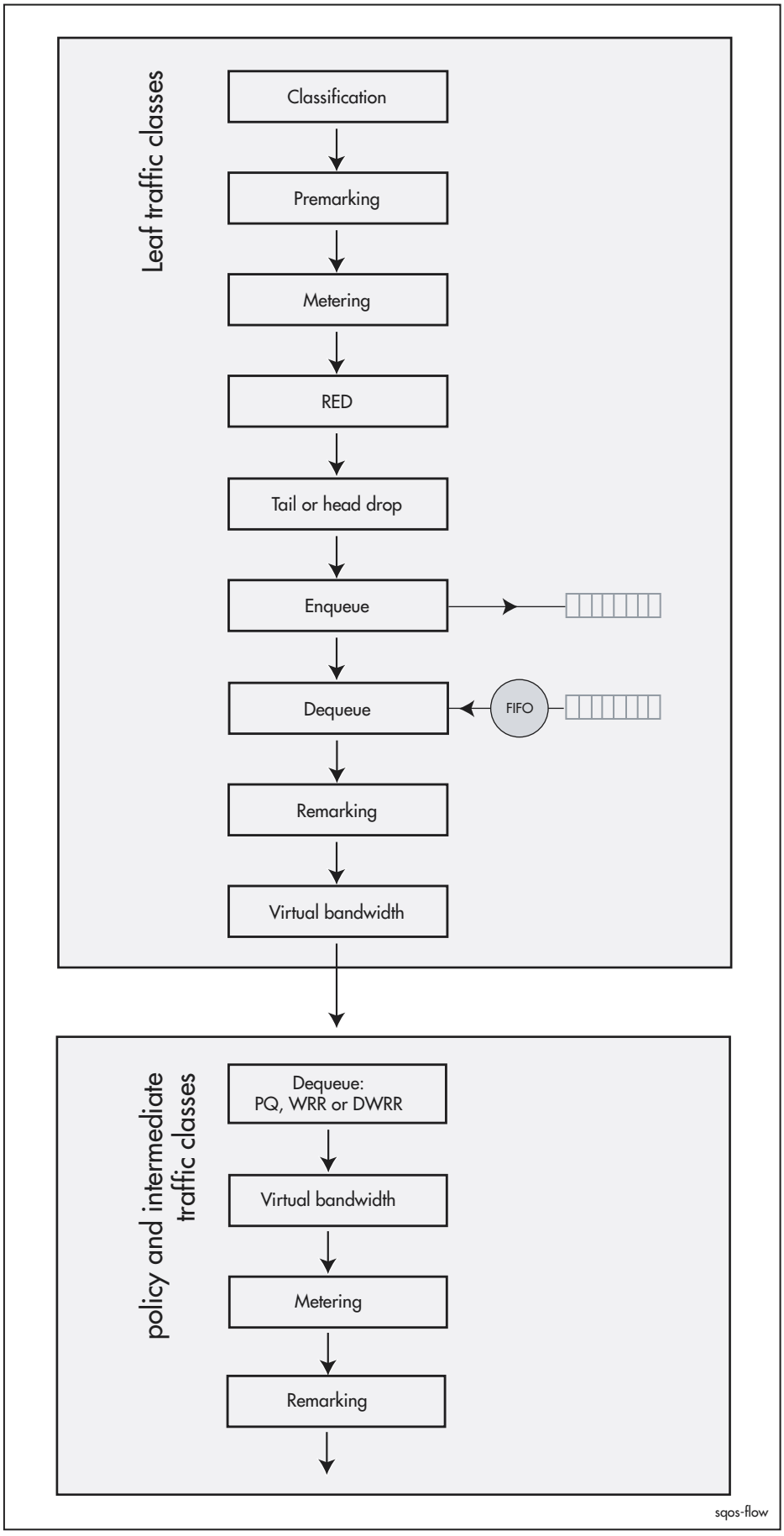
- Dequeuing: removing packets from the queues in appropriate order for transmission.

  Virtual bandwidth, metering, and re-marking can apply to packets as they are dequeued.

Packet Flow

Figure 39-10 on page 39-19 shows the path of a packet through the QoS engine. The diagram shows all possible QoS stages for completeness, but few QoS solutions will require every tool. As described in "Processing Points" on page 39-15, each stage applies equally to QoS on ingress interfaces, egress interfaces and tunnels; the only difference is the characteristics on which you can classify traffic.

Figure 39-10: Detailed packet flow through the software QoS system



Classification: Identifying and Sorting Traffic

Once you have enabled Software QoS, all traffic is examined by the classifier, which assigns packets to a traffic class on the basis of any combination of a large number of characteristics. Therefore, the first step in QoS is to create classifiers for each type of sensitive traffic. You can also configure the router to create dynamic classifiers for VoIP traffic, which are referred to as DAR objects.

For information about the classifier see Chapter 38, Generic Packet Classifier. For more information about dynamic classifiers see "Dynamic Application Recognition for Voice and Video" on page 39-14.

Bandwidth Class

Before and during QoS processing the router assigns packets to bandwidth conformance class 1, 2, or 3. At the start of the QoS process, the bandwidth class may indicate how well the packet's flow conformed with its bandwidth allocation at the previous hop. This is done during the *premarking* process. During the QoS process, the bandwidth class indicates how well the packet's flow conforms with its bandwidth allocation at this router. Bandwidth class can also be described using a 3-colour model as shown in the following table.

Bandwidth class	Colour	Meaning
1	Green	Conformant: Processing this packet leaves the flow within acceptable bandwidth limits. If this packet causes the flow to burst, the burst is acceptably low.
2	Yellow	Partially conformant: Processing this packet causes the flow to burst, but the burst is not unacceptably large.
3	Red	Nonconformant: Processing this packet causes an unacceptably high burst.

Premarking: Labelling Packets Before Metering

Premarking assigns the packet to a bandwidth class and/or replaces the packet's initial DSCP value. It occurs immediately after classification, before the router applies any bandwidth metering to a traffic class.

The "pre" part of premarking means this process happens before any bandwidth metering takes place, so involves no measurement of actual bandwidth use. The "marking" part refers to attaching QoS information to packets.

Premarking to assign the packet to a bandwidth class enables you to use the bandwidth conformance information from the previous hop. By default, all packets are assigned to bandwidth class 1 (green, conformant).

Premarking to set the DSCP enables you to identify or mark traffic appropriately at the edge of a DiffServ domain. For example, you can create a traffic class for each of the four AF classes, and premark packets with the appropriate DSCP for that AF class (see Table 39-1).

The following options let you specify a new bandwidth class and DSCP:

- directly, by specifying a new value for all packets that belong to the traffic class.
- by using the premarking table of a DSCP map. The router reads the packet's current DSCP and looks up the table to determine the new values for that DSCP. DSCP maps enable you to premark packets with different incoming DSCPs differently.

Table 39-2: Conceptual diagram of part of a premarking table in a DSCP map

Original DSCP	New Bandwidth Class	New DSCP
0	newbwclass	newdscp
1	newbwclass	newdscp
. . .		
63	newbwclass	newdscp

Metering: Bandwidth Conformance

Metering means measuring how much bandwidth packets in a traffic flow use, and how well usage conforms to bandwidth specifications for the traffic class to which the flow belongs. It assigns the packet to a bandwidth class depending on its conformance. For more information, see Bandwidth Class on page 20.

Note that the purpose of metering is to assign the packet to a bandwidth class, not to discard or queue it. Other QoS processes use metering results later to determine how to process the packet.

The router supports Single Rate Three Colour Marker meters as described in RFC 2697, and Two Rate Three Colour Marker meters as described in RFC 2698.

Both meters are based on the concept of a *committed* rate plus a level of committed burst, below which packets conform. A certain excess is allowed above this level. If the flow exceeds the committed rate and exhausts the committed burst size, but not the excess, its packets are marked partially conformant. For the single rate meter, the excess can only be a temporary burst (the **excess** burst). If packets are sent at a steady rate that exceeds the committed rate, the single rate meter eventually marks them as non-conformant. For the two rate meter, the excess can be steady. The two rate marker only marks packets as non-conformant if they exceed its second rate (the *peak* rate). Therefore, the two rate marker can give you two different rate options.

Single Rate Three Colour Marker Meter

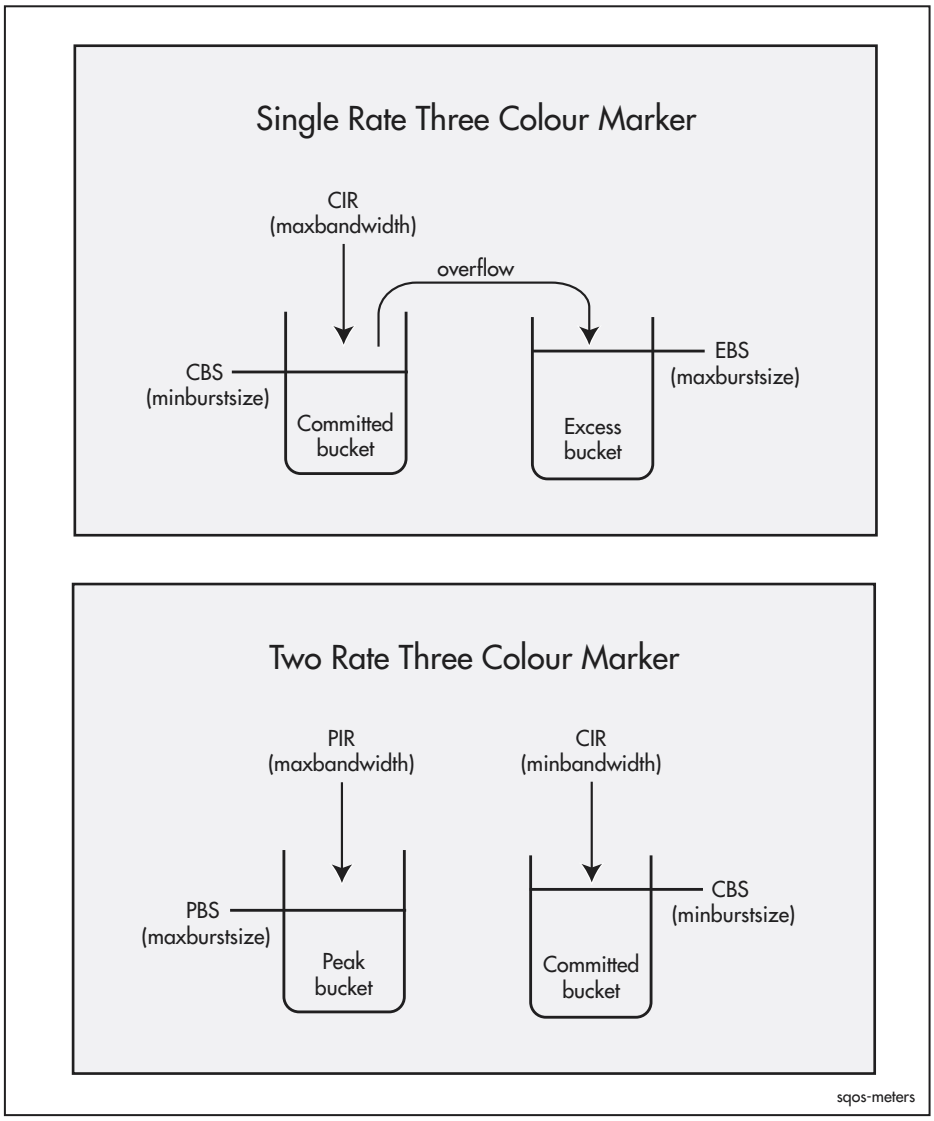
The meter is based on a token bucket model with two token buckets. Each token represents one byte. The size of one bucket is the Committed Burst Size (CBS) and the other is the Excess Burst Size (EBS). The CBS bucket is refilled at a rate called the Committed Information Rate (CIR) and the EBS bucket is refilled from the overflow of the CBS bucket. Figure 39-11 on page 39-22 shows the model and the names of the associated parameters.

Colour-blind

The meter can be either colour-blind or colour-aware. When a packet arrives at a colour-blind meter, the router determines if the packet is smaller than the number of tokens (bytes) in the CBS bucket. If the CBS bucket contains enough tokens, the router removes those tokens from the CBS bucket and assigns the packet to bandwidth class 1 (conformant, green). If the CBS bucket does not contain enough tokens but the EBS bucket does, the router removes the tokens from the EBS bucket and assigns the packet to bandwidth class 2 (partially conformant, yellow). If the packet is larger than the number of tokens in either bucket, the router assigns the packet to bandwidth class 3 (non-conformant, red).

Note that the metering process compares the number of tokens in the buckets to the size of the packet, for each packet. Therefore for metering to work logically, the buckets must be at least as big as the packets passing through the system. The burst sizes define the bucket size, so one or both of the CBS (**minburstsize**) or EBS (**maxburstsize**) must be at least as large as the packets being metered. It is usually best to configure a burst size that is several times the MTU.

Figure 39-11: Single and two rate three colour marker meters



Colour-aware

The process is similar for a colour-aware meter, except that the metering process depends on the initial bandwidth class (colour) of the packet. If the packet is in bandwidth class 3 (red) before metering, the router leaves it in bandwidth class 3. If the packet is in bandwidth class 2 (yellow) before metering, the router meters it from the EBS token bucket only. If the EBS bucket has enough tokens, the packet stays in bandwidth class 2; if not, the router assigns it to bandwidth class 3. If the packet is in bandwidth class 1 (green) before metering, the router uses both token buckets, as described above for a colour-blind meter. Both meters have the same effect on packets that were conformant at the previous router, but you can use a colour-aware meter to stop packets that were non-conformant or partially-conformant at the previous router from being marked conformant at this router.

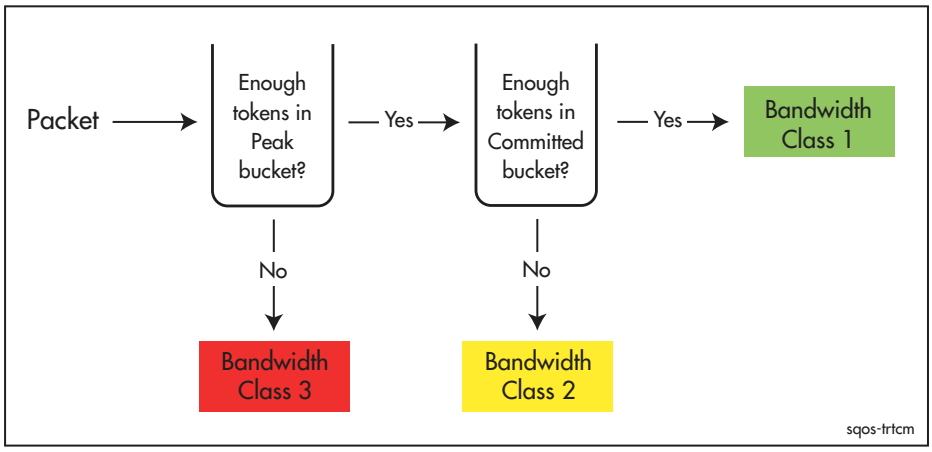
Two Rate Three Colour Marker Meter

The two rate meter also has two token buckets, but they are filled separately, and are inspected in the opposite order. The first token bucket is called the Peak bucket. It is refilled at the Peak Information Rate (PIR) to a maximum of the Peak Burst Size (PBS). The second token bucket is called the Committed bucket. It is refilled at the Committed Information Rate (CIR) to a maximum of the Committed Burst Size (CBS). Figure 39-11 on page 39-22 shows the model and the names of the associated parameters.

Figure 39-12 on page 39-23 shows the meter's decision flow. The meter first compares the packet size with the number of tokens in the Peak bucket. If the Peak bucket does not have enough tokens, the meter assigns the packet to bandwidth class 3 (non-conformant, red). If the Peak bucket has enough tokens, the meter subtracts them from the Peak bucket and then compares the packet size with the number of tokens in the Committed bucket. If the Committed bucket does not have enough tokens, the meter assigns the packet to bandwidth class 2 (partially conformant, yellow). If the Committed bucket has enough tokens, the meter subtracts the tokens and assigns the packet to bandwidth class 1 (conformant, green).

The meter functions correctly only when the Committed bucket is less full than the Peak bucket most of the time. Therefore, PIR must be greater than or equal to CIR.

Figure 39-12: How the two rate three colour marker assigns bandwidth class



Acting on Flow Conformance

If the meter assigns a packet to bandwidth class 3 (non-conformant), you can configure it to drop the packet. You can also configure it to pause the traffic flow by not dequeuing packets belonging to that flow for some seconds. If it pauses a traffic flow, it can produce a log message and SNMP trap.

The packet can be marked according to the metering results when it leaves the router, by changing the DSCP. These options are described in "Re-Marking" on page 39-31.

Packet Queuing

Each leaf traffic class contains a queue, which stores packets. The router first enqueues packets that belong to the traffic class—puts packets into the queue. When appropriate resource becomes available, it dequeues them—removes them from the queue—and sends them out in order of priority at controlled speed. Therefore queues can reduce jitter by smoothing the traffic flow.

You can also use enqueuing to control which packets the router drops when congested. Parameters which interact include:

- Queue length. When the queue gets too full, packets are dropped.
- RED. RED is an active queue management technique that randomly discards packets as the queue builds up. RED is described below.
- Head or tail drop. With tail drop, the router drops packets from the tail of the queue. With head drop, the router drops packets from the queue head. Tail dropping drops the newest packets; head dropping drops the oldest.

  Head drop may be more appropriate than tail drop for real-time applications, for example voice traffic, because the latest data is of greater interest than older data.

RED Curves

Random Early Detection/Discard (RED) is a congestion avoidance mechanism that allows the router to drop packets randomly before the egress queue exceeds the allocated maximum queue length. It is bandwidth class aware, therefore it can drop less conformant packets when some congestion occurs, and can drop more conformant packets as congestion becomes more severe.

Each RED curve set consists of three curves, one for each bandwidth class. Generally RED curves are designed so that bandwidth class 3 packets start to be dropped when the average queue length reaches a reasonably low threshold. Bandwidth class 2 packets start to be dropped at a higher average queue-length threshold and bandwidth class 1 packets start to be dropped at a higher still threshold.

RED stops the router from dropping bursts of packets and therefore breaks the global synchronisation of TCP flows. This maximises link utilisation. Using RED on UDP traffic flows offers little advantage because UDP has no inherent congestion detection mechanism and does not react to packet drops.

For each bandwidth class *x*, the parameters used in defining a RED curve are:

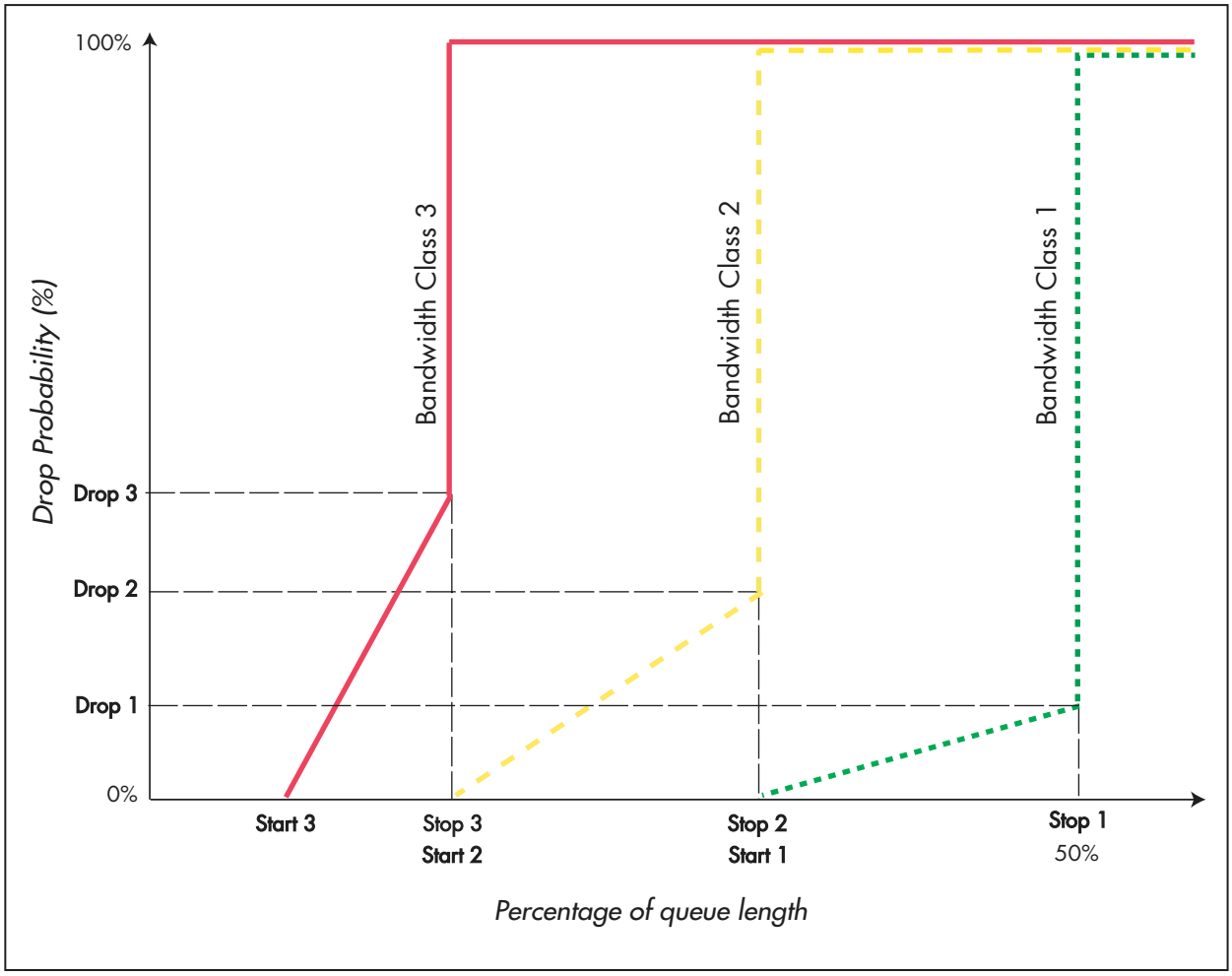
- **start***x*: the average percentage of queue length below which packets belonging to bandwidth class *x* are always accepted.
- **stop***x*: The average percentage of queue length above which all packets belonging to bandwidth class *x* are discarded.

  As the queue length increases, the random probability that a packet in that class will be dropped increases linearly until the queue length reaches the **stop***x* value.
- **drop***x*: The probability that a packet belonging to bandwidth class *x* will be dropped at the queue length determined by the **stop***x* value.

The RED algorithm calculates a running average queue length which the Start and Stop values use. Because they are averages, Start and Stop must be less than 100% of the maximum queue length.

Figure 39-13 shows an example of a RED curve set. Note that packets from bandwidth class 3 begin to be dropped while the queue is still quite short (Start 3), whereas the queue is half full before packets from bandwidth class 1 begin to be dropped (Start 1), even in this aggressive RED curve set.

Figure 39-13: Default aggressive RED curve set



Three RED curve sets exist by default. The following table shows their properties.

ID	Description	BW Class 1 (green)			BW Class 2 (yellow)			BW Class 3 (red)		
		Start	Stop	Drop prob.	Start	Stop	Drop prob.	Start	Stop	Drop prob.
0	Aggressive	35%	50%	20%	20%	35%	30%	10%	20%	40%
1	Medium	50%	70%	20%	30%	50%	30%	15%	30%	40%
2	Passive	80%	95%	20%	60%	80%	30%	40%	60%	40%

Dequeuing

The processes discussed up to this point are part of *enqueuing*; they determine which packets the router puts into the egress queues for each traffic class. The next processes are applied during *dequeuing*, when the router removes packets from the queues to send out.

The traffic class tree is the central structure that supports the dequeuing process. Each leaf traffic class has its own queue. This section gives an overview of the process of dequeuing, and the following sections discuss aspects of the dequeuing process in detail.

Summary of Dequeuing Flow of Events

In the dequeuing process, all the weights and/or priorities assigned to the traffic classes actually come into play. The general flow in the dequeuing process is:

1. The physical interface (or egress tunnel) calls on the policy to give it a packet to send onto the line.
2. The policy, in turn, calls on one of the traffic classes below it, to send up a packet (so that it can be passed onto the interface). It uses the WRR scheduling algorithm to determine whether the root, system, or default classes gets the current opportunity to send a packet.
3. The chosen class, in turn, calls upon one of its subclasses to send up a packet. Again, the choice of which subclass it calls on depends on a priority or round-robin scheduling algorithm.
4. And so on, right down to the leaf traffic classes.

Pull not push

This method by which packets make their way from the traffic classes to the interface is a *pull* mechanism. In this pull mechanism, the traffic class tree as a whole implements the scheduling algorithm, providing multiple choices as to which traffic class is called upon to pass a packet up to the next level of the tree.

Applying QoS Controls on Intermediate Classes

The dequeuing process involves more than just the pulling of packets up through the traffic class tree. Intermediate traffic classes, and the policy itself, can apply the following processes to packets on the way through:

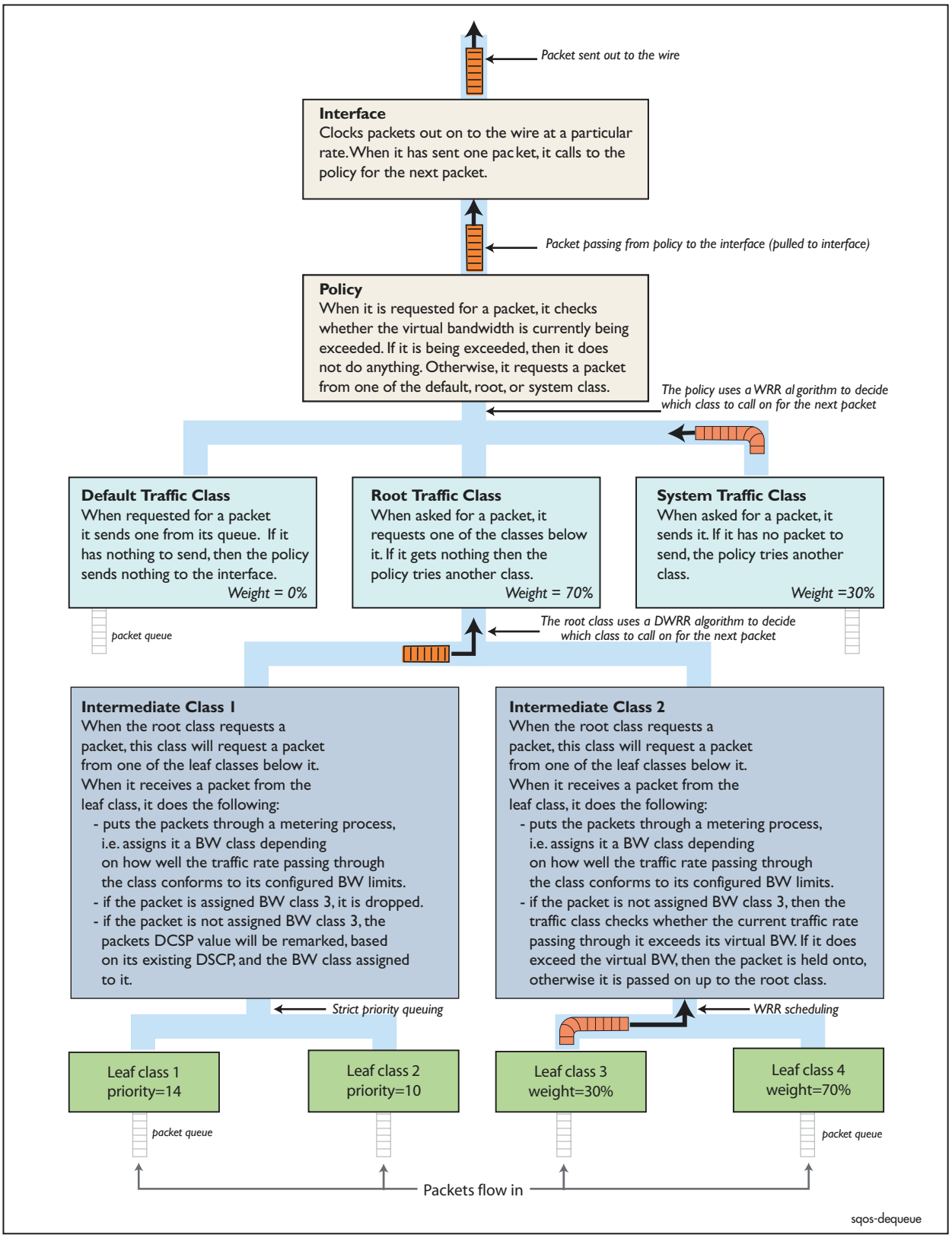
- metering
- re-marking
- virtual bandwidth shaping

Any combination of these processes can be defined on any of the intermediate traffic classes and on the policy. To illustrate this, consider an example in which a policy has:

- a virtual bandwidth of Vp
- a default traffic class with weight = 0%
- a system traffic class with weight = 30%
- a root traffic class with weight = 70%
  The root class is using DWRR to schedule the drawing of packets from the classes beneath it.
- two intermediate traffic classes below the root traffic class, one configured with metering and DSCP re-marking, the other configured with metering and virtual bandwidth. Both intermediate traffic classes are configured to drop bandwidth class 3 packets.
- two leaf classes on each intermediate class

The dequeuing process for this example is shown in Figure 39-14 on page 39-28.

Figure 39-14: Example of dequeuing process



Queue Scheduling

Queue scheduling refers to how the router determines the order in which to empty queues, and what proportion of the bandwidth goes to packets from each queue.

In a multi-level traffic class tree, the policy pulls packets out of the leaf traffic class queues on a first-in-first-out (FIFO) basis into their intermediate traffic class queues, as described in "Dequeuing" on page 39-26. It then uses the scheduling method defined for the intermediate traffic class to dequeue the packets upwards in the tree. This continues until they reach the policy and are sent out.

The scheduling methods for queues are:

- Priority Queuing (PQ)

  To select priority queuing, you assign a priority to each traffic class. Queues from higher priority traffic classes are emptied before any packets are transmitted from queues from lower priority traffic classes. For example, this means that queues for a traffic class with priority 7 must be totally empty before any packets from a traffic class with priority 6 or lower are sent. We suggest you put latency and jitter-sensitive traffic, such as VoIP, into a high priority traffic class, rather than a weighted class.

  A risk with priority queuing is that high priority traffic may use all the available bandwidth, forcing the router to drop all medium and low priority traffic.

  Note that when all your traffic classes use priorities, the policy still uses WRR or DWRR to assign a proportion of the bandwidth to system and default traffic.

  For the procedure to configure this method, see "Priority queuing" on page 39-43.

- Weighted Round Robin (WRR)

  To select WRR, you assign a weighting to each traffic class and specify WRR in the policy or intermediate traffic class to which the traffic classes belong. The traffic classes share bandwidth on the basis of these weightings. Using this method, the router can transmit packets from all traffic classes even under conditions of high congestion. You can configure the weightings to ensure that more packets per second are sent from traffic classes that are more sensitive to packet drop, than from less sensitive or lower priority traffic classes.

  A disadvantage of WRR is that flows with large packets get more than their fair share of the bandwidth. We suggest you plan your traffic class tree so that the VoIP traffic class is not a child of a weighted traffic class (except for the root class).

  For the procedure to configure this method, see "WRR and DWRR queuing" on page 39-44.

- Deficit Weighted Round Robin (DWRR)

  To select DWRR, you assign a weighting to each traffic class and specify DWRR in the policy or intermediate traffic class to which the traffic classes belong.

  DWRR is similar to WRR, but much fairer across packets of different sizes. DWRR provides similar functionality to Class Based Weighted Fair Queuing (CBWFQ), with less performance impact.

  For the procedure to configure this method, see "WRR and DWRR queuing" on page 39-44.

- Mixed scheduling

  This method is equivalent to Low Latency Queuing (LLQ), and occurs when some traffic classes are priority classes and some are weighted.

  To select mixed scheduling on a policy or intermediate traffic class, you assign priorities to one or more of its sub traffic classes and weightings to the rest. The router first services all the priority traffic classes, then the remaining bandwidth is divided among the weighted traffic classes.

  For the procedure to configure this method, see "Mixed scheduling" on page 39-45.

Example of Mixed Scheduling

To demonstrate the effect of mixed scheduling, consider an interface with a speed limit of 2 Mbps, and 4 different flows, each arriving at up to 2 Mbps. The interface has a policy attached to it that contains 4 traffic classes, one for each flow:

- Traffic class 1, priority=15
- Traffic class 2, weight=50
- Traffic class 3, weight=30
- Traffic class 4, weight=20

The policy uses DWRR to schedule the 3 weighted traffic classes. The following table shows the egress rate ("out rate") for each traffic class, when traffic arrives at the traffic class at different rates ("in rate"). This table shows that the policy schedules all packets from the priority traffic class first, up to the 2 Mbps limit. If any bandwidth is left over, the policy schedules packets from the weighted traffic classes. The bandwidth is divided among the traffic classes that have packets to send, in proportion to their weights.

traffic class 1		**traffic class 2**		**traffic class 3**		**traffic class 4**	
in rate (Mbps)	**out rate (Mbps)**	**in rate (Mbps)**	**out rate (Mbps)**	**in rate (Mbps)**	**out rate (Mbps)**	**in rate (Mbps)**	**out rate (Mbps)**
2	2	2	0	2	0	2	0
1	1	2	0.5	2	0.3	2	0.2
1	1	0	0	2	0.6	2	0.4
0	0	0	0	0	0	2	2

Re-Marking

The router re-marks packets after it removes them from the traffic class queue, as part of the process of sending the packet out the egress interface. Unlike premarking, re-marking occurs after metering, so the packet markers depend on its conformance to its bandwidth allocation. Re-marking can change the packet:

- DSCP value. This enables you to mark the packet with information about its conformance for devices further downstream to use. For example, you may choose to mark bandwidth class 3 packets with a particular DSCP so a downstream device can give them a lower priority. The following options let you specify a new DSCP:
  - directly, by specifying a new DSCP for the traffic class
  - using the re-marking table of a DSCP map (see Table 39-3). The router reads the packet's DSCP and metered bandwidth class, and looks up the table to determine the new DSCP value for that combination of DSCP and bandwidth class.
- bandwidth class. This enables hierarchical processing within a traffic class tree. To re-mark with a new bandwidth class, the router uses the re-marking table of a DSCP map. It reads the packet's current DSCP and metered bandwidth class and looks up the table to determine the new bandwidth class for that combination.
- VLAN priority. This enables downstream routers to give a higher priority to packets, assuming those downstream routers are configured to do so.

Table 39-3: Conceptual diagram of part of a re-marking table in a DSCP map

Current DSCP	Bandwidth Class 1		Bandwidth Class 2		Bandwidth Class 3	
	New BW class	New DSCP	New BW class	New DSCP	New BW class	New DSCP
0	newbwclass	newdscp	newbwclass	newdscp	newbwclass	newdscp
1	newbwclass	newdscp	newbwclass	newdscp	newbwclass	newdscp
. . .						
63	newbwclass	newdscp	newbwclass	newdscp	newbwclass	newdscp

Virtual Bandwidth

The virtual bandwidth for a policy or traffic class determines the maximum rate at which data can leave the internal queues to be transmitted onto the physical media. Setting a virtual bandwidth limit allows you to shape traffic by limiting the total bandwidth available to a policy or an intermediate traffic class.

Note that the rate you set needs to be lower than the required actual transmission rate, because the virtual transmission rate does not include the transmission of bits for the inter-frame gap, and some packet framing. For most packet types the difference is a few bytes. Virtual bandwidth may not be a useful tool for tunnelled packets, because packet padding may mean the difference is many bytes.

If you specify a virtual bandwidth for a policy or traffic class (intermediate or leaf), give the appropriate leaf traffic classes large maximum queue lengths. This enables them to buffer bursts of packets and avoids packet loss.

Configuring a Software QoS Hierarchy

This section first describes how to build up your software QoS solution for an interface, out of classifiers, DAR objects, traffic classes and a policy. Then it describes how to change the default traffic class for QoS on unclassified traffic.

Later sections describe how to configure individual stages of the QoS solution.

The Total Software QoS Solution

This section gives a complete sequence of steps for configuring a QoS hierarchy, to show all commands available and their connections.

Before you start

We strongly recommend that you plan the QoS scheme for your whole network on paper before applying it on the router. In particular, we recommend you minimise the number of classifiers you configure, to maximise performance.

You will need the following information about your network:

- The different types of traffic flow that are currently performing below an acceptable standard, or that need bandwidth control
- Whether the router is dropping an unacceptable number of packets at ingress
- The parameters you can use to uniquely identify each flow
- The interfaces through which the flows ingress and/or egress
- The tunnels through which the flows egress
- The level of service each flow requires (for example, bandwidth requirements, relative priorities)

Configuration rules

The following rules apply when building up a software QoS hierarchy

- An interface may only have one ingress policy and/or one egress policy.
- A tunnel may only have one software QoS policy.
- A policy may be assigned to many interfaces and tunnels.
- A policy may have many traffic classes.
- A traffic class may only be assigned to one policy or intermediate traffic class.
- An intermediate traffic class may have many sub traffic classes.
- A classifier may only be assigned to leaf traffic classes.
- A DAR object may be assigned to many leaf traffic classes. However, a DAR object should only be used once within a policy. A DAR object applies dynamic classifiers to traffic classes to which it is assigned.
- A leaf traffic class may have many classifiers and/or DAR objects.
- A classifier may be assigned to many leaf traffic classes. However, a classifier may only be used once within a policy.

Configuration order

The steps for configuring a software QoS hierarchy are in Table 39-4 in a functional order. You probably do not need all the steps and can rearrange them, but use the following guidelines when doing so:

- For elements like DSCP maps, meters and RED curves, you must create the element before you can use it in a traffic class or policy. If desired you can **create** the traffic class or policy first and then **set** it to use the appropriate element.
- For elements like classifiers, DAR objects, traffic classes and policies, you must create the element before you can add it to the element above it in the hierarchy.
- You must combine the elements into a hierarchy from the top down. Add traffic classes to the policy, then subclasses to the traffic classes if required, then classifiers and/or DAR objects to the traffic classes or subclasses.

Table 39-4: Overall procedure for configuring a software QoS hierarchy

Step	Command	Action
1	create classifier=0..9999 *options* **create sqos dar**=*id-list* [codec={audio|video|any}] [description=*description*] [dstip={*ipadd*[/0..32]|*ipv6add*[/0..128]}] [h323port=1..65535] [inactivetimeout={1..60|none}] [protocol={sip|rtsp|h323|all}] [rtspport=1..65535] [sipport=1..65535] [srcip={*ipadd*[/0..32]|*ipv6add*[/0..128]}]	Create static and/or dynamic classifiers to sort traffic into flows. For more information about static classifiers, see Chapter 38, Generic Packet Classifier. For more information about dynamic classifiers, see "Configuring DAR for Voice and Video Traffic" on page 39-47.
2	**create sqos meter**=*id-list* [description=*description*] [ignorebwclass3={yes|no}] [minbandwidth=*rate*[kbps|mbps|gbps]] [maxbandwidth=*rate*[kbps|mbps|gbps]] [minburstsize=*burstsize*[bytes|kbytes|mbytes|gbytes]] [maxburstsize=*burstsize*[bytes|kbytes|mbytes|gbytes]] [type={srtcm|trtcm}]	If required, create meters. Meters determine the bandwidth used by the packet and how conformant it is. For more information, see "Metering" on page 39-37.
3	**create sqos dscpmap**=*id-list* [description=*description*] **set sqos dscpmap**=*id-list* table=premark [description=*description*] [dscp=*dscp-list*] [newbwclass=1..3] [newdscp=0..63] **set sqos dscpmap**=*id-list* table=remark [description=*description*] [bwclass=*bwclass-list*] [dscp=*dscp-list*] [newbwclass=1..3] [newdscp=0..63]	If required, create DSCP maps and configure the tables in them. For each policy, the router can use the tables in the policy DSCP map to set the DSCP bits and/or bandwidth class. The premarking table applies before metering, and the remarking table applies after. For more information, see "Premarking" on page 39-36 and "Re-Marking" on page 39-41.

Table 39-4: Overall procedure for configuring a software QoS hierarchy (cont.)

Step	Command	Action
4	**create sqos red**=*id-list* [description=*description*] [start1=0..100] [stop1=0..100] [drop1=0..100] [start2=0..100] [stop2=0..100] [drop2=0..100] [start3=0..100] [stop3=0..100] [drop3=0..100]	If required, create extra RED curves. RED curves allow early dropping of TCP packets to slow and smooth a congested TCP flow. For more information, see "RED" on page 39-40.
5	**create sqos policy**=*id-list* [bwclass3action={drop|pause|none}] [description=*description*] [dscpmap={0..9999|none}] [meter={0..9999|none}] [pauseaction={none|log|trap|both}] [pausetime={1..30}] [remarking={0..63|usedscpmap|none}] [remarkvlanpri={0..7|none}] [systemtraffic={5..50}] [virtbw={*bandwidth*[kbps|mbps|gbps]|none}] [weightscheduler={wrr|dwrr}]	Create a policy for each interface or tunnel, and specify QoS processing parameters if required. The policy determines the QoS processing for each flow on the interface. You can set QoS processing parameters on the policy or on the traffic classes attached to it. You can use the same policy on multiple interfaces or tunnels if they have sufficiently similar traffic flows.
6	**create sqos trafficclass**=*id-list* [bwclass3action={drop|pause|none}] [description=*description*] [maxqlen=1..1023] [meter={0..9999|none}] [pauseaction={none|log|trap|both}] [pausetime={1..30}] [premarkbwcl={1..3|usedscpmap}] [premarkdscp={0..63|usedscpmap|none}] [{priority=0..15|weight=0..100}] [qlimitexceedaction={none|log|trap|both}] [queuedrop={head|tail}] [red={0..9999|none}] [remarking=0..63|usedscpmap|none}] [remarkvlanpri={0..7|none}] [virtbw={*bandwidth*[kbps|mbps|gbps]|none}] [weightscheduler={wrr|dwrr}]	Create traffic classes. Traffic classes group similar traffic flows together and specify the QoS actions taken on them. Each traffic class contains an egress queue.
7	**add sqos interface dar** add sqos interface=*interface* dar=*id-list*	If configuring DAR, add DAR objects to interfaces.
8	**add sqos policy trafficclass** add sqos policy=0..9999 trafficclass=*id-list*	Add traffic classes to policies.
9	**add sqos trafficclass subclass** add sqos trafficclass=0..9999 subclass=*id-list*	Add sub traffic classes to traffic classes if you need a multi-level traffic class tree. Continue until you have built up the traffic class tree for each policy.
10	**add sqos trafficclass classifier** add sqos trafficclass=0..9999 classifier=*id-list* or **add sqos trafficclass dar** add sqos trafficclass=0..9999 dar=*id-list*	Add static and/or dynamic classifiers to leaf traffic classes. You can use up to 64 classifiers per policy. Both static classifiers and the dynamic classifiers created by DAR objects count towards this quantity.

Table 39-4: Overall procedure for configuring a software QoS hierarchy (cont.)

Step	Command	Action
11	**set sqos interface**=*interface* [inpolicy={0..9999|none}] [outpolicy={0..9999|none}] [tunnelpolicy={0..9999|none}]	Apply policies to interfaces.
12	**enable sqos**	Enable software QoS.

Default Traffic Class

By default, the router attaches a non-configurable Default Traffic Class (DTC) to each new policy, for unclassified traffic. The settings of this DTC are:

- queuedrop= tail
- weight=0
- maxqlen=64

To configure a DTC with different settings, follow the instructions in Table 39-5.

Table 39-5: Procedure for configuring a default traffic class

Step	Command	Action
1	**create sqos trafficclass**=0..9999 weight=0..20 [*options*]	Create a new traffic class with the desired parameters.
2	**create sqos policy**=0..9999 defaulttrafficclass=0..9999 **set sqos policy**=0..9999 defaulttrafficclass=0..9999	Make the new traffic class the default traffic class for the policy.

Configuring QoS Stages

This section describes how to configure each of the stages of a software QoS solution. Figure 39-10 on page 39-19 shows the stages.

Premarking

Premarking is performed at the traffic class level. Options are:

- Directly specifying a new bandwidth class and/or DSCP for all packets that belong to the traffic class. This is the approach you need for an AF domain. Use one of the commands:

```
create sqos trafficclass=id-list premarkbwcl=1..3
   premarkdscp=0..63 [other-options]

set sqos trafficclass=id-list premarkbwcl=1..3
   premarkdscp=0..63 [other-options]
```

- Using the policy's DSCP map, with the packet's current DSCP as an index into the table. Follow the instructions in Table 39-6.

The command order in the table is one of several possible orders. See "Configuration order" on page 39-33 for more information.

Premarking with a DSCP map

Table 39-6: Procedure for configuring premarking using a DSCP map

Step	Command	Action
1	**create sqos dscpmap**=*id-list* [description=*description*]	Create the DSCP map.
2	**set sqos dscpmap**=*id-list* table=premark [dscp=*dscp-list*] newbwclass=1..3 newdscp=0..63	Configure the map's premarking table. For each incoming **dscp** that you want to change, specify a **newdscp**. For each incoming **dscp** that you want to assign to a particular bandwidth class, specify the **newbwclass**.
3	**create sqos policy**=*id-list* dscpmap=0..9999 [*other-options*]	Create the policy and specify the DSCP map.
4	**create sqos trafficclass**=*id-list* [premarkbwcl=usedscpmap] [premarkdscp=usedscpmap] [*other-options*]	Create the traffic class and specify premarking.
5	**add sqos policy trafficclass** add sqos policy=0..9999 trafficclass=*id-list*	Add the traffic class to the policy.
6	create classifier **create sqos dar** **create sqos meter** **create sqos red**	Create the remaining QoS elements as required. For configuration details, see Table 39-4 on page 39-33.
7	**add sqos interface dar** **add sqos trafficclass subclass** **add sqos trafficclass classifier** **add sqos trafficclass dar**	Build your QoS elements into a QoS hierarchy. For configuration details, see Table 39-4 on page 39-33.

Table 39-6: Procedure for configuring premarking using a DSCP map (cont.)

Step	Command	Action
8	**set sqos interface** **enable sqos**	Apply the policy to the required interface and enable software QoS. For configuration details, see Table 39-4 on page 39-33.
9	**show sqos dscpmap** **show sqos policy** **show sqos trafficclass**	Check the configuration.

Metering

Meters measure the bandwidth conformance of packets. You have a choice of meters:

- Single Rate Three Colour Marker (Table 39-7).
- Two Rate Three Colour Marker (Table 39-8).

The most common use of metering is to determine which packets are non-conformant and drop them. When you meter on traffic classes in which TCP flows are prevalent, and drop non-conformant packets, you need to choose burst sizes carefully. If burst sizes are too low, packets are dropped from flows that exceed their guaranteed bandwidth by small amounts. TCP flows react drastically to dropped packets—for example, they may halve the sending rate. This reduces the total TCP rate significantly and may stop the flows from getting their guaranteed bandwidths.

Burst sizes should be in proportion to the minimum and maximum rates, so if you increase the rates, also increase the burst sizes.

The command order in the tables is one of several possible orders. See "Configuration order" on page 39-33 for more information.

Single rate meter

Table 39-7: Procedure for creating and using a Single Rate Three Colour Marker meter

Step	Command	Action
1	**create sqos meter**=*id-list* [description=*description*] [ignorebwclass3={yes|no}] [maxbandwidth=*rate*[kbps|mbps|gbps]] [minburstsize=*burstsize*[bytes|kbytes| mbytes|gbytes]] [maxburstsize=*burstsize*[bytes|kbytes| mbytes|gbytes]]	Create the meter. The default meter type is single rate.
2	**create sqos policy**=*id-list* meter=0..9999 [bwclass3action={drop|pause|none}] [pauseaction={none|log|trap|both}] [pausetime={1..30}] [*other-options*] or **create sqos trafficclass**=*id-list* meter=0..9999 [bwclass3action={drop|pause|none}] [pauseaction={none|log|trap|both}] [pausetime={1..30}] [*other-options*]	Create the policy or traffic class, and specify the meter. You can also specify that the router drop non-conformant packets or pause that flow. **Drop** discards the packet. **Pause** discards the packet and stops dequeuing packets from the flow for **pausetime** seconds. The router can generate log messages and SNMP traps when it pauses a flow.
3	create classifier **create sqos dar** **create sqos red** **create sqos dscpmap** **set sqos dscpmap**	Create the remaining QoS elements as required. For configuration details, see Table 39-4 on page 39-33.
4	**add sqos interface dar** **add sqos policy trafficclass** **add sqos trafficclass subclass** **add sqos trafficclass classifier** **add sqos trafficclass dar**	Build your QoS elements into a QoS hierarchy. For configuration details, see Table 39-4 on page 39-33.
5	**set sqos interface** **enable sqos**	Apply the policy to the required interface and enable software QoS. For configuration details, see Table 39-4 on page 39-33.
6	**show sqos meter** **show sqos policy** **show sqos trafficclass**	Check the configuration.

Two rate meter

Table 39-8: Procedure for creating and using a Two Rate Three Colour Marker meter

Step	Command	Action
1	**create sqos meter**=*id-list* type=trtcm [description=*description*] [ignorebwclass3={yes|no}] [maxbandwidth=*rate*[kbps|mbps|gbps]] [maxburstsize=*burstsize*[bytes|kbytes| mbytes|gbytes]] [minbandwidth=*rate*[kbps|mbps|gbps]] [minburstsize=*burstsize*[bytes|kbytes| mbytes|gbytes]]	Create the meter, specifying that it is a two rate meter.
2	**create sqos policy**=*id-list* meter=0..9999 [bwclass3action={drop|pause|none}] [pauseaction={none|log|trap|both}] [pausetime={1..30}] [*other-options*] **create sqos trafficclass**=*id-list* meter=0..9999 [bwclass3action={drop|pause|none}] [pauseaction={none|log|trap|both}] [pausetime={1..30}] [*other-options*]	Create the policy or traffic class, and specify the meter. You can also specify that the router drop non-conformant packets or pause that flow. **Drop** discards the packet. **Pause** discards the packet and stops dequeuing packets from the flow for **pausetime** seconds. The router can generate log messages and SNMP traps when it pauses a flow.
3	create classifier **create sqos dar** **create sqos red** **create sqos dscpmap** **set sqos dscpmap**	Create the remaining QoS elements as required. For configuration details, see Table 39-4 on page 39-33.
4	**add sqos interface dar** **add sqos policy trafficclass** **add sqos trafficclass subclass** **add sqos trafficclass classifier** **add sqos trafficclass dar**	Build your QoS elements into a QoS hierarchy. For configuration details, see Table 39-4 on page 39-33.
5	**set sqos interface** **enable sqos**	Apply the policy to the required interface and enable software QoS. For configuration details, see Table 39-4 on page 39-33.
6	**show sqos meter** **show sqos policy** **show sqos trafficclass**	Check the configuration.

RED

RED curve sets allow gradual dropping of TCP packets as the traffic class queue becomes congested. Packets are dropped instead of being enqueued. You can:

- Use one of the default RED curve sets (Table 39-9).
- Create another RED curve set for your particular requirements (Table 39-10).

The command order in the tables is one of several possible orders. See "Configuration order" on page 39-33 for more information.

Using default RED curve sets

Table 39-9: Procedure for using one of the default RED curve sets

Step	Command	Action
1	**create sqos trafficclass**=*id-list* red=0..2 [maxqlen=1..1023] [queuedrop={head|tail}] [*other-options*]	Create the traffic class, and specify the RED curve set. Only use RED on leaf traffic classes. RED curve **0** is aggressive and starts dropping packets early. RED curve **1** is medium and starts dropping packets later. RED curve **2** is passive and only drops packets when the queue is almost full. You can also specify the queue length and whether to tail or head drop. The default is tail drop.
2	create classifier **create sqos dar** **create sqos policy** **create sqos meter** **create sqos dscpmap** **set sqos dscpmap**	Create the remaining QoS elements as required. For configuration details, see Table 39-4 on page 39-33.
3	**add sqos interface dar** **add sqos policy trafficclass** **add sqos trafficclass subclass** **add sqos trafficclass classifier** **add sqos trafficclass dar**	Build your QoS elements into a QoS hierarchy. For configuration details, see Table 39-4 on page 39-33.
4	**set sqos interface** **enable sqos**	Apply the policy to the required interface and enable software QoS. For configuration details, see Table 39-4 on page 39-33.
5	**show sqos red** **show sqos trafficclass**	Check the configuration.

Creating new RED curve sets

Table 39-10: Procedure for creating and using a new RED curve set

Step	Command	Action
1	**create sqos red**=*id-list* [description=*description*] [start1=0..100] [stop1=0..100] [drop1=0..100] [start2=0..100] [stop2=0..100] [drop2=0..100] [start3=0..100] [stop3=0..100] [drop3=0..100]	Create the RED curve.
2	**create sqos trafficclass**=*id-list* red=3..9999 [maxqlen=1..1023] [queuedrop={head|tail}] [*other-options*]	Create the traffic class, and specify the RED curve set Only use RED on leaf traffic classes. You can also specify the queue length and whether to tail or head drop. The default is tail drop.
3	create classifier **create sqos dar** **create sqos policy** **create sqos meter** **create sqos dscpmap** **set sqos dscpmap**	Create the remaining QoS elements as required. For configuration details, see Table 39-4 on page 39-33.
4	**add sqos interface dar** **add sqos policy trafficclass** **add sqos trafficclass subclass** **add sqos trafficclass classifier** **add sqos trafficclass dar**	Build your QoS elements into a QoS hierarchy. For configuration details, see Table 39-4 on page 39-33.
5	**set sqos interface** **enable sqos**	Apply the policy to the required interface and enable software QoS. For configuration details, see Table 39-4 on page 39-33.
6	**show sqos red** **show sqos trafficclass**	Check the configuration.

Re-Marking

Re-marking is performed at the traffic class or policy level. You can replace any combination of DSCP, VLAN priority and bandwidth class. Options are:

- Directly specifying a new DSCP for packets belonging to a traffic class or policy, using one of the commands:

  ```
  create sqos trafficclass=id-list remarking=0..63
  ```

  ```
  create sqos policy=id-list remarking=0..63
  ```

- Using the policy's DSCP map, with the packet's metered bandwidth class and current DSCP as an index into the table. Follow the instructions in Table 39-11.
- Directly specifying a new VLAN priority for packets belonging to a traffic class or policy, using one of the commands:

  ```
  create sqos trafficclass=id-list remarkvlanpri=0..7
  ```

  ```
  create sqos policy=id-list remarkvlanpri=0..7
  ```

The command order in the table is one of several possible orders. See "Configuration order" on page 39-33 for more information.

Re-marking with a DSCP map

Table 39-11: Procedure for configuring re-marking using a DSCP map

Step	Command	Action
1	**create sqos dscpmap**=*id-list* [description=*description*]	Create the DSCP map.
2	**set sqos dscpmap**=*id-list* table=remark [bwclass=*bwclass-list*] [dscp=*dscp-list*] newbwclass=1..3 newdscp=0..63	Configure the map's re-marking table. For each combination of **dscp** and bandwidth class (**bwclass**) that you want to change, specify a **newdscp** and/or **newbwclass**.
3	**create sqos meter**=*id-list* [description=*description*] [ignorebwclass3={yes|no}] [maxbandwidth=*rate*[kbps|mbps|gbps]] [maxburstsize=*burstsize*[bytes|kbytes|mbytes|gbytes]] [minbandwidth=*rate*[kbps|mbps|gbps]] [minburstsize=*burstsize*[bytes|kbytes|mbytes|gbytes]] [type={srtcm|trtcm}]	Create a meter to determine bandwidth conformance. For configuration details, see "Metering" on page 39-37.
4	**create sqos policy**=*id-list* dscpmap=0..9999 [meter=0..9999] [remarking=usedscpmap] [*other-options*]	Create the policy and specify the DSCP map. If you want traffic from all traffic classes in the policy to be re-marked in the same way, specify remarking as part of the policy. If you want to use one meter for the whole policy, specify the meter as part of the policy.
5	**create sqos trafficclass**=*id-list* [remarking=usedscpmap] [meter=0..9999] [*other-options*]	Create the traffic class. If you want traffic from different traffic classes in the policy to be remarked differently, specify re-marking as part of the traffic class. If you want to use different meters for different traffic classes in the policy, specify the meter as part of traffic class.
6	**add sqos policy trafficclass** add sqos policy=0..9999 trafficclass=*id-list*	Add the traffic class to the policy.
7	create classifier **create sqos dar** **create sqos red**	Create the remaining QoS elements as required. For configuration details, see Table 39-4 on page 39-33.
8	**add sqos interface dar** **add sqos trafficclass subclass** **add sqos trafficclass classifier** **add sqos trafficclass dar**	Build your QoS elements into a QoS hierarchy. For configuration details, see Table 39-4 on page 39-33.
9	**set sqos interface** **enable sqos**	Apply the policy to the required interface and enable software QoS. For configuration details, see Table 39-4 on page 39-33.

Table 39-11: Procedure for configuring re-marking using a DSCP map (cont.)

Step	Command	Action
10	**show sqos dscpmap** **show sqos policy** **show sqos trafficclass**	Check the configuration.

Queue Scheduling

The queue scheduling mechanism determines the order in which the router empties traffic class queues, and therefore which packets it sends out.

For this method...	See the procedure in...
Priority Queuing (PQ)	Table 39-12
Weighted Round Robin (WRR) Deficit Weighted Round Robin (DWRR)	Table 39-13
Mixed scheduling	Table 39-14

For details about scheduling methods along with suggestions, see "Queue Scheduling" on page 39-29.

The command order in the tables is one of several possible orders. See "Configuration order" on page 39-33 for more information.

Priority queuing

Table 39-12: Procedure for configuring priority queuing

Step	Command	Action
1	**create sqos trafficclass**=*id-list* priority=0..15 [*other-options*]	Create the required traffic classes, giving each a priority. The higher the number, the higher the priority.
2	**create sqos policy**=0..9999 [systemtraffic={5..50}] [weightscheduler={wrr|dwrr}] [*other-options*]	Create the policy. If necessary, specify the proportion of bandwidth that the policy allows for system traffic. By default, the policy uses WRR to schedule the system, root and default traffic classes, which are all weighted classes. If required, change to DWRR.
3	**create sqos trafficclass**=0..9999 weight=0..100 [*other-options*] **set sqos policy**=0..9999 defaulttrafficclass=0..9999	If necessary, change the proportion of bandwidth that the policy allows for default traffic. First create a new traffic class with the desired weighting, then make the new traffic class the default traffic class for the policy.
4	**add sqos policy trafficclass** add sqos policy=0..9999 trafficclass=*id-list*	Add the traffic class to the policy.

Table 39-12: Procedure for configuring priority queuing (cont.)

Step	Command	Action
5	create classifier **create sqos dar** **create sqos meter** **create sqos red** **create sqos dscpmap** **set sqos dscpmap**	Create the remaining QoS elements as required. For configuration details, see Table 39-4 on page 39-33.
6	**add sqos interface dar** **add sqos trafficclass subclass** **add sqos trafficclass classifier** **add sqos trafficclass dar**	Build your QoS elements into a QoS hierarchy. For configuration details, see Table 39-4 on page 39-33.
7	**set sqos interface** **enable sqos**	Apply the policy to the required interface and enable software QoS. For configuration details, see Table 39-4 on page 39-33.
8	**show sqos dscpmap** **show sqos policy** **show sqos trafficclass**	Check the configuration.

WRR and DWRR queuing

Table 39-13: Procedure for configuring WRR or DWRR queue scheduling

Step	Command	Action
1	**create sqos trafficclass**=*id-list* weight=0..100 [*other-options*]	Create the required traffic classes, giving each a weight. The higher the number, the higher the proportion of bandwidth allocated to the traffic class. Weights need not total 100%. When they do not, normalised weights are used.
2	**create sqos trafficclass**=*id-list* [weightscheduler={wrr|dwrr}] [*other-options*]	If you are building a multi-level traffic class tree, create intermediate traffic classes and attach the weighted traffic classes to them. By default, the intermediate traffic class uses WRR to schedule the weighted traffic classes. If required, change to DWRR.
3	**create sqos policy**=0..9999 [systemtraffic={5..50}] [weightscheduler={wrr|dwrr}] [*other-options*]	Create the policy. If necessary, specify the proportion of bandwidth that the policy allows for system traffic. By default, the intermediate traffic class uses WRR to schedule the weighted traffic classes, including the system and default traffic classes. If required, change to DWRR.

Table 39-13: Procedure for configuring WRR or DWRR queue scheduling (cont.)

Step	Command	Action
4	**create sqos trafficclass**=0..9999 weight=0..100 [*other-options*] **set sqos policy**=0..9999 defaulttrafficclass=0..9999	If necessary, change the proportion of bandwidth that the policy allows for default traffic. First create a new traffic class with the desired weighting, then make the new traffic class the default traffic class for the policy.
5	**add sqos policy trafficclass** add sqos policy=0..9999 trafficclass=*id-list*	Add the traffic class to the policy.
6	create classifier **create sqos dar** **create sqos meter** **create sqos red** **create sqos dscpmap** **set sqos dscpmap**	Create the remaining QoS elements as required. For configuration details, see Table 39-4 on page 39-33.
7	**add sqos interface dar** **add sqos trafficclass subclass** **add sqos trafficclass classifier** **add sqos trafficclass dar**	Build your QoS elements into a QoS hierarchy. For configuration details, see Table 39-4 on page 39-33.
8	**set sqos interface** **enable sqos**	Apply the policy to the required interface and enable software QoS. For configuration details, see Table 39-4 on page 39-33.
9	**show sqos dscpmap** **show sqos policy** **show sqos trafficclass**	Check the configuration.

Mixed scheduling

Table 39-14: Procedure for configuring mixed scheduling

Step	Command	Action
1	**create sqos trafficclass**=*id-list* priority=0..15 [*other-options*]	Create the required priority traffic classes. The higher the priority number, the higher the priority.
1	**create sqos trafficclass**=*id-list* weight=0..100 [*other-options*]	Create the required weighted traffic classes. The higher the weight number, the higher the proportion of bandwidth allocated to the traffic class. Weights do not have to total 100%. If they do not, normalised weights are used.
2	**create sqos trafficclass**=*id-list* [weightscheduler={wrr|dwrr}] [*other-options*]	If you are building a multi-level traffic class tree, create intermediate traffic classes and attach the traffic classes to them. By default, the intermediate traffic class uses WRR to schedule the weighted traffic classes. If required, change to DWRR.

Table 39-14: Procedure for configuring mixed scheduling (cont.)

Step	Command	Action
3	**create sqos policy**=0..9999 [systemtraffic={5..50}] [weightscheduler={wrr|dwrr}] [*other-options*]	Create the policy. If necessary, specify the proportion of bandwidth that the policy allows for system traffic. By default, the policy uses WRR to schedule the weighted traffic classes that are attached to the root traffic class. If required, change to DWRR.
4	**create sqos trafficclass**=0..9999 weight=0..100 [*other-options*] **set sqos policy**=0..9999 defaulttrafficclass=0..9999	If necessary, change the proportion of bandwidth that the policy allows for default traffic. First create a new traffic class with the desired weighting, then make the new traffic class the default traffic class for the policy.
5	**add sqos policy trafficclass** add sqos policy=0..9999 trafficclass=*id-list*	Add the traffic class to the policy.
6	create classifier **create sqos dar** **create sqos meter** **create sqos red** **create sqos dscpmap** **set sqos dscpmap**	Create the remaining QoS elements as required. For configuration details, see Table 39-4 on page 39-33.
7	**add sqos interface dar** **add sqos trafficclass subclass** **add sqos trafficclass classifier** **add sqos trafficclass dar**	Build your QoS elements into a QoS hierarchy. For configuration details, see Table 39-4 on page 39-33.
8	**set sqos interface** **enable sqos**	Apply the policy to the required interface and enable software QoS. For configuration details, see Table 39-4 on page 39-33.
9	**show sqos dscpmap** **show sqos policy** **show sqos trafficclass**	Check the configuration.

Configuring DAR for Voice and Video Traffic

Configuring Dynamic Application Recognition (DAR) involves creating the required DAR objects, creating the rest of the software QoS policy, and applying the policy and DAR objects to the appropriate interfaces.

On a slow interface that carries voice (VoIP) traffic, you must also force large (non-voice) packets to be fragmented by setting a low interface MTU (maximum transmission unit), such as 256 bytes. This stops large data packets from delaying the voice packets, which are small. For example, a 1500 byte packet takes at least 190 milliseconds to send over a 64 kbps link. Acceptable total end-to-end latency for VoIP packets is only 150 ms.

On an interface that carries both VoIP and video traffic, it may or may not be desirable to configure a low MTU. This is because a low MTU forces the fragmentation of the video packets, and so cause significant overhead in the processing of the video stream. You may need to tune the MTU value to get a good balance between the latency in the VoIP and the CPU load induced by fragmenting the video packets.

We suggest you put latency and jitter-sensitive traffic, such as VoIP, into a high priority traffic class, rather than a weighted class. You should also plan your traffic class tree so that the VoIP traffic class is not a child of a weighted traffic class (except for the root class).

For an example, see Configuration Example "2: Guaranteeing VoIP Traffic using DAR" on page 39-68.

Table 39-15: Procedure for configuring Dynamic Application Recognition for VoIP and video traffic

Step	Commands	Action
1	**create sqos dar**=*id-list* [codec={audio|video|any}] [description=*description*] [dstip={*ipadd*[/0..32]|*ipv6add*[/0..128]}] [srcip={*ipadd*[/0..32]|*ipv6add*[/0..128]}] [inactivetimeout={1..3600|none}] [protocol={sip|rtsp|h323|all}] [h323port=1..65535] [rtspport=1..65535] [sipport=1..65535]	Create the DAR object. If necessary, limit it to matching packets with particular codec, protocol or IP settings.
2	**add sqos interface dar** add sqos interface=*interface* dar=*id-list*	Add the DAR object to the interface that voice or video session initiation messages are received on.

Table 39-15: Procedure for configuring Dynamic Application Recognition for VoIP and video traffic (cont.)

Step	Commands	Action
3	**create sqos policy**=*id-list* [bwclass3action={drop|pause|none}] [description=*description*] [dscpmap={0..9999|none}] [meter={0..9999|none}] [pauseaction={none|log|trap|both}] [pausetime={1..30}] [remarking={0..63|usedscpmap|none}] [remarkvlanpri={0..7|none}] [systemtraffic={5..50}] [virtbw={*bandwidth*[kbps|mbps|gbps]|none}] [weightscheduler={wrr|dwrr}]	Create a policy for the interface or tunnel through which voice or video traffic egresses, and specify QoS processing parameters if required.
4	**create sqos trafficclass**=*id-list* [bwclass3action={drop|pause|none}] [description=*description*][maxqlen=1..1023] [meter={0..9999|none}] [pauseaction={none|log|trap|both}] [pausetime={1..30}] [premarkbwcl={1..3|usedscpmap}] [premarkdscp={0..63|usedscpmap|none}] [{priority=0..15|weight=0..100}] [qlimitexceedaction={none|log|trap|both}] [queuedrop={head|tail}] [red={0..9999|none}] [remarking=0..63|usedscpmap|none}] [remarkvlanpri={0..7|none}] [virtbw={*bandwidth*[kbps|mbps|gbps]|none}] [weightscheduler={wrr|dwrr}]	Create at least one traffic class, and specify QoS processing parameters as required. Traffic classes group similar traffic flows together. Each traffic class contains an egress queue. VoIP traffic should go into a high-priority traffic class, not a weighted class.
5	**add sqos policy trafficclass** add sqos policy=0..9999 trafficclass=*id-list*	Add the traffic classes to the policy.
6	**add sqos trafficclass subclass** add sqos trafficclass=0..9999 subclass=*id-list*	If required, add sub traffic classes to the traffic classes. Continue until you have built up the traffic class tree for the policy.
7	**add sqos trafficclass dar** add sqos trafficclass=0..9999 dar=*id-list*	Add the DAR object to the appropriate leaf traffic class.
8	**set sqos interface**=*interface* outpolicy=0..9999 **set sqos interface**=*interface* tunnelpolicy=0..9999	Apply the policy to the interface or tunnel through which voice or video traffic egresses.
9	**enable sqos**	Enable software QoS.
10	**set interface mtu** set interface=*interface* mtu=256	If the interface is slow and carries voice traffic, which is sensitive to latency, force large packets to be fragmented.

Configuring Software QoS on Specific Interfaces

This section describes how to configure a software QoS solution on PPP, PPPoE, and Frame Relay interfaces, and the switch instance. The biggest difference between the interfaces is the valid classifier options.

PPP and PPPoE

For PPP interfaces over:

- a synchronous port (SYN*n*)
- an ISDN call (ISDN-*callname*)
- an ACC call (ACC-*callname*)
- a MIOX circuit (MIOX*n-circuitname*)
- a TDM group (TDM-*groupname*)
- an L2TP call (TNL-*callname*)
- an ATM channel (for example, atm0.1)

configure Software QoS on the PPP interface. Follow the instructions in Table 39-16.

For PPPoE interfaces over Ethernet ports (ETH*n-servicename)*, configure software QoS on the Ethernet port. Follow the instructions in Table 39-17.

For PPPoE interfaces over VLANs (VLAN*n-servicename*), configure software QoS on swi0. Follow the instructions in Table 39-19 on page 39-54. Use the **dvlan** parameter in the classifier to identify packets for each VLAN if required.

Table 39-16: Procedure for configuring software QoS on PPP interfaces

Step	Commands	Action
1		Configure the underlying physical interface as required. See Chapter 9, Interfaces or the chapter for the physical interface that you are using.
2	create ppp=*ppp-interface* over=*physical-interface* [*other-options*]	Create and configure the PPP interface. See Chapter 15, Point-to-Point Protocol (PPP).
3	create classifier=1..9999 [iinterface=*interface*] [iport=*port*] [pppprotocolid={*ppp-protocol-id*|ip|ipv6|any}] [ipdaddr={*ipadd*[/0..32]|*ipv6add*[/0..128]|any}] [ipsaddr={*ipadd*[/0..32]|*ipv6add*[/0..128]|any}] [ipdscp={*dscp-list*|any}] [iptos={0..7|any}] [ipfrag={yes|no|any}] [ipoptions={yes|no|any}] [ipflowlabel={0..1048575|any}] [ipprotocol={tcp|udp|icmp|igmp|ospf|nontcpudp|any|*ip-protocol*} [icmptype={any|echorply|unreachable|quench|redirect|echo|advertisement|solicitation|timeexceed|parameter|tstamp|tstamprply|inforeq|inforep|addrreq|addrrep|namereq|namerply|*icmp-type*}] [icmpcode={any|filter|fragment|fragreassm|hostcomm|hostisolated|hostprec|hostredirect|hostrtos|hosttos|hostunknown|hostunreach|netcomm|netredirect|netrtos|nettos|netunknown|netunreach|noptr|portunreach|precedent|protunreach|ptrproblem|sourceroute|ttl|*icmp-code*}] [tcpflags={{urg|ack|rst|syn|fin}[,...]|any}] [tcpdport={*port-range*|any}] [tcpsport={*port-range*|any}] [udpdport={*port-range*|any}] [udpsport={*port-range*|any}]	Create classifiers. Valid parameters include: ● ingress interface or port, for egress QoS only ● PPP protocol ID ● Layer 3 ● Layer 4
4		Create the QoS policy and its underlying hierarchy. See Table 39-4 on page 39-33 for details.
5	**set sqos interface**=*ppp-interface* inpolicy=0..9999 and/or **set sqos interface**=*ppp-interface* outpolicy=0..9999	Attach the policy to the PPP interface.
6	**enable sqos**	Enable software QoS.
7	**set interface mtu** set interface=*interface* mtu=256	If the interface carries traffic that is sensitive to latency, such as voice traffic, force large packets to be fragmented.

Table 39-17: Procedure for configuring software QoS on PPP over Ethernet interfaces

Step	Commands	Action
1	create ppp=*ppp-interface* over=eth*n-servicename* [*other-options*]	Create and configure the PPP interface. See Chapter 15, Point-to-Point Protocol (PPP).
2	create classifier=1..9999 [pppindex=0..1023] [iinterface=*interface*] [iport=*port*] [svlan={*vlan-name*|1..4094|any}] [dvlan={*vlan-name*|1..4094|any}] [vlanpriority={*priority-list*|any}] [ethformat={802.2|ethii|netwareraw|snap|any}] [macdaddr={*macadd*|any}] [macsaddr={*macadd*|any}] [mactype={l2ucast|l2bmcast|any}] [protocol={*protocol-type*|arp|ip|ipv6|ipx|any}] [ipdaddr={*ipadd*[/0..32]|*ipv6add*[/0..128]|any}] [ipsaddr={*ipadd*[/0..32]|*ipv6add*[/0..128]|any}] [ipdscp={*dscp-list*|any}] [iptos={0..7|any}] [ipfrag={yes|no|any}] [ipoptions={yes|no|any}] [ipflowlabel={0..1048575|any}] [ipprotocol={tcp|udp|icmp|igmp|ospf| nontcpudp|any|*ip-protocol*} [icmptype={any|echorply|unreachable|quench| redirect|echo|advertisement|solicitation| timeexceed|parameter|tstamp|tstamprply| inforeq|inforep|addrreq|addrrep|namereq| namerply|*icmp-type*}] [icmpcode={any|filter|fragment|fragreassm| hostcomm|hostisolated|hostprec|hostredirect| hostrtos|hosttos|hostunknown|hostunreach| netcomm|netredirect|netrtos|nettos| netunknown|netunreach|noptr|portunreach| precedent|protunreach|ptrproblem|sourceroute| ttl|*icmp-code*}] [tcpflags={{urg|ack|rst|syn|fin}[,...]|any}] [tcpdport={*port-range*|any}] [tcpsport={*port-range*|any}] [udpdport={*port-range*|any}] [udpsport={*port-range*|any}]	Create classifiers. Valid parameters include: ● PPP index. This lets you separate different PPP interfaces over one Ethernet interface. ● ingress interface or port, for egress QoS only ● VLAN settings ● Ethernet settings ● Layer 3 ● Layer 4
3		Create the QoS policy and its underlying hierarchy. See Table 39-4 on page 39-33 for details.
4	**set sqos interface**=*eth-interface* inpolicy=0..9999 and/or **set sqos interface**=*eth-interface* outpolicy=0..9999	Attach the policy to the underlying Ethernet interface.
5	**enable sqos**	Enable software QoS.
6	**set interface mtu** set interface=*interface* mtu=256	If the interface carries traffic that is sensitive to latency, such as voice traffic, force large packets to be fragmented.

Frame Relay

Software QoS treats each frame relay interface as a single ingress or egress interface with a single policy. If you need to control the quality of service given to traffic on individual DLCs, classify packets into different traffic classes according to DLCI and specify the appropriate QoS controls in each traffic class.

Table 39-18: Procedure for configuring software QoS on frame relay interfaces

Step	Commands	Action
1		If necessary, configure the underlying physical interface. See Chapter 9, Interfaces or the chapter for the physical interface that you are using.
2	create framerelay=*fr-interface* over=*physical-interface* [*other-options*]	Create and configure the frame relay interface. See Chapter 14, Frame Relay.
3	create classifier=1..9999 [iinterface=*interface*] [eport=*port*] [iport=*port*] [dlci={*dlci-range*|any}] [protocol={*protocol-type*|arp|ip|ipv6|ipx|any}] [ipdaddr={*ipadd*[/0..32]|*ipv6add*[/0..128]|any}] [ipsaddr={*ipadd*[/0..32]|*ipv6add*[/0..128]|any}] [ipdscp={*dscp-list*|any}] [iptos={0..7|any}] [ipfrag={yes|no|any}] [ipoptions={yes|no|any}] [ipflowlabel={0..1048575|any}] [ipprotocol={tcp|udp|icmp|igmp|ospf| nontcpudp|any|*ip-protocol*}] [icmptype={any|echorply|unreachable|quench| redirect|echo|advertisement|solicitation| timeexceed|parameter|tstamp|tstamprply| inforeq|inforep|addrreq|addrrep|namereq| namerply|*icmp-type*}] [icmpcode={any|filter|fragment|fragreassm| hostcomm|hostisolated|hostprec|hostredirect| hostrtos|hosttos|hostunknown|hostunreach| netcomm|netredirect|netrtos|nettos| netunknown|netunreach|noptr|portunreach| precedent|protunreach|ptrproblem|sourceroute| ttl|*icmp-code*}] [tcpflags={{urg|ack|rst|syn|fin}[,...]|any}] [tcpdport={*port-range*|any}] [tcpsport={*port-range*|any}] [udpdport={*port-range*|any}] [udpsport={*port-range*|any}]	Create classifiers. Valid parameters include: ● interface and port ● DLCI ● Layer 3 ● Layer 4
4		Create the QoS policy and its underlying hierarchy. See Table 39-4 on page 39-33 for details.
5	**set sqos interface**=*ppp-interface* inpolicy=0..9999 and/or **set sqos interface**=*ppp-interface* outpolicy=0..9999	Attach the policy to the PPP interface.
6	**enable sqos**	Enable software QoS.

Table 39-18: Procedure for configuring software QoS on frame relay interfaces (cont.)

Step	Commands	Action
7	**set interface mtu** set interface=*interface* mtu=256	If the interface carries traffic that is sensitive to latency, such as voice traffic, force large packets to be fragmented.

The Switch Instance

You can configure software QoS on the *switch instance*, swi0, which is the internal interface to the switch ports. This is helpful if:

- You use both Ethernet ports on an AR450S router for high-speed WAN connections, because the switch instance may form a bottleneck (Figure 39-15).
- You need to control the quality of service given to traffic destined for particular VLANs or ports. In this case, you can classify packets into different traffic classes according to VLAN or port, specify the appropriate QoS controls in each traffic class, add the traffic classes to a policy, and apply the policy to swi0.
- You need to control the quality of service for a PPPoVLAN interface. You can classify packets according to their PPP index, which lets you control different PPP interfaces over the same VLAN.

Figure 39-15: Example of when you may require software QoS on the switch instance

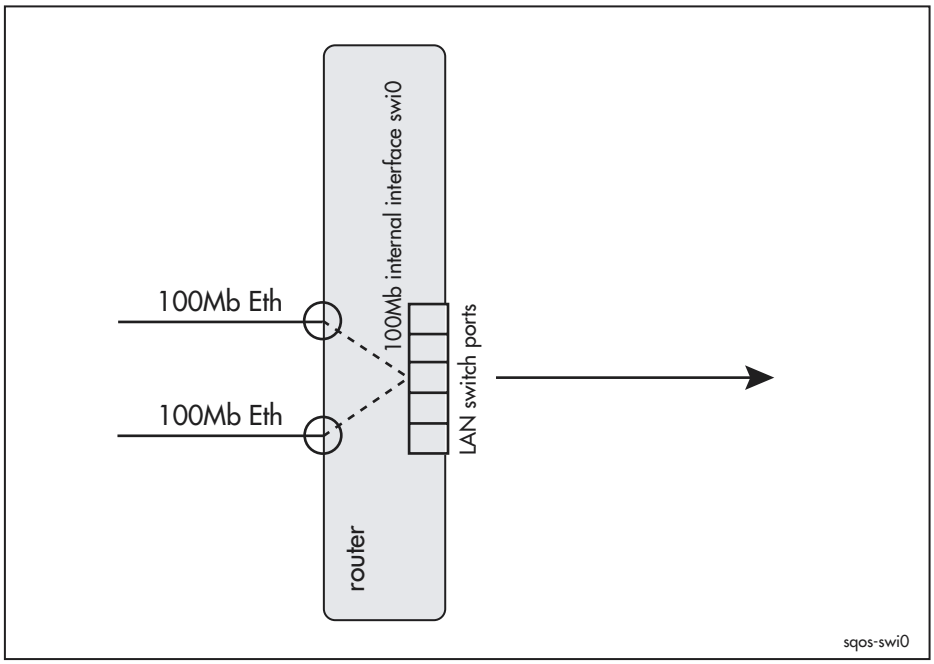


Table 39-19: Procedure for configuring software QoS on the switch instance

Step	Commands	Action
1	create classifier=1..9999 [iinterface=*interface*] [eport=*port*] [iport=*port*] [svlan={*vlan-name*|1..4094|any}] [dvlan={*vlan-name*|1..4094|any}] [vlanpriority={*priority-list*|any}] [ethformat={802.2|ethii|netwareraw|snap|any}] [macdaddr={*macadd*|any}] [macsaddr={*macadd*|any}] [mactype={l2ucast|l2bmcast|any}] [protocol={*protocol-type*|arp|ip|ipv6|ipx|any}] [pppindex=0..1023] [ipdaddr={*ipadd*[/0..32]|*ipv6add*[/0..128]|any}] [ipsaddr={*ipadd*[/0..32]|*ipv6add*[/0..128]|any}] [ipdscp={*dscp-list*|any}] [iptos={0..7|any}] [ipfrag={yes|no|any}] [ipoptions={yes|no|any}] [ipflowlabel={0..1048575|any}] [ipprotocol={tcp|udp|icmp|igmp|ospf|nontcpudp|any|*ip-protocol*}] [icmptype={any|echorply|unreachable|quench|redirect|echo|advertisement|solicitation|timeexceed|parameter|tstamp|tstamprply|inforeq|inforep|addrreq|addrrep|namereq|namerply|*icmp-type*}] [icmpcode={any|filter|fragment|fragreassm|hostcomm|hostisolated|hostprec|hostredirect|hostrtos|hosttos|hostunknown|hostunreach|netcomm|netredirect|netrtos|nettos|netunknown|netunreach|noptr|portunreach|precedent|protunreach|ptrproblem|sourceroute|ttl|*icmp-code*}] [tcpflags={{urg|ack|rst|syn|fin}[,...]|any}] [tcpdport={*port-range*|any}] [tcpsport={*port-range*|any}] [udpdport={*port-range*|any}] [udpsport={*port-range*|any}]	Create classifiers. Valid parameters include: ● interface, port and VLAN. This lets you separate traffic destined for different VLANs or ports. ● Ethernet ● PPP index for PPPoVLAN traffic. This lets you separate different PPP interfaces on the same VLAN. ● Layer 3 ● Layer 4
2		Create the QoS policy and its underlying hierarchy. See Table 39-4 on page 39-33 for details.
3	**set sqos interface**=swi0 outpolicy=0..9999	Attach the policy to swi0.
4	**enable sqos**	Enable software QoS.
5	**set interface mtu** set interface=*interface* mtu=256	If the interface carries traffic that is sensitive to latency, such as voice traffic, force large packets to be fragmented.

Configuring Software QoS on Tunnels

QoS is performed on packets before they are encapsulated and enter the tunnel.

VPN

VPN tunnels provide a secure connection across a WAN link.

Table 39-20: Procedure for configuring software QoS on VPN tunnels

Step	Commands	Action
1		Create the VPN tunnel. See Chapter 48, IP Security (IPsec) for commands and examples.
2	create classifier=1..9999 [iinterface=*interface*] [iport=*port*] [ipdaddr={*ipadd*[/0..32]|*ipv6add*[/0..128]|any}] [ipsaddr={*ipadd*[/0..32]|*ipv6add*[/0..128]|any}] [ipdscp={*dscp-list*|any}] [iptos={0..7|any}] [ipfrag={yes|no|any}] [ipoptions={yes|no|any}] [ipflowlabel={0..1048575|any}] [ipprotocol={tcp|udp|icmp|igmp|ospf| nontcpudp|any|*ip-protocol*} [icmptype={any|echorply|unreachable|quench| redirect|echo|advertisement|solicitation| timeexceed|parameter|tstamp|tstamprply| inforeq|inforep|addrreq|addrrep|namereq| namerply|*icmp-type*}] [icmpcode={any|filter|fragment|fragreassm| hostcomm|hostisolated|hostprec|hostredirect| hostrtos|hosttos|hostunknown|hostunreach| netcomm|netredirect|netrtos|nettos| netunknown|netunreach|noptr|portunreach| precedent|protunreach|ptrproblem|sourceroute| ttl|*icmp-code*}] [tcpflags={{urg|ack|rst|syn|fin}[,...]|any}] [tcpdport={*port-range*|any}] [tcpsport={*port-range*|any}] [udpdport={*port-range*|any}] [udpsport={*port-range*|any}]	Create classifiers. Valid parameters include: ● ingress interface and port ● IP and IPv6 Layer 3 ● Layer 4
3		Create the QoS policy and its underlying hierarchy. See Table 39-4 on page 39-33 for details.
4	**set sqos interface**=ipsec-*policyname* tunnel|policy=0..9999 where *policyname* is the name of the IPsec policy	Attach the policy to the VPN tunnel.
5	**enable sqos**	Enable software QoS.
6	**set interface mtu** set interface=*interface* mtu=256	If the interface carries traffic that is sensitive to latency, such as voice traffic, force large packets to be fragmented.

6 to 4

IPv6 to IPv4 tunnels take IPv6 traffic from your LAN and send it out over an IPv4 WAN link.

Table 39-21: Procedure for configuring software QoS on 6-to-4 tunnels

Step	Commands	Action
1		Create the VPN tunnel. See Chapter 31, Internet Protocol version 6 (IPv6) for commands and an example.
2	create classifier=1..9999 [iinterface=*interface*] [iport=*port*] [ipdaddr={*ipv6add*[/0..128]|any}] [ipsaddr={*ipv6add*[/0..128]|any}] [ipdscp={*dscp-list*|any}] [ipflowlabel={0..1048575|any}] [ipprotocol={tcp|udp|icmp|any|*ip-protocol*} [icmptype={any|echorply|unreachable|quench| redirect|echo|advertisement|solicitation| timeexceed|parameter|tstamp|tstamprply| inforeq|inforep|addrreq|addrrep|namereq| namerply|*icmp-type*}] [icmpcode={any|filter|fragment|fragreassm| hostcomm|hostisolated|hostprec|hostredirect| hostrtos|hosttos|hostunknown|hostunreach| netcomm|netredirect|netrtos|nettos| netunknown|netunreach|noptr|portunreach| precedent|protunreach|ptrproblem|sourceroute| ttl|*icmp-code*}] [tcpflags={{urg|ack|rst|syn|fin}[,...]|any}] [tcpdport={*port-range*|any}] [tcpsport={*port-range*|any}] [udpdport={*port-range*|any}] [udpsport={*port-range*|any}]	Create classifiers. Valid parameters include: ● ingress interface and port ● IPv6 Layer 3 ● IPv6 Layer 4
3		Create the QoS policy and its underlying hierarchy. See Table 39-4 on page 39-33 for details.
4	**set sqos interface**=*virtn* tunnel|policy=0..9999 where *virtx* is the name of the tunnel (e.g. virt0)	Attach the policy to the 6-to-4 tunnel.
5	**enable sqos**	Enable software QoS.
6	**set interface mtu** set interface=*interface* mtu=256	If the interface carries traffic that is sensitive to latency, such as voice traffic, force large packets to be fragmented.

Generic Router Encapsulation (GRE)

Table 39-22: Procedure for configuring software QoS on GRE tunnels

Step	Commands	Action
1	GRE commands	Create the GRE tunnel. See Chapter 30, Generic Routing Encapsulation (GRE) for commands and examples.
2	create classifier=1..9999 [iinterface=*interface*] [iport=*port*] [ipdaddr={*ipadd*[/0..32]|any}] [ipsaddr={*ipadd*[/0..32]|any}] [ipdscp={*dscp-list*|any}] [iptos={0..7|any}] [ipfrag={yes|no|any}] [ipoptions={yes|no|any}] [ipprotocol={tcp|udp|icmp|igmp|ospf|nontcpudp|any|*ip-protocol*} [icmptype={any|echorply|unreachable|quench|redirect|echo|advertisement|solicitation|timeexceed|parameter|tstamp|tstamprply|inforeq|inforep|addrreq|addrrep|namereq|namerply|*icmp-type*}] [icmpcode={any|filter|fragment|fragreassm|hostcomm|hostisolated|hostprec|hostredirect|hostrtos|hosttos|hostunknown|hostunreach|netcomm|netredirect|netrtos|nettos|netunknown|netunreach|noptr|portunreach|precedent|protunreach|ptrproblem|sourceroute|ttl|*icmp-code*}] [tcpflags={{urg|ack|rst|syn|fin}[,...]|any}] [tcpdport={*port-range*|any}] [tcpsport={*port-range*|any}] [udpdport={*port-range*|any}] [udpsport={*port-range*|any}]	Create classifiers. Valid parameters include: ● ingress interface and port ● IP Layer 3 ● Layer 4
3		Create the QoS policy and its underlying hierarchy. See Table 39-4 on page 39-33 for details.
4	**set sqos interface**=*gren* tunnel|policy=0..9999 where *gren* is the name of the tunnel (e.g. gre1)	Attach the policy to the GRE tunnel.
5	**enable sqos**	Enable software QoS.
6	**set interface mtu** set interface=*interface* mtu=256	If the interface carries traffic that is sensitive to latency, such as voice traffic, force large packets to be fragmented.

Interaction with Other Modules

This section describes the effect of software QoS on some other software features. Some of these are alternatives to software QoS, and some interact with it.

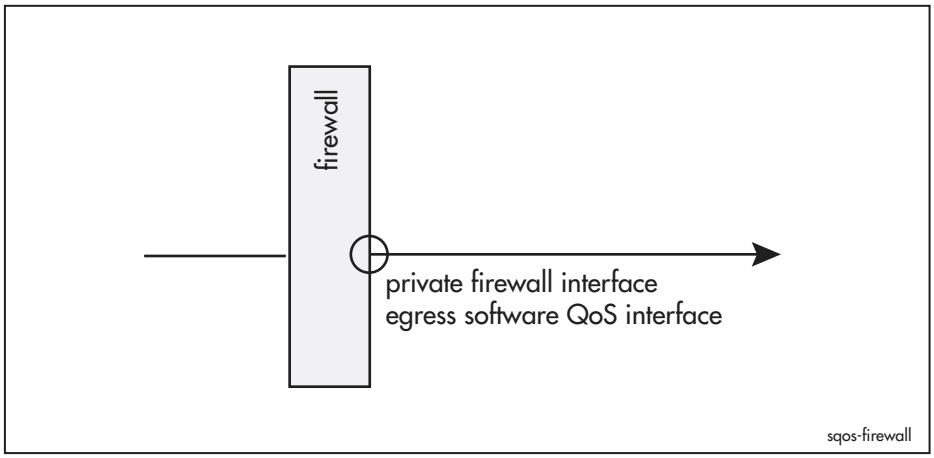
Network Address Translation (NAT)

Network Address Translation (NAT) is available through the firewall or the IP NAT feature. It translates between private IP settings on the LAN and public IP settings on the WAN. Therefore it can change a packet's source or destination IP address, TCP port or UDP port, depending on the NAT settings and direction of traffic flow. In general, NAT gives the same IP settings to all translated packets that leave your LAN through a given interface.

Software QoS may use these IP settings to classify packets. By default, it uses the pre-NAT settings for classification because these contain the distinguishing information. You need to use the post-NAT settings instead if **all** the following points apply (Figure 39-16):

- you are configuring software QoS on traffic that comes from a WAN link to your LAN
- you are applying QoS at the egress (LAN) interface
- you are classifying on destination IP address, destination TCP port or destination UDP port
- the interface is a bottleneck

Figure 39-16: Firewall interface that requires post-NAT IP settings



To use post-NAT IP settings, use one of the commands:

```
create sqos policy=id-list ignoreprenatinfo=yes
   [other-options]

set sqos policy=id-list ignoreprenatinfo=yes [other-options]
```

Resource Reservation Protocol (RSVP)

RSVP and software QoS cannot effectively be used together. RSVP assigns packets to a reserved queue, which software QoS treats as system traffic. Software QoS policies use WRR to schedule system traffic, and limit it to 20% of the available bandwidth by default, and a maximum of 50%.

Priority Filters

Priority filters and software QoS cannot be used together. Priority filters provide a limited subset of QoS functionality. Software QoS offers much greater control over service quality.

Policy Filters

Policy filters cannot be used in a DiffServ domain. Policy filters use the TOS precedence field of the IP packet to determine routing. DiffServ also uses this field.

Bandwidth Limiting on Ethernet Interfaces

When software QoS is enabled on an Ethernet interface such as eth0, the command **set eth maxbandwidth** has no effect. To use software QoS to limit the bandwidth, configure a virtual bandwidth limit on the appropriate policy or traffic class.

Counters

You can view the following counters for software QoS:

- for **classifier** counters, use the command:

```
show sqos counters classifier[=id-list]
   [direction={in|out|tunnel|all}] [interface=interface]
```

- for **DAR** counters, use the command:

```
show sqos counters dar[=id-list]
   [direction={in|out|tunnel}] [interface=interface]
```

- for **traffic class** counters, use the command:

```
show sqos counters trafficclass[={id-list|default|system}]
   [direction={in|out|tunnel|all}] [interface=interface]
```

- for **policy** counters, use the command:

```
show sqos counters policy[=id-list]
   [direction={in|out|tunnel}] [interface=interface]
```

For each category of counter, you can view counters relating to:

- a particular entity, for example a particular classifier or a particular traffic class, by specifying its ID
- actions in ingress QoS, egress QoS, or QoS on tunnel interfaces, by specifying the **direction** parameter
- QoS on a particular interface, by specifying the **interface** parameter

Information about counters help you determine if your QoS configuration is working as intended. For example, you can work out which traffic classes are the busiest and where most of the packets are dropped. If the traffic distribution across classes is not what you want, you can rearrange the properties of the classes and monitor the configuration until it operates as desired.

If a Software QoS policy does not appear to be processing packets in the way you expect, the counters can help you track down what is wrong. For example, if a classifier has been misconfigured so it is actually getting no hits, the classifier counters show this inactivity. Some statistics that you can collect from these counters are:

- how many hits there have been on a given classifier
- how many dynamic classifiers have been created by a DAR
- a list of the currently active dynamic classifiers for a given DAR
- total packets processed by a traffic class or policy
- current and average queue lengths for a traffic class
- average latency for packets passing through a given traffic class
- number of packets classified green, yellow, and red by a given traffic class
- number of packets dropped by the RED curves on a given traffic class

Counters also provide clear evidence of whether Service Level Agreements are adhered to, by showing whether the traffic levels for any part of a contracted service are meeting the contracted requirements.

Debugging

The following table describes software QoS debug options:

Option	Meaning
all	All debugging modes.
dar	Notifications when DAR objects and instances are created or destroyed.
dardata	More detailed information about SIP and RTSP data.
engine	Debugging information related to the packet conditioning engine.
error	Critical error debugging information, including a stack trace.
info	General command debugging information.
mark	Packet marking debugging information.
pkt	Packet debugging.

Some of these debug modes may help you check that software QoS is functioning as expected. For example, the **info** option gives additional information when the router carries out commands. The **mark** option shows that packets are being marked with different priority values. The **dar** option shows that dynamic classifiers have been created for appropriate voice traffic.

To enable debugging, use the command:

```
enable sqos
   debug={all|dar|dardata|engine|error|info|mark|pkt}
```

The following figures show examples of the output from some of these debugging modes.

Figure 39-17: Example output from the command **enable sqos debug=dar**

```
Manager >
SQOS DAR: Classifer=10000 tc=1 ip=192.168.2.1/32 port=45678-45679 created
Manager >
SQOS DAR: Classifer=10001 tc=1 ip=192.168.1.1/32 port=38168-38169 created
Manager >
SQOS DAR: Classifer=10001 tc=1 ip=192.168.1.1/32 port=38168-38169 destroyed
SQOS DAR: Classifer=10000 tc=1 ip=192.168.2.1/32 port=45678-45679 destroyed
```

Figure 39-18: Example output from the command **enable sqos debug=dardata**

```
Manager >
SQOS DARDATA: INVITE sip:192.168.1.1:5060 SIP/2.0
SQOS DARDATA: Via: SIP/2.0/UDP 192.168.2.3:5060
SQOS DARDATA: Max-Forwards: 70
SQOS DARDATA: From: "user1"
<sip:user123456@mycompa.com>;tag=3b2ec08b66ee47b496f3a49a58aa7fa2
SQOS DARDATA: To: <sip:192.168.1.1>
SQOS DARDATA: Call-ID: 470f605893de4c4e9bc9e95adedee8b1@192.168.0.1=CALL-ID
SQOS DARDATA: CSeq: 1 INVITE
SQOS DARDATA: Contact: <sip:192.168.2.3:5060>
SQOS DARDATA: User-Agent: RTC/1.2
SQOS DARDATA: Content-Type: application/sdp
SQOS DARDATA: CONTENT-LENGTH: 284=284
SQOS DARDATA: <END>
SQOS DARDATA: v=0
SQOS DARDATA: o=- 0 0 IN IP4 192.168.2.1
SQOS DARDATA: s=session
SQOS DARDATA: c=IN IP4 192.168.2.1
SQOS DARDATA: b=CT:110
SQOS DARDATA: t=0 0
SQOS DARDATA: m=audio 45678 RTP/AVP 97 0 8 4 101
SQOS DARDATA: a=rtpmap:97 red/8000
SQOS DARDATA: a=rtpmap:0 PCMU/8000
SQOS DARDATA: a=rtpmap:8 PCMA/8000
SQOS DARDATA: a=rtpmap:4 G723/8000
SQOS DARDATA: a=rtpmap:101 telephone-event/8000
SQOS DARDATA: a=fmtp:101 0-16
SQOS DARDATA: a=encryption:rejected
```

Figure 39-19: Example output from the command **enable sqos debug=pkt**

```
Manager >
SQOS PKT: POLI=1 (OUT) TC=sys qPkts=1 qBytes=42 Enqueued
SQOS PKT: POLI=1 (OUT) TC=sys qPkts=0 qBytes=0 Dequeued
SQOS PKT: POLI=1 (OUT) TC=1 qPkts=1 qBytes=1512 Enqueued
SQOS PKT: POLI=1 (OUT) TC=1 qPkts=0 qBytes=0 Dequeued
SQOS PKT: POLI=1 (OUT) TC=1 qPkts=1 qBytes=1512 Enqueued
SQOS PKT: POLI=1 (OUT) TC=1 qPkts=0 qBytes=0 Dequeued
Manager >
SQOS PKT: POLI=1 (OUT) TC=1 qPkts=1 qBytes=1512 Enqueued
SQOS PKT: POLI=1 (OUT) TC=1 qPkts=0 qBytes=0 Dequeued
```

Figure 39-20: Example output from the command **enable sqos debug=mark**

```
Manager >
SQOS MARK: Pkt 04b7880c, Old DSCP 0
SQOS MARK: IPv4 Pkt 04b7880c, New DSCP 10
SQOS MARK: Packet dump after marking
    0050fc31 d7ad0000 cd08106f 08004528 05dab22d 40003f11 27d9ac70 0101ac72
    0201041b 138905c6 07e80000 000041a6 57740009 939c0000 00000000 00010000
    13890000 00000000 fa00ffff ff9c3637 38393031 32333435 36373839 30313233
    34353637
```

Figure 39-21: Example output from the command **enable sqos debug=engine**

```
SQOS ENGINE: callback eth0 2
SQOS ENGINE: callback eth0 2
SQOS ENGINE: callback eth0 2
SQOS ENGINE: callback eth0 2
SQOS ENGINE: callback eth0 2
SQOS ENGINE: callback eth0 2
SQOS ENGINE: callback eth0 2
```

Figure 39-22: Example output from the command **enable sqos debug=info**

```
Manager > set sqos int=eth0 outpolicy=1

SQOS INFO: SQOS Active on eth0 (Egress)
Info (1123003): Operation successful.
```

Configuration Examples

Examples in this section include common and complex network situations:

- **1: Guaranteeing VoIP Traffic**
- **2: Guaranteeing VoIP Traffic using DAR**
- **3: Guaranteeing VoIP Traffic While Maintaining File Server Traffic**
- **4: Guaranteeing VoIP Traffic over a VPN Tunnel**
- **5: VoIP, Critical Database, and File Server Traffic**
- **6: Multiple Applications over Frame Relay**

The first example begins with the simple goal of making VoIP calls at the same time as non-critical file server downloads over a 128 kbps PPP link; the next four examples build on this one. The last example shows multiple applications running over a frame relay link, including voice, video conferencing, network monitoring, and server traffic.

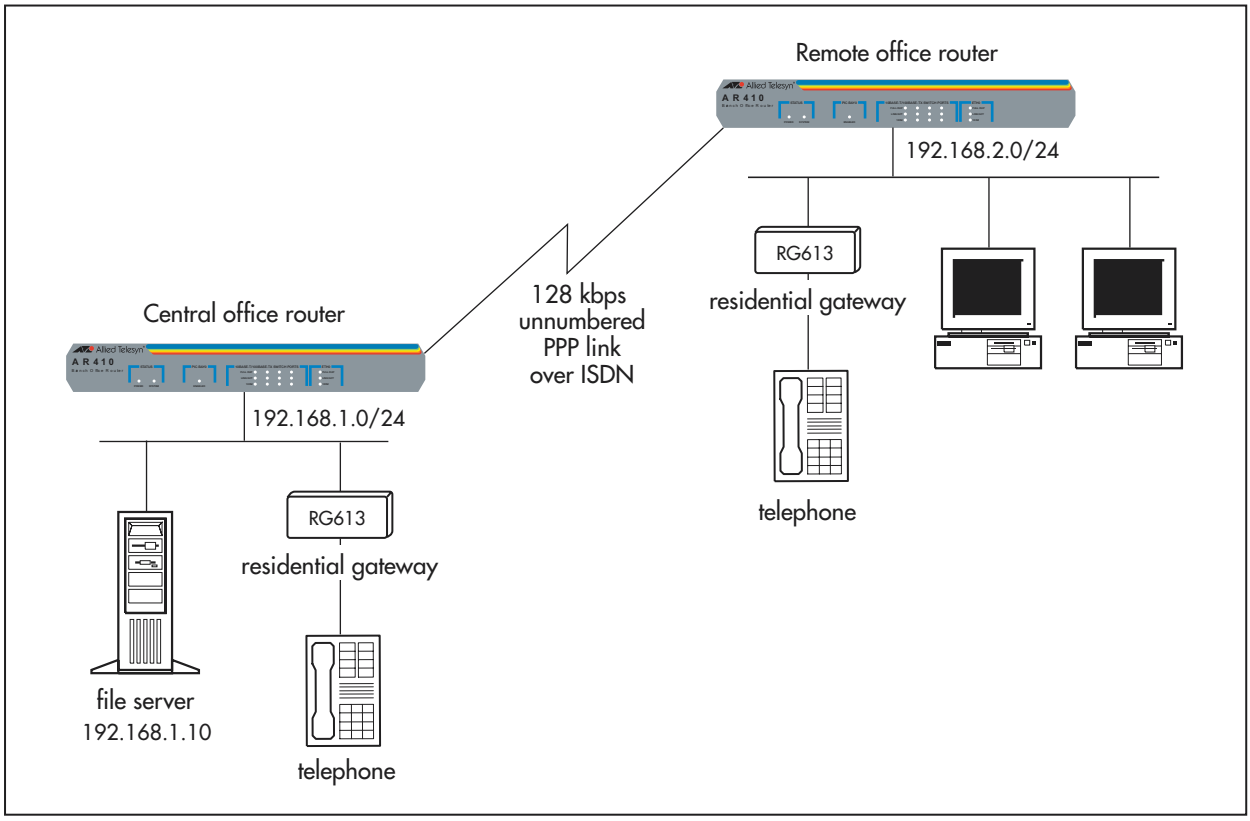
Some interface types, port types, or command options in these examples may not be supported on your router. Interfaces and port types vary depending on the router model, and whether an expansion unit (PIC, NSM) is installed. For more information, see the Hardware Reference.

1: Guaranteeing VoIP Traffic

In this scenario (Figure 39-23):

- VoIP traffic has the highest priority possible and a short queue, so the routers always transmit VoIP traffic with low drop rate, delay, and jitter.
- the routers identify VoIP traffic by the UDP ports it uses.
- the routers drop other traffic as necessary so they can send the VoIP traffic at high quality.

Figure 39-23: Configuration that guarantees VoIP traffic



```
# Guaranteeing VoIP traffic on 128kbps PPP link over ISDN
# Central office configuration

set system name=central
set system territory=europe

# Configure ISDN and PPP
add user=remote password=rempass login=no telnet=no
add isdn call=office number=0 prec=in dir=in searchsub=local
create ppp=0 over=isdn-office authentication=chap echo=30 lqr=off bap=off

# Configure IP
enable ip
add ip interface=vlan1 ip=192.168.1.254
add ip interface=ppp0 ip=0.0.0.0 mask=0.0.0.0
add ip route=192.168.2.0 int=ppp0 next=0.0.0.0

# Create a classifier for VoIP (This example assumes voice traffic
# uses UDP ports between 16300 and 16320)
create class=1 udpdport=16300-16320

# Create a classifier to match on the SIP signaling traffic
create class=2 udpdport=5060

# Create a QoS policy.
create sqos policy=1

# Create a traffic class for VoIP traffic and give it the highest
# priority and a short queue
create sqos trafficclass=1 priority=15 maxqlen=10

# Create a traffic class for the SIP signalling traffic and give
# it the second highest priority
create sqos trafficclass=2 priority=14

# Add the traffic classes to the policy.
add sqos policy=1 trafficclass=1,2

# Add the classifiers to the traffic classes
add sqos trafficclass=1 class=1
add sqos trafficclass=2 class=2

# Use the policy on ppp0
set sqos interface=ppp0 outpolicy=1

# Enable software QoS
enable sqos
```

```
# Guaranteeing VoIP traffic on 128kbps PPP link over ISDN
# Remote office configuration

set system name=remote
set system territory=europe

# Configure ISDN and PPP
add isdn call=office number=your-central-office-number prec=out outsub=local
create ppp=0 over=isdn-office username=remote passw=rempass echo=30 lqr=off bap=off

# Configure IP
enable ip
add ip interface=vlan1 ip=192.168.2.254
add ip interface=ppp0 ip=0.0.0.0 mask=0.0.0.0
add ip route=192.168.1.0 int=ppp0 next=0.0.0.0

# Create a classifier for VoIP (This example assumes Voice traffic
# is using UDP ports between 16300 and 16320)
create class=1 udpdport=16300-16320

# Create a classifier to match on the SIP signaling traffic
create  class=2 udpdport=5060

# Create a QoS policy.
create sqos policy=1

# Create a traffic class for VoIP traffic and give it the highest
# priority and a short queue
create sqos trafficclass=1 priority=15 maxqlen=10

# Create a traffic class for the SIP signalling traffic and give
# it the second highest priority
create sqos trafficclass=2 priority=14

# Add the traffic classes to the policy.
add sqos policy=1 trafficclass=1,2

# Add the classifiers to the traffic classes
add sqos trafficclass=1 class=1
add sqos trafficclass=2 class=2

# Use the policy on ppp0
set sqos interface=ppp0 outpolicy=1

# Enable software QoS
enable sqos
```

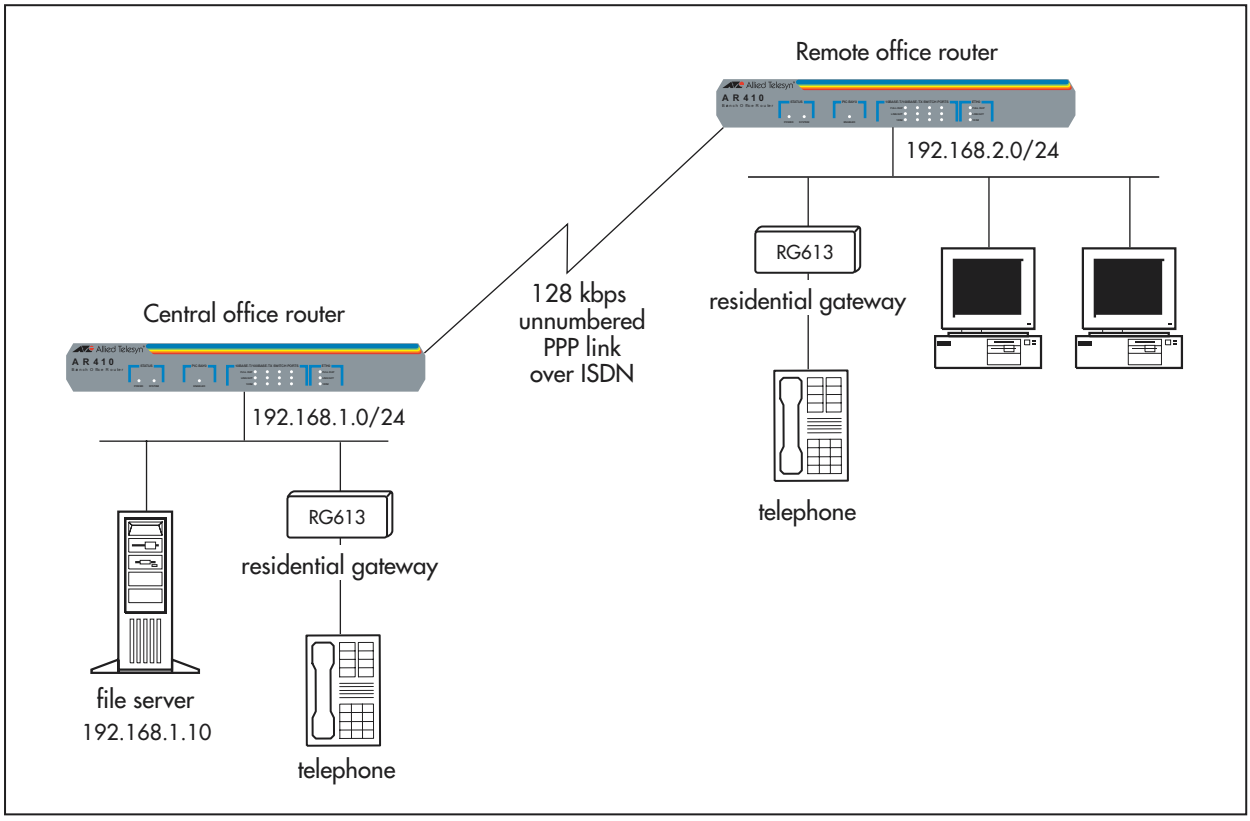
2: Guaranteeing VoIP Traffic using DAR

This scenario is an alternative to Scenario 1, and uses DAR to identify the voice traffic. The network set-up is identical to Scenario 1.

In this scenario (Figure 39-24):

- VoIP traffic has the highest priority possible and a short queue, so the routers always transmit VoIP traffic with low drop rate, delay, and jitter.
- the routers drop traffic as necessary so that they can send VoIP traffic at high quality.

Figure 39-24: Configuration that guarantees VoIP traffic by using DAR



```
# Guaranteeing VoIP traffic using DAR on 128kbps PPP link over ISDN
# Central office configuration

set system name=central
set system territory=europe

# Configure ISDN and PPP
add user=remote password=rempass login=no telnet=no
add isdn call=office number=0 prec=in dir=in searchsub=local
create ppp=0 over=isdn-office authentication=chap echo=30 lqr=off bap=off

# Configure IP
enable ip
add ip interface=vlan1 ip=192.168.1.254
add ip interface=ppp0 ip=0.0.0.0 mask=0.0.0.0
add ip route=192.168.2.0 int=ppp0 next=0.0.0.0

# Create a DAR object for VoIP, to match sessions initiated by SIP signalling
create sqos dar=1 prot=sip

# The DAR does not match the signaling traffic itself, so create a separate
# classifier to match on the SIP signaling traffic
create  class=1 udpdport=5060

# Put the DAR onto the PPP interface so it will recognise when a phone call is set
# up over the PPP interface
add sqos interface=ppp0 dar=1

# Create a QoS policy
create sqos policy=1

# Create a traffic class for VoIP traffic and give it the highest priority and a
# short queue
create sqos trafficclass=1 priority=15 maxqlen=10

# Create a traffic class for the SIP signalling traffic and give it the second highest
# priority
create sqos trafficclass=2 priority=14

# Add the traffic classes to the policy
add sqos policy=1 trafficclass=1,2

# Add the DAR object and SIP classifier to the traffic classes
add sqos trafficclass=1 dar=1
add sqos trafficclass=2 class=1

# Use the policy on ppp0
set sqos interface=ppp0 outpolicy=1

# Enable software QoS
enable sqos
```

```
# Guaranteeing VoIP traffic using DAR on 128kbps PPP link over ISDN
# Remote office configuration

set system name=remote
set system territory=europe

# Configure ISDN and PPP
add isdn call=office number=your-central-office-number prec=out outsub=local
create ppp=0 over=isdn-office username=remote passw=rempass echo=30 lqr=off bap=off

# Configure IP
enable ip
add ip interface=vlan1 ip=192.168.2.254
add ip interface=ppp0 ip=0.0.0.0 mask=0.0.0.0
add ip route=192.168.1.0 int=ppp0 next=0.0.0.0

# Create a DAR object for VoIP, to match sessions initiated by SIP signalling
create sqos dar=1 prot=sip

# The DAR does not match the signaling traffic itself, so create a separate
# classifier to match on the SIP signaling traffic
create class=1 udpdport=5060

# Put the DAR onto the PPP interface so it will recognise when a phone call is set
# up over the PPP interface
add sqos interface=ppp0 dar=1

# Create a QoS policy
create sqos policy=1

# Create a traffic class for VoIP traffic and give it the highest priority and a
# short queue
create sqos trafficclass=1 priority=15 maxqlen=10

# Create a traffic class for the SIP signalling traffic and give it the second highest
# priority
create sqos trafficclass=2 priority=14

# Add the traffic classes to the policy
add sqos policy=1 trafficclass=1,2

# Add the DAR object and SIP classifier to the traffic classes
add sqos trafficclass=1 dar=1
add sqos trafficclass=2 class=1

# Use the policy on ppp0
set sqos interface=ppp0 outpolicy=1

# Enable software QoS
enable sqos
```

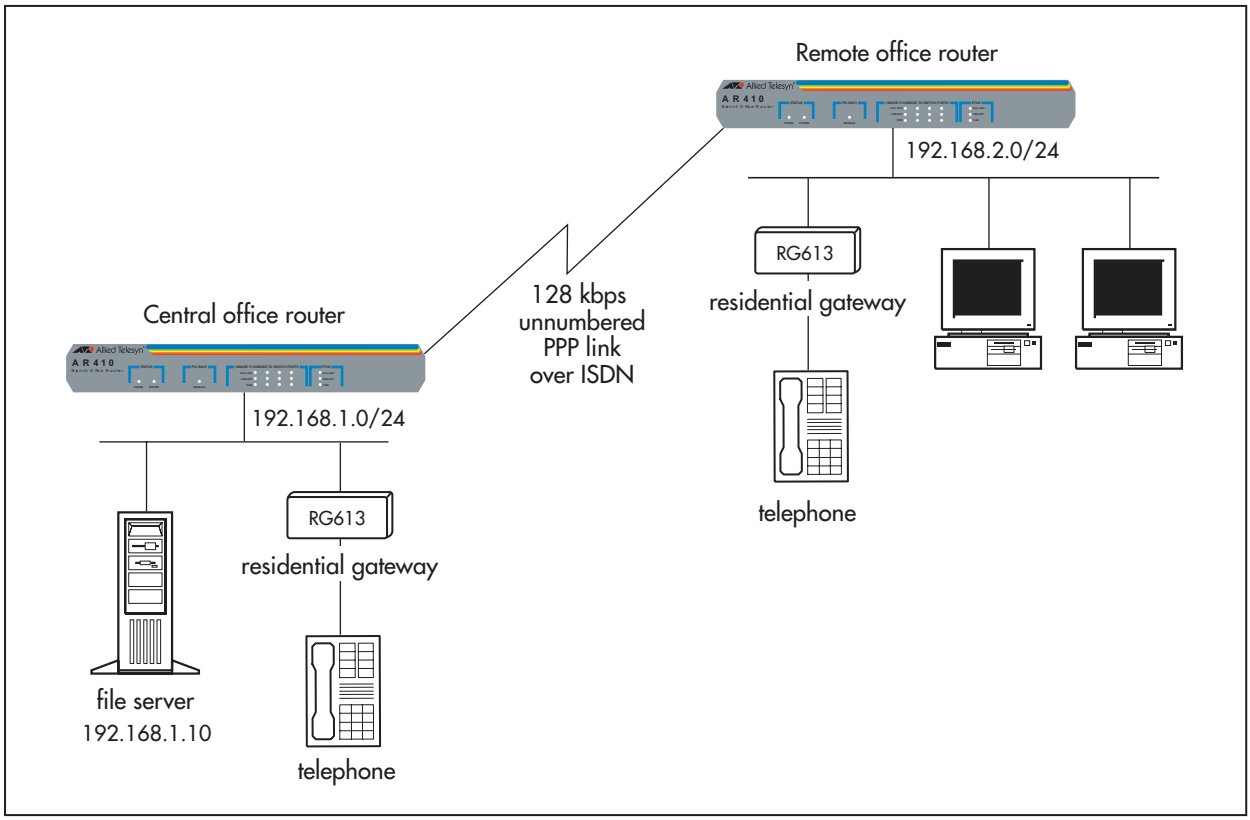
3: Guaranteeing VoIP Traffic While Maintaining File Server Traffic

This scenario expands on Scenario 1 by improving the quality of service for file server traffic. The network set-up is identical to Scenario 1.

In this scenario (Figure 39-25):

- VoIP traffic has the highest priority possible and a short queue, so the routers always transmit VoIP traffic with low drop rate, delay, and jitter.
- the routers identify VoIP traffic by the UDP ports it uses.
- file server traffic has the next highest priority. The routers use a medium RED curve set to drop file server traffic as necessary and control the TCP flows. We recommend RED because the traffic class may include multiple simultaneous flows to and from the file server.

Figure 39-25: Configuration that guarantees VoIP traffic while maintaining file server traffic



```
# Guaranteeing VoIP traffic and maintaining file server downloads
# Central office configuration

set system name=central
set system territory=europe

# Configure ISDN and PPP
add user=remote password=rempass login=no telnet=no
add isdn call=office number=0 prec=in dir=in searchsub=local
create ppp=0 over=isdn-office authentication=chap echo=30 lqr=off bap=off

# Configure IP
enable ip
add ip interface=vlan1 ip=192.168.1.254
add ip interface=ppp0 ip=0.0.0.0 mask=0.0.0.0
add ip route=192.168.2.0 int=ppp0 next=0.0.0.0

# Create a classifier for VoIP (This example assumes voice traffic uses UDP ports
# between 16300 and 16320)
create class=1 udpdport=16300-16320

# Create a classifier to match on the SIP signaling traffic
create class=2 udpdport=5060

# Create a classifier for traffic from the file server
create class=3 ipsadd=192.168.1.10

# Create a QoS policy
create sqos poli=1

# Create a traffic class for VoIP traffic and give it the highest priority and a
# short queue
create sqos trafficclass=1 priority=15 maxqlen=10

# Create a traffic class for the SIP signalling traffic and give it the second highest
# priority
create sqos trafficclass=2 priority=14

# Create a traffic class for file server traffic, give it the next highest priority
# and use a medium RED curve set
create sqos trafficclass=3 priority=13 red=1

# Add the traffic classes to the policy
add sqos policy=1 trafficclass=1-3

# Add the classifiers to the traffic classes
add sqos trafficclass=1 classifier=1
add sqos trafficclass=2 classifier=2
add sqos trafficclass=3 classifier=3

# Use the policy on ppp0
set sqos interface=ppp0 outpolicy=1

# Enable software QoS
enable sqos
```

```
# Guaranteeing VoIP traffic and maintaining file server downloads
# Remote office configuration

set system name=remote
set system territory=europe

# Configure ISDN and PPP
add isdn call=office number=your-central-office-number prec=out outsub=local
create ppp=0 over=isdn-office username=remote passw=rempass echo=30 lqr=off bap=off

# Configure IP
enable ip
add ip interface=vlan1 ip=192.168.2.254
add ip interface=ppp0 ip=0.0.0.0 mask=0.0.0.0
add ip route=192.168.1.0 int=ppp0 next=0.0.0.0

# Create a classifier for VoIP (This example assumes voice traffic uses UDP ports
# between 16300 and 16320)
create class=1 udpdport=16300-16320

# Create a classifier to match on the SIP signaling traffic
create class=2 udpdport=5060

# Create a classifier for traffic to the file server
create class=3 ipdadd=192.168.1.10

# Create a QoS policy
create sqos poli=1

# Create a traffic class for VoIP traffic and give it the highest priority and a
# short queue
create sqos trafficclass=1 priority=15 maxqlen=10

# Create a traffic class for the SIP signalling traffic and give it the second highest
# priority
create sqos trafficclass=2 priority=14

# Create a traffic class for file server traffic, give it the next highest priority
# and use a medium RED curve set
create sqos trafficclass=3 priority=13 red=1

# Add the traffic classes to the policy
add sqos policy=1 trafficclass=1-3

# Add the classifiers to the traffic classes
add sqos trafficclass=1 classifier=1
add sqos trafficclass=2 classifier=2
add sqos trafficclass=3 classifier=3

# Use the policy on ppp0
set sqos interface=ppp0 outpolicy=1

# Enable software QoS
enable sqos
```

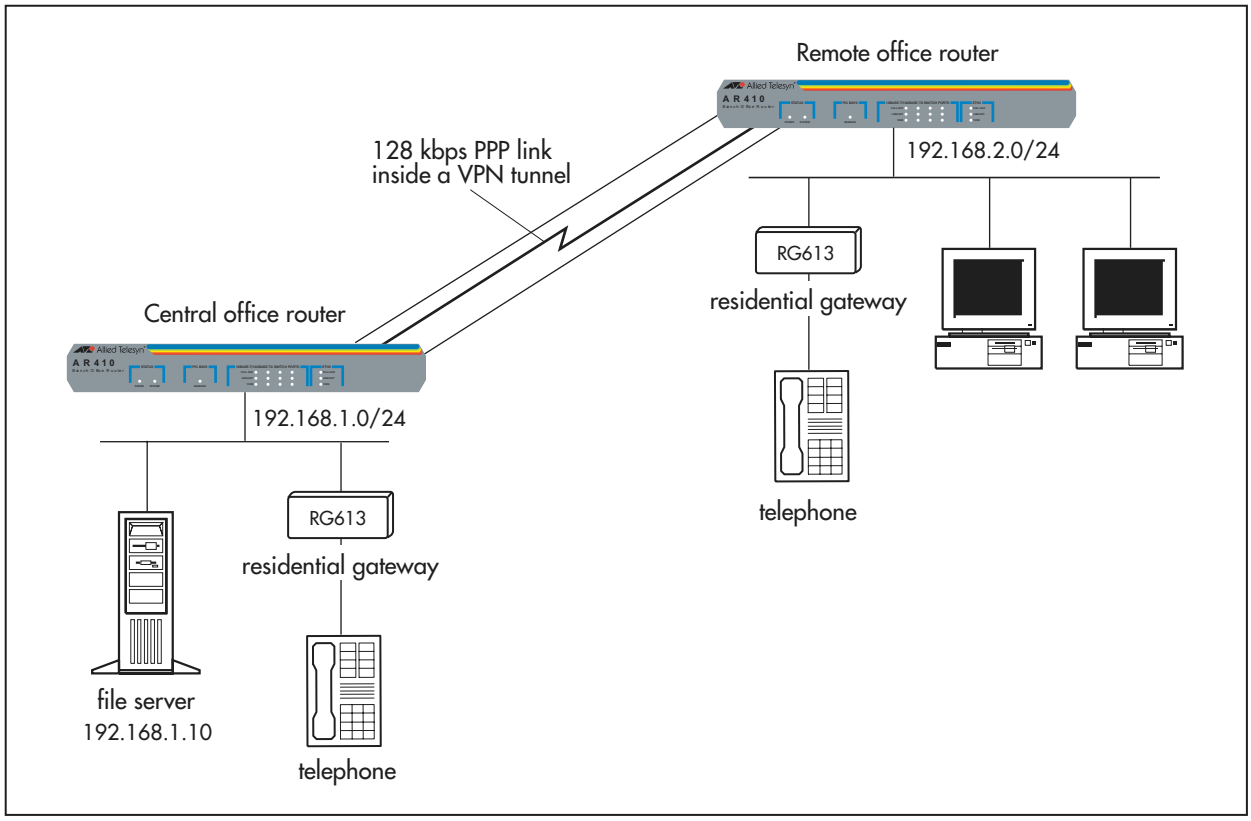
4: Guaranteeing VoIP Traffic over a VPN Tunnel

This scenario expands on Scenario 3 by sending all traffic between the central office and the remote office securely over a VPN tunnel.

In this scenario (Figure 39-26):

- VoIP traffic has the highest priority possible and a short queue, so the routers always transmit VoIP traffic with low drop rate, delay, and jitter.
- the routers identify VoIP traffic by the UDP ports it uses.
- file server traffic has the next highest priority. The routers use a medium RED curve set to drop file server traffic as necessary and control the TCP flows. We recommend RED because the traffic class may include multiple simultaneous flows to and from the file server.

Figure 39-26: Configuration for software QoS for traffic over a VPN tunnel



```
# Guaranteeing VoIP traffic and maintaining file server downloads over a VPN tunnel
# Central office configuration

set system name=central
set system territory=europe

# Create a user with Security Officer privilege and enable secure mode
add user=secoff pass=verysecret priv=securityOfficer lo=yes
set user=secoff telnet=no netmask=255.255.255.255
enable system security_mode
set user securedelay=600

# create an encryption key for ISAKMP to protect and authenticate its messages
create enco key=1 type=general value=123456789

# Configure ISDN and PPP
add user=remote password=rempass login=no telnet=no
add isdn call=office number=0 prec=in dir=in searchsub=local
create ppp=0 over=isdn-office authentication=chap echo=30 lqr=off bap=off

# Configure IP
enable ip
add ip int=vlan1 ip=192.168.1.254
add ip int=ppp0 ip=10.0.0.1
add ip rou=192.168.2.0 int=ppp0 next=10.0.0.2

# Configure ISAKMP
create isakmp pol=office pe=10.0.0.2 key=1
enable isakmp

# Configure IPsec
create ipsec sas=1 key=isakmp prot=esp enc=des hasha=sha
create ipsec bund=1 key=isakmp string=1

# Create policies which allow ISAKMP to bypass IPsec processing, but other
# traffic to be processed by IPSec
create ipsec pol=isakmp int=ppp0 ac=permit lp=500 rp=500
create ipsec pol=office key=isakmp isa=office int=ppp0 ac=ipsec bund=1 peer=10.0.0.2
set ipsec pol=office lad=192.168.1.0 lma=255.255.255.0 rad=192.168.2.0
rma=255.255.255.0
enable ipsec

# Create a classifier for VoIP (This example assumes voice traffic uses UDP ports
# between 16300 and 16320)
create class=1 udpdport=16300-16320

# Create a classifier to match on the SIP signaling traffic
create class=2 udpdport=5060

# Create a classifier for traffic from the file server
create class=3 ipsadd=192.168.1.10

# Create a QoS policy
create sqos poli=1

# Create a traffic class for VoIP traffic and give it the highest priority and a
# short queue
create sqos trafficclass=1 priority=15 maxqlen=10
```

continued on next page...

.

```
# Guaranteeing VoIP traffic and maintaining file server downloads over a VPN tunnel
# Central office configuration continued

# Create a traffic class for the SIP signalling traffic and give it the second
# highest priority
create sqos trafficclass=2 priority=14

# Create a traffic class for file server traffic, give it the next highest priority
# and use a medium RED curve set
create sqos trafficclass=3 priority=13 red=1

# Add the traffic classes to the policy
add sqos policy=1 trafficclass=1-3

# Add the classifiers to the traffic classes
add sqos trafficclass=1 classifier=1
add sqos trafficclass=2 classifier=2
add sqos trafficclass=3 classifier=3

# Use the policy on the IPSec tunnel
set sqos interface=ipsec-office tunnelpolicy=1

# Enable software QoS
enable sqos
```

```
# Guaranteeing VoIP traffic and maintaining file server downloads over a VPN tunnel
# Remote office configuration

set system name=remote
set system territory=europe

# Create a user with Security Officer privilege and enable secure mode
add user=secoff pass=friend priv=securityOfficer lo=yes
set user=secoff telnet=no netmask=255.255.255.255
enable system security_mode
set user securedelay=600

# Create an encryption key for ISAKMP to protect and authenticate its messages
create enco key=1 type=general value=123456789

# Configure ISDN and PPP
add isdn call=office number=your-central-office-number prec=out outsub=local
create ppp=0 over=isdn-office username=remote passw=rempass echo=30 lqr=off bap=off

# Configure IP
enable ip
add ip interface=vlan1 ip=192.168.2.254
add ip interface=ppp0 ip=10.0.0.2
add ip route=192.168.1.0 int=ppp0 next=10.0.0.1

# Configure ISAKMP
create isakmp pol=office pe=10.0.0.1 key=1
enable isakmp

# Configure IPsec
create ipsec sas=1 key=isakmp prot=esp enc=des hasha=sha
create ipsec bund=1 key=isakmp string=1

# Create policies which allow ISAKMP to bypass IPsec processing, but other
# traffic to be processed by IPsec
create ipsec pol=isakmp int=ppp0 ac=permit lp=500 rp=500
create ipsec pol=office key=isakmp isa=office int=ppp0 ac=ipsec bund=1 peer=10.0.0.1
set ipsec pol=office lad=192.168.2.0 lma=255.255.255.0 rad=192.168.1.0
rma=255.255.255.0
enable ipsec

# Create a classifier for VoIP (This example assumes voice traffic uses UDP ports
# between 16300 and 16320)
create class=1 udpdport=16300-16320

# Create a classifier to match on the SIP signaling traffic
create class=2 udpdport=5060

# Create a classifier for traffic to the file server
create class=3 ipdadd=192.168.1.10

# Create a QoS policy
create sqos poli=1

# Create a traffic class for VoIP traffic and give it the highest priority and a
# short queue
create sqos trafficclass=1 priority=15 maxqlen=10
```

continued on next page...

```
# Guaranteeing VoIP traffic and maintaining file server downloads over a VPN tunnel
# Remote office configuration continued

# Create a traffic class for the SIP signalling traffic and give it the second highest
# priority
create sqos trafficclass=2 priority=14

# Create a traffic class for file server traffic, give it the next highest priority
# and use a medium RED curve set
create sqos trafficclass=3 priority=13 red=1

# Add the traffic classes to the policy
add sqos policy=1 trafficclass=1-3

# Add the classifiers to the traffic classes
add sqos trafficclass=1 classifier=1
add sqos trafficclass=2 classifier=2
add sqos trafficclass=3 classifier=3

# Use the policy on the ipsec tunnel
set sqos interface=ipsec-office tunnelpolicy=1

# Enable software QoS
enable sqos
```

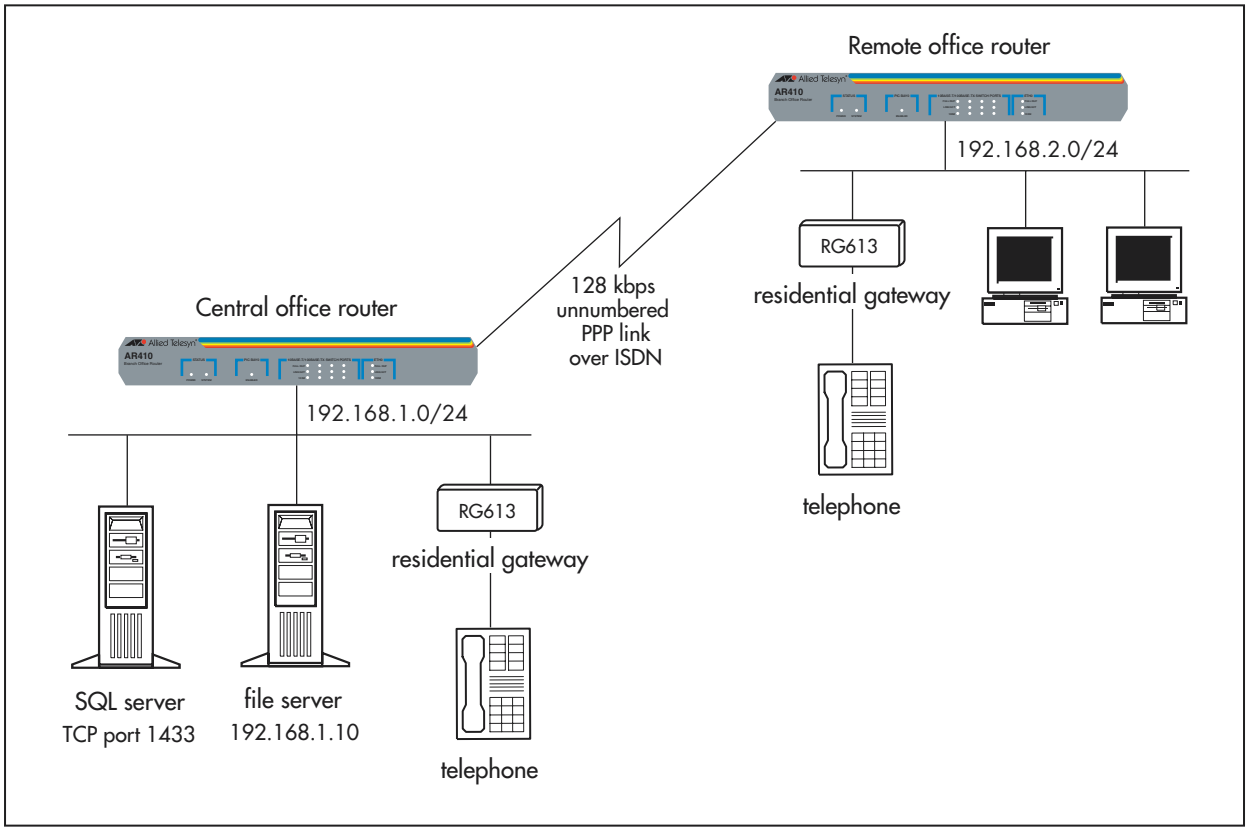
5: VoIP, Critical Database, and File Server Traffic

This scenario expands on Scenario 3 by adding a critical SQL database.

In this scenario (Figure 39-27):

- VoIP traffic has the highest priority possible and a short queue, so the routers always transmit VoIP traffic with low drop rate, delay, and jitter.
- the routers identify VoIP traffic by the UDP ports it uses.
- SQL traffic has the next highest priority. The routers use a medium RED curve set to drop SQL server traffic as necessary. We recommend RED because the traffic class may include multiple simultaneous flows to and from the server.
- file server traffic has the next highest priority, so is sent only when there is no VoIP or SQL traffic waiting. The routers use a medium RED curve set to drop file server traffic as necessary and control the TCP flows. We recommend RED because the traffic class may include multiple simultaneous flows to and from the file server.

Figure 39-27: Configuration for VoIP, critical database, and file server traffic



```
# Guaranteeing VoIP traffic and maintaining SQL and file server downloads
# Central office configuration

set system name=central
set system territory=europe

# Configure ISDN and PPP
add user=remote password=rempass login=no telnet=no
add isdn call=office number=0 prec=in dir=in searchsub=local
create ppp=0 over=isdn-office authentication=chap echo=30 lqr=off bap=off

# Configure IP
enable ip
add ip interface=vlan1 ip=192.168.1.254
add ip interface=ppp0 ip=0.0.0.0 mask=0.0.0.0
add ip route=192.168.2.0 int=ppp0 next=0.0.0.0

# Create classifiers for VoIP (this example assumes voice traffic uses UDP ports
# between 16300 and 16320), and to match the SIP signaling traffic
create class=1 udpdport=16300-16320
create class=2 udpdport=5060

# Create a classifier for traffic from the SQL server
create class=3 tcpsport=1433

# Create a classifier for traffic from the file server
create class=4 ipsadd=192.168.1.10

# Create a QoS policy
create sqos poli=1

# Create a traffic class for VoIP traffic and give it the highest priority and a
# short queue
create sqos trafficclass=1 priority=15 maxqlen=10

# Create a traffic class for the SIP signalling traffic and give it the second highest
# priority
create sqos trafficclass=2 priority=14

# Create a traffic class for SQL server traffic, give it the next highest priority
# and use a medium RED curve set
create sqos trafficclass=3 priority=13 red=1

# Create a traffic class for file server traffic, give it a lower priority
# and use a medium RED curve set
create sqos trafficclass=4 priority=12 red=1

# Add the traffic classes to the policy
add sqos policy=1 trafficclass=1-4

# Add the classifiers to the traffic classes
add sqos trafficclass=1 classifier=1
add sqos trafficclass=2 classifier=2
add sqos trafficclass=3 classifier=3
add sqos trafficclass=4 classifier=4

# Use the policy on ppp
set sqos interface=ppp0 outpolicy=1

# Enable software QoS
enable sqos
```

```
# Guaranteeing VoIP traffic and maintaining SQL and file server downloads
# Remote office configuration

set system name=remote
set system territory=europe

# Configure ISDN and PPP
add isdn call=office number=your-central-office-number prec=out outsub=local
create ppp=0 over=isdn-office username=remote passw=rempass echo=30 lqr=off bap=off

# Configure IP
enable ip
add ip interface=vlan1 ip=192.168.2.254
add ip interface=ppp0 ip=0.0.0.0 mask=0.0.0.0
add ip route=192.168.1.0 int=ppp0 next=0.0.0.0

# Create classifiers for VoIP (this example assumes voice traffic uses UDP ports
# between 16300 and 16320), and to match the SIP signaling traffic
create class=1 udpdport=16300-16320
create class=2 udpdport=5060

# Create a classifier for traffic to the SQL server
create class=3 tcpdport=1433

# Create a classifier for traffic to the file server
create class=4 ipdadd=192.168.1.10

# Create a QoS policy
create sqos poli=1

# Create a traffic class for VoIP traffic and give it the highest priority and a
# short queue
create sqos trafficclass=1 priority=15 maxqlen=10

# Create a traffic class for the SIP signalling traffic and give it the second highest
# priority
create sqos trafficclass=2 priority=14

# Create a traffic class for SQL server traffic, give it the next highest priority
# and use a medium RED curve set
create sqos trafficclass=3 priority=13 red=1

# Create a traffic class for file server traffic, give it a lower priority
# and use a medium RED curve set
create sqos trafficclass=4 priority=12 red=1

# Add the traffic classes to the policy
add sqos policy=1 trafficclass=1-4

# Add the classifiers to the traffic classes
add sqos trafficclass=1 classifier=1
add sqos trafficclass=2 classifier=2
add sqos trafficclass=3 classifier=3
add sqos trafficclass=4 classifier=4

# Use the policy on ppp
set sqos interface=ppp0 outpolicy=1

# Enable software QoS
enable sqos
```

6: Multiple Applications over Frame Relay

This scenario shows multiple applications running over a 1.5 Mbps frame relay link, including voice, video conferencing, network monitoring, and server traffic.

In this scenario (Figure 39-28):

- a traffic class tree prioritises voice, video conferencing, and network monitoring traffic, higher than server downloads, while weighting the server downloads so that both SQL and file server data can be sent (Figure 39-29 on page 39-83).
- VoIP traffic has the highest priority possible and a short queue, so the routers always transmit VoIP traffic with low drop rate, delay, and jitter. Voice has a minimum bandwidth guarantee of 300 kbps, which allows 6-8 calls at once, and is limited to 500 kbps.
- video conferencing traffic has the next highest priority. It is limited to 250 kbps, which allows only one video conference at a time.
- network monitoring traffic (SNMP) has the next highest priority.
- SQL and file server traffic, combined, have the next highest priority, so is only sent if there is no VoIP, video, or network monitoring traffic waiting. SQL and file server traffic share a weighted intermediate traffic class, with SQL traffic having 70% of the weight and file server traffic 30%. This prevents the SQL traffic from throttling the file server downloads.
- the routers use a medium RED curve set to drop SQL and file server traffic as necessary. We recommend RED because the traffic classes may include multiple simultaneous flows to and from the servers.

Figure 39-28: Configuration for software QoS with multiple applications

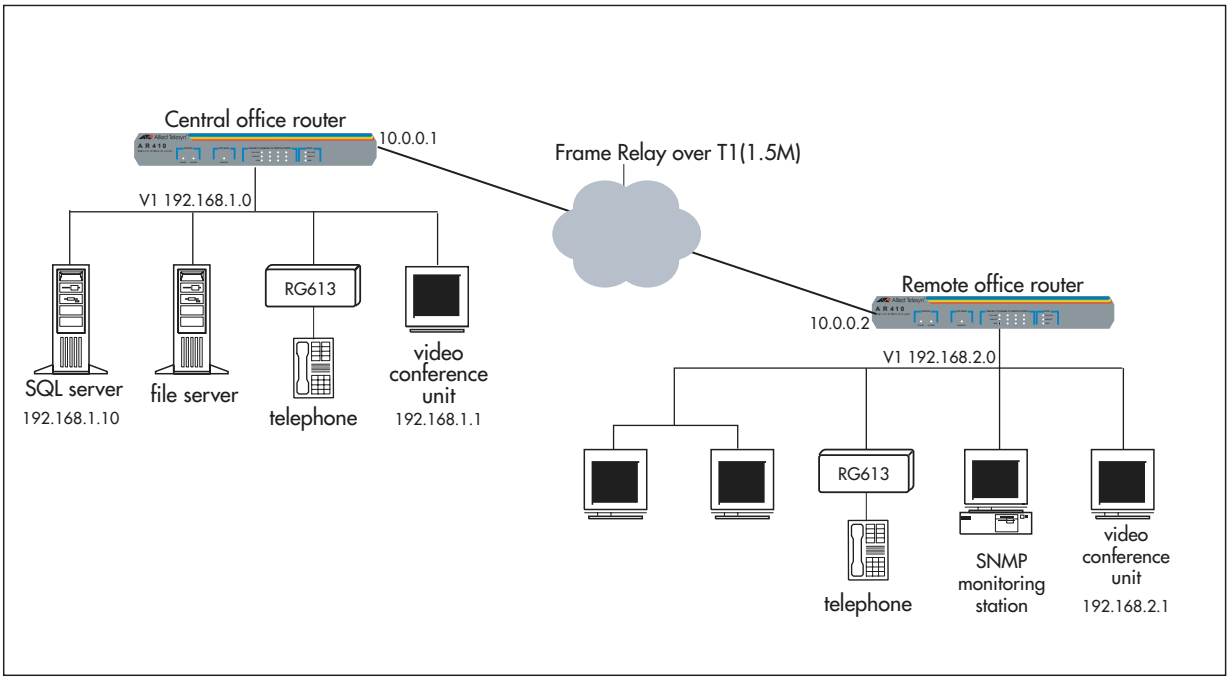
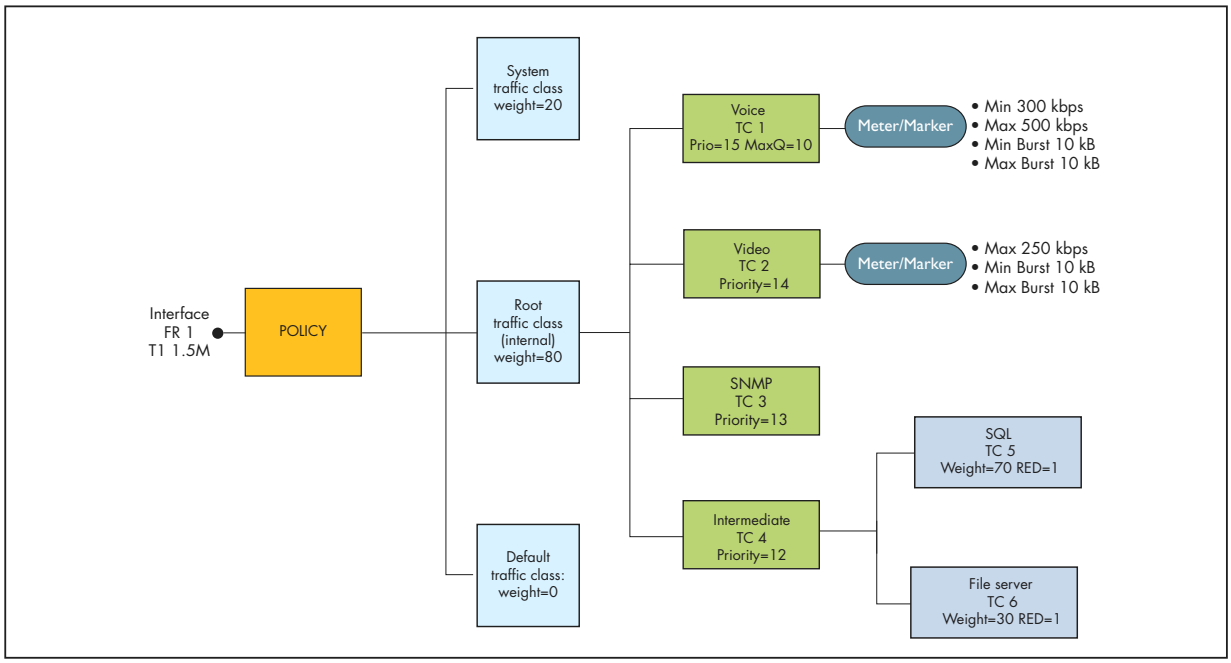


Figure 39-29: Traffic class tree for multiple applications over a frame relay link




```
# VoIP, Video, SNMP and Data traffic between two offices using FR/T1 (1.5M) link
# Central office configuration

set system name=central

# Configure the Frame Relay interface over the T1 (1.5M) PRI link
set pri=bay0.pri0 mode=tdm
set pri=bay0.pri0 cl=int

create tdm group=office interface=bay0.pri0 unstructured

create fr=1 over=office lmischeme=none
add fr=1 li=301 type=ptp
add fr=1 dlc=301
set fr=1 dlc=301 li=301

# Configure IP
enable ip
add ip interface=vlan1 ip=192.168.1.254
add ip interface=fr1.301 ip=10.0.0.1
add ip route=192.168.2.0 int=fr1.301 next=10.0.0.2

# Create a classifier for VoIP (This example assumes voice traffic uses UDP ports
# between 16300 and 16320)
create class=1 udpdport=16300-16320

# Create a classifier to match on the SIP signaling traffic
create class=2 udpdport=5060

# Create a classifier for the video conferencing traffic
create class=3 ipsadd=192.168.1.1

# Create a classifier for the SNMP traffic
create class=4 udpdport=161
```

continued on next page...

```
# VoIP, Video, SNMP and Data traffic between two offices using FR/T1 (1.5M) link
# Central office configuration continued

# Create a classifier for traffic from the SQL server
create class=5 tcpsport=1433

# Create a classifier for traffic from the file server
create class=6 ipsadd=192.168.1.10

# Create a QoS policy
create sqos poli=1

# Create meters for the voice and video traffic
create sqos meter=1 descrip=voice type=trtcm min=300K max=500K minbu=10K maxbu=10K
create sqos meter=2 descrip=video type=srtcm max=250K minbu=10K maxbu=10K

# Create a traffic class for VoIP/SIP traffic and give it the highest priority and
# a short queue. Assign the meter and set it to drop traffic over the maximum
# bandwidth setting
create sqos trafficclass=1 priority=15 maxqlen=10 meter=1 bwclass3action=drop

# Create a traffic class for video traffic and give it the second highest priority.
# Assign the meter and set it to drop traffic over the maximum bandwidth setting
create sqos trafficclass=2 priority=14 meter=2 bwclass3action=drop

# Create a traffic class for SNMP traffic and give it the next highest priority
# after the real-time voice and video
create sqos trafficclass=3 priority=13

# Create an intermediate traffic class which will have sub traffic classes for SQL
# and file server traffic attached.  Give it a lower priority and set it to use WRR
# to send SQL and file server traffic
create sqos trafficclass=4 priority=12 weightscheduler=wrr

# Create a traffic class for SQL server traffic, give it a higher weight than the
# file server traffic class and use a medium RED curve set
create sqos trafficclass=5 weight=70 red=1

# Create a traffic class for file server traffic, give it a lower weight than the
# SQL traffic class and use a medium RED curve set
create sqos trafficclass=6 weight=30 red=1

# Add the traffic classes to the policy
add sqos policy=1 trafficclass=1-4

# Add the sub traffic classes for SQL and File server traffic to intermediate traffic
# class 4
add sqos trafficclass=4 subclass=5,6

# Add the classifiers to the traffic classes
add sqos trafficclass=1 classifier=1,2
add sqos trafficclass=2 classifier=3
add sqos trafficclass=3 classifier=4
add sqos trafficclass=5 classifier=5
add sqos trafficclass=6 classifier=6

# Use the policy on FR1
set sqos interface=fr1 outpolicy=1

# Enable software QoS
enable sqos
```

```
# VoIP, Video, SNMP and Data traffic between two offices using FR/T1 (1.5M) link
# Remote office configuration

set system name=remote

# Configure the Frame Relay interface over the T1 (1.5M) PRI link
set pri=bay0.pri0 mode=tdm

create tdm group=office interface=bay0.pri0 unstructured

create fr=1 over=office lmischeme=none
add fr=1 li=301 type=ptp
add fr=1 dlc=301
set fr=1 dlc=301 li=301

# Configure IP
enable ip
add ip interface=vlan1 ip=192.168.1.254
add ip interface=fr1.301 ip=10.0.0.2
add ip route=192.168.1.0 int=fr1.301 next=10.0.0.1

# Create a classifier for VoIP (This example assumes voice traffic uses UDP ports
# between 16300 and 16320)
create class=1 udpdport=16300-16320

# Create a classifier to match on the SIP signaling traffic
create class=2 udpdport=5060

# Create a classifier for the video conferencing traffic
create class=3 ipsadd=192.168.2.1

# Create a classifier for the SNMP traffic
create class=4 udpdport=161

# Create a classifier for traffic to the SQL server
create class=5 tcpdport=1433

# Create a classifier for traffic to the file server
create class=6 ipdadd=192.168.1.10

# Create a QoS policy
create sqos poli=1

# Create meters for the voice and video traffic
create sqos meter=1 descrip=voice type=trtcm min=300K max=500K minbu=10K maxbu=10K
create sqos meter=2 descrip=video type=srtcm max=250K minbu=10K maxbu=10K

# Create a traffic class for VoIP/SIP traffic and give it the highest priority and
# a short queue. Assign the meter and set it to drop traffic over the maximum
# bandwidth setting
create sqos trafficclass=1 priority=15 maxqlen=10 meter=1 bwclass3action=drop

# Create a traffic class for Video traffic and give it the second highest priority.
# Assign the meter and set it to drop traffic over the maximum bandwidth setting
create sqos trafficclass=2 priority=14 meter=2 bwclass3action=drop
```

continued on next page...

```
# VoIP, Video, SNMP and Data traffic between two offices using FR/T1 (1.5M) link
# Remote office configuration continued

# Create a traffic class for the SNMP traffic, giving it the next highest priority
# after the real-time voice and video
create sqos trafficclass=3 priority=13

# Create an intermediate traffic class which will have sub traffic classes for SQL
# and data traffic attached.Give it a lower priority and set it to use WRR to send
# SQL and data
create sqos trafficclass=4 priority=12 weightscheduler=wrr

# Create a traffic class for SQL server traffic, give it a higher weight than the
# file server traffic class and use a medium RED curve set
create sqos trafficclass=5 weight=70 red=1

# Create a traffic class for File server traffic, give it a lower weight than the
# SQL traffic class and use a medium RED curve set
create sqos trafficclass=6 weight=30 red=1

# Add the traffic classes to the policy
add sqos policy=1 trafficclass=1-4

# Add the sub traffic classes for SQL and file server traffic to intermediate traffic
# class 4
add sqos trafficclass=4 subclass=5,6

# Add the classifiers to the traffic classes
add sqos trafficclass=1 classifier=1,2
add sqos trafficclass=2 classifier=3
add sqos trafficclass=3 classifier=4
add sqos trafficclass=5 classifier=5
add sqos trafficclass=6 classifier=6

# Use the policy on FR1
set sqos interface=fr1 outpolicy=1

# Enable software QoS
enable sqos
```

Command Reference

This section describes the commands available for configuring and monitoring Software QoS on the router.

The shortest valid command is denoted by capital letters in the Syntax section. See "Conventions" on page -lxv of , About this Software Reference in the front of this manual for details of the conventions used to describe command syntax. See Appendix A, Messages, for a complete list of messages and their meanings.

add sqos interface dar

Syntax `ADD SQOS INTerface=`*`interface`* `DAR=`*`id-list`*

Description This command adds one or more DAR (Dynamic Application Recognition) objects to the interface, which should be the interface at which voice or video session initiation control messages arrive. The router uses the DAR object to identify matching session initiation control messages, and creates dynamic classifiers for the associated voice or video flow.

Parameter	Description
INTerface	Interface or tunnel to which to add the DAR object. Valid entry types are: Layer 1 and 2 interfaces: ● eth (such as. eth0) ● ATM channel (such as atm0.0) ● frame relay (such as fr0) ● PPP (such as ppp0) ● the switch instance (such as swi0) Layer 3 tunnels: ● GRE (such as gre1) ● IPv6 6-to-4 virtual interface (such as virt9) ● the name of an IPSec policy (such as ipsec-*policyname*) To see a list of current valid Layer 1 and 2 interfaces, use the **show interface** command on page 9-72 of Chapter 9, Interfaces.
DAR	DAR object to add to the interface. *id-list* is an integer from 0 to 9999, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once. The DAR objects must already exist. Each DAR object can only belong to one interface.

Examples To attach DAR object 0 to ppp0, use the command:

```
add sqos int=ppp0 dar=0
```

To attach DAR object 0 to the IPsec policy *central*, use the command:

```
add sqos int=ipsec-central dar=0
```

Related Commands

add sqos trafficclass dar
create sqos dar
delete sqos interface dar
destroy sqos dar
set sqos dar
set sqos interface
show sqos dar

add sqos policy trafficclass

Syntax `ADD SQOS POLIcy=0..9999 TRafficclass=`*`id-list`*

Description This command adds one or more traffic classes to the specified QoS policy.

Parameter	Description
POLIcy	Policy to which to add the traffic class. An integer from 0 to 9999. The policy must already exist.
TRafficclass	Traffic class to add to the policy. *id-list* is an integer from 0 to 9999, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once. The traffic classes must already exist. Each traffic class can only belong to one policy.

Example To create a traffic class hierarchy (shown in Figure 39-30 on page 39-91) in which:

- policy 1 contains leaf traffic class 1 and intermediate traffic class 2
- intermediate traffic class 2 contains intermediate traffic class 3 and leaf traffic class 4
- intermediate traffic class 3 contains leaf traffic classes 5 and 6

first create the traffic classes and the policy, then combine them into the hierarchy using the commands:

```
add sqos poli=1 tr=1,2
add sqos tr=2 subc=3,4
add sqos tr=3 subc=5,6
```

Related Commands

create sqos policy
create sqos trafficclass
delete sqos policy trafficclass
destroy sqos policy
destroy sqos trafficclass
set sqos policy
set sqos trafficclass
show sqos policy
show sqos trafficclass

add sqos trafficclass classifier

Syntax `ADD SQOS TRafficclass=0..9999 CLASSifier=`*`id-list`*

Description This command adds one or more classifiers to the specified traffic class. The traffic class must be a leaf traffic class; you cannot also add sub traffic classes to it.

Parameter	Description
TRafficclass	Leaf traffic class to which to add the classifier. An integer from 0 to 9999. The traffic class must already exist.
CLASSifier	Classifier to add to the traffic class. *id-list* is an integer from 0 to 9999, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once. The classifiers must already exist. Each classifier can only belong to one traffic class.

Example To add classifiers 1, 2, 4, 5, and 6 to leaf traffic class 3, use the command:

```
add sqos tr=3 class=1,2,4-6
```

Related Commands

add sqos trafficclass classifier
create sqos trafficclass
delete sqos trafficclass classifier
destroy sqos trafficclass
set sqos trafficclass
show sqos trafficclass

add sqos trafficclass dar

Syntax `ADD SQOS TRafficclass=0..9999 DAR=`*`id-list`*

Description This command associates one or more DAR (Dynamic Application Recognition) objects with a traffic class. The traffic class then processes voice or video traffic flows that the DAR objects identify.

The traffic class must be a leaf traffic class; you cannot also add sub traffic classes to it.

Parameter	Description
TRafficclass	Leaf traffic class to which to add the DAR object. An integer from 0 to 9999. The traffic class must already exist.
DAR	DAR object to add to the traffic class. *id-list* is an integer from 0 to 9999, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once. The DAR objects must already exist. Each DAR object can only belong to one traffic class.

Example To add DAR objects 1, 2, 4, 5, and 6 to leaf traffic class 3, use the command:

```
add sqos tr=3 dar=1,2,4-6
```

Related Commands

add sqos interface dar
create sqos dar
create sqos trafficclass
delete sqos interface dar
delete sqos trafficclass dar
destroy sqos dar
set sqos dar
set sqos trafficclass
show sqos dar
show sqos trafficclass

add sqos trafficclass subclass

Syntax

```
ADD SQOS TRafficclass=0..9999 SUBClass=id-list
```

Description

This command adds one or more traffic classes to sub traffic class to create a traffic class hierarchy.

Parameter	Description
TRafficclass	Intermediate traffic class to which to add a sub traffic class. An integer from 0 to 9999. The intermediate traffic class must already exist and must be attached to a policy.
SUBClass	Sub traffic class to add to the intermediate traffic class. *id-list* is an integer from 0 to 9999, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once. The sub traffic classes must already exist. Each sub traffic class can only belong to one intermediate traffic class.

Example

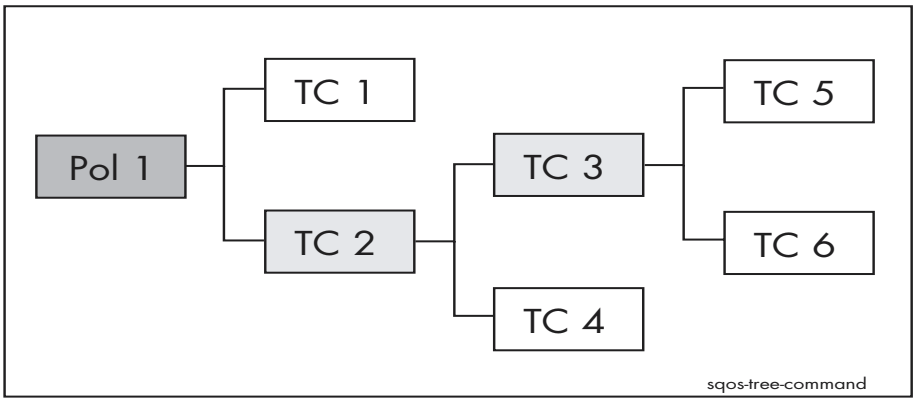
To create a traffic class hierarchy (shown in Figure 39-30) in which:

- policy 1 contains leaf traffic class 1 and intermediate traffic class 2
- intermediate traffic class 2 contains intermediate traffic class 3 and leaf traffic class 4
- intermediate traffic class 3 contains leaf traffic classes 5 and 6

first create the traffic classes and the policy, then combine them into the hierarchy using the commands:

```
add sqos poli=1 tr=1,2
add sqos tr=2 subc=3,4
add sqos tr=3 subc=5,6
```

Figure 39-30: Example of a traffic class tree



Related Commands

add sqos policy trafficclass
add sqos trafficclass classifier
add sqos trafficclass dar
create sqos trafficclass
delete sqos trafficclass subclass
destroy sqos trafficclass
set sqos trafficclass
show sqos trafficclass

create sqos dar

Syntax

```
CREate SQOS DAR=id-list [CODEC={AUDio|VIDeo|ANY}]
   [DESCription=description] [DSTIp={ipadd[/
   0..32]|ipv6add[/0..128]}] [SRCIp={ipadd[/
   0..32]|ipv6add[/0..128]}]
   [INACtivetimeout={1..3600|NONE}]
   [PROTocol={SIP|RTSp|H323|ALL}]
   [H323Port=1..65535] [RTSPPort=1..65535]
   [SIPPort=1..65535]
```

Description

This command creates one or more Dynamic Application Recognition objects. The router identifies voice or video packets that match the settings in the DAR object and dynamically creates a classifier for that traffic flow.

You can create up to 64 DAR objects.

After you have created the DAR object, use the **add sqos trafficclass dar** command to assign it to a leaf traffic class, and the **add sqos interface dar** command to assign it to the interface at which voice or video session initiation control packets arrive.

Parameter	Description	
DAR	ID number of the new DAR object. *id-list* is an integer from 0 to 9999, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once.	
CODEC	Coder/decoder for the DAR object to use to match packets. Default: **any**	
	AUDio	Matches traffic flows that use any audio codec.
	VIDeo	Matches traffic flows that use any video codec.
	ANY	The DAR object ignores the codec. A DAR object with **codec=any** matches any sessions set up by the specified protocol, not just voice and video sessions.
DESCription	Description of the DAR object, which has no effect on its operation. A string 1 to 100 characters long. All printable characters are valid. If *description* contains spaces, it must be in double quotes. Default: no default	
DSTIp	Destination IPv4 or IPv6 address or subnet. The DAR object matches voice or video traffic flows to that address or network only. IPv4 addresses are specified in dotted decimal notation. IPv6 addresses are specified as eight pairs of octets, separated by colons. The CIDR mask for IPv4 and prefix length for IPv6 are optional. For IPv4, if you specify a subnet address without specifying a mask, the default mask for that subnet is used. For IPv6, the default prefix length is 128. If you also specify **srcip**, either both must be IPv4 addresses or both must be IPv6 addresses. Default: no default (ignores destination IP address)	
H323Port	TCP port that H.323 session control messages are received on. Default: **1720**	

Parameter	Description (cont.)
INACtivetimeout	Time from 1 to 3600 seconds (up to 60 minutes). If a classified flow is idle for this length of time, its entry is deleted. Default: **600**
PROTocol	Protocol for the DAR object to use to match packets. Default: **all** (ignores protocol)
RTSPPort	TCP port that RTSP session control messages are received on. Default: **554**
SIPPort	UDP port that SIP messages are received on. Default: **5060**
SRCIp	Source IPv4 or IPv6 address or subnet. The DAR object only matches voice or video traffic flows from that address or network. IPv4 addresses are specified in dotted decimal notation. IPv6 addresses are specified as eight pairs of octets, separated by colons. The CIDR mask for IPv4 and prefix length for IPv6 are optional. For IPv4, if you specify a subnet address without specifying a mask, the default mask for that subnet is used. For IPv6, the default prefix length is 128. If you also specify **dstip**, either both must be IPv4 addresses or both must be IPv6 addresses. Default: no default (ignores source IP address)

Example To create a DAR object to identify and classify voice packets destined for the 192.168.1.0 subnet, use the command:

```
cre sqos dar=0 codec=audio dsti=192.168.1.0/24
```

Related Commands

add sqos interface dar
add sqos trafficclass dar
delete sqos interface dar
destroy sqos dar
set sqos dar
show sqos dar

create sqos dscpmap

Syntax `CREate SQOS DSCPMap=`*`id-list`* `[DESCription=`*`description`*`]`

Description This command creates up to 64 DSCP maps with the default settings. The map's default settings are to leave all values unchanged. Use the **set sqos dscpmap** command to change defaults.

DSCP maps consist of a premarking table and a re-marking table. Software QoS uses the premarking table before the metering stage, to map a packet's DSCP value to a new DSCP and/or bandwidth class. The map's default settings are to map all DSCPs to bandwidth class 1 (green). Software QoS uses the re-marking table after the metering stage. It uses the bandwidth class the meter assigned the packet to, and the packet's DSCP value, to give the packet a new DSCP and/or bandwidth class.

Parameter	Description
DSCPMap	ID number of the new DSCP map. *id-list* is an integer from 0 to 9999, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 1,3,4-9). An integer cannot appear in the list more than once.
DESCription	Description of the DSCP map, which has no effect on its operation. A string 1 to 100 characters long. All printable characters are valid. If *description* contains spaces, it must be in double quotes. Default: no default

Example To create a DSCP map to use to re-mark non-conformant traffic with a DSCP of 10, use the commands:

```
cre dscpm=1 desc=nonconformant_10
set dscpm=1 table=rem bwc=3 newd=10
```

Related Commands
create sqos trafficclass
destroy sqos dscpmap
set sqos dscpmap
show sqos dscpmap

create sqos meter

Syntax

```
CREate SQOS METer=id-list [DESCription=description]
   [IGNorebwclass={Yes|No}]
   [MINbandwidth=rate[Kbps|Mbps|Gbps]]
   [MAXbandwidth=rate[Kbps|Mbps|Gbps]]
   [MINBUrstsize=burstsize[Bytes|Kbytes|Mbytes|Gbytes]]
   [MAXBUrstsize=burstsize[Bytes|Kbytes|Mbytes|Gbytes]]
   [TYPE={SRtcm|TRtcm}]
```

Description

This command creates up to 64 Three Colour Marker meters, as described in RFC 2697, *A Single Rate Three Color Marker* and RFC 2698, *A Two Rate Three Color Marker*. The meter measures how much bandwidth packets in a traffic flow use, and how well bandwidth usage conforms with the specifications for the traffic class to which the flow belongs. It assigns the packet to a bandwidth class depending on its conformance. For more information, see Bandwidth Class on page 20.

Parameter	Description	
METer	ID number of the new meter. *id-list* is an integer from 0 to 9999, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once.	
DESCription	Description of the meter, which has no effect on its operation. A string 1 to 100 characters long. All printable characters are valid. If *description* contains spaces, it must be in double quotes. Default: no default	
IGNorebwclass	Whether the meter acknowledges any previous bandwidth class assigned to packets. A meter that acknowledges previous conformance is called *colour aware*. Default: **no** (the meter is colour aware)	
	Yes	The metering function is colour blind and ignores any bandwidth class previously assigned to packets. It sets the meter bandwidth class according to only the metered conformance level of the flow.
	No	The metering function is colour aware and uses any bandwidth class previously assigned to packets, as well as the metered conformance level, to set the bandwidth class. Packets previously labelled non-conformant (bandwidth class 3, red) remain in bandwidth class 3. Packets previously labelled partially-conformant (bandwidth class 2, yellow) are assigned to bandwidth class 2 or 3, depending on metered conformance.

Parameter	Description (cont.)
MAXbandwidth	For the single rate meter of RFC 2697, the highest rate at which a steady stream of packets can arrive at the meter and be assigned to bandwidth class 1 (conformant, green). This is the Committed Information Rate (CIR) of the RFC. For the two rate meter of RFC 2698, the Peak Information Rate (PIR) of the RFC. See "Metering: Bandwidth Conformance" on page 39-21 for a description of PIR. It must equal or exceed **minbandwidth**. *rate* is from 0 to 16000000 kilobits per second, specified in Kbps, Mbps or Gbps (in upper or lower case). If you do not specify a unit, it uses kbps. If you specify Mbps or Gbps, *rate* may contain a decimal fraction with up to 3 decimal places, for example, 1.25 Mbps. Default: **1Mbps**
MAXBUrstsize	For the single rate meter of RFC 2697, the amount by which a packet can exceed **maxbandwidth** plus **minburstsize** and still possibly be assigned to bandwidth class 2. This is the Excess Burst Size (EBS) of the RFC. For the two rate meter of RFC 2698, the amount by which a packet can exceed **maxbandwidth** and still possibly be assigned to bandwidth class 2. This is the Peak Burst Size (PBS) of the RFC. *burstsize* is from 0 to 16777216 bytes (16 MB), specified in bytes, kbytes, Mbytes, or Gbytes (in upper or lower case). If you do not specify a unit, it uses bytes. If you specify kB, MB or GB, *burstsize* may contain a decimal fraction, for example, 1.25M. For a single rate meter, at least one of **minburstsize** and **maxburstsize** should equal or exceed the size of the largest IP packet you expect on the metered flow. For a two rate meter, both **minburstsize** and **maxburstsize** must exceed zero, and the RFC recommends that both values equal or exceed the size of the largest IP packet you expect on the metered flow. To create a single rate two colour meter (green and red), set **maxburstsize** to 0 (zero). Default: **10kbytes**
MINbandwidth	For the two rate meter of RFC 2698, the Committed Information Rate (CIR) of the RFC. See "Metering: Bandwidth Conformance" on page 39-21 for a description of CIR. It must not exceed **maxbandwidth**. *rate* is from 0 to 16000000 kilobits per second, specified in Kbps, Mbps or Gbps (in upper or lower case). If you do not specify a unit, it uses kbps. If you specify Mbps or Gbps, *rate* may contain a decimal fraction with up to 3 decimal places, for example, 1.25 Mbps. Only valid if **type**=**trtcm.** Default: **1Mbps**

Parameter	Description (cont.)
MINBUrstsize	For the single rate meter of RFC 2697, the amount by which a packet can exceed **maxbandwidth** and still possibly be assigned to bandwidth class 1. This is the Committed Burst Size (CBS) of the RFC. For the two rate meter of RFC 2698, the amount by which a packet can exceed **minbandwidth** and still possibly be assigned to bandwidth class 1. This is the Committed Burst Size (CBS) of the RFC. *burstsize* is from 0 to 16777216 bytes (16 MB), specified in bytes, kbytes, Mbytes, or Gbytes (in upper or lower case). If you do not specify a unit, it uses bytes. If you specify kB, MB or GB, *burstsize* may contain a decimal fraction, for example, 1.25M. For a single rate meter, at least one of **minburstsize** and **maxburstsize** should equal or exceed the size of the largest IP packet you expect on the metered flow. For a two rate meter, both **minburstsize** and **maxburstsize** must exceed 0 (zero), and the RFC recommends that both values equal or exceed the size of the largest IP packet you expect on the metered flow. For a single rate meter, if you set **minburstsize** to 0 (zero) the meter assigns all packets to bandwidth class 2 or 3. Default: **10kbytes**
TYPE	The type of meter. For a description of meters, see "Metering: Bandwidth Conformance" on page 39-21. Default: SRTCM
	SRTCM The Single Rate Three Colour Marker of RFC 2697.
	TRTCM The Two Rate Three Colour Marker of RFC 2698.

Example To create a colour-blind single-rate meter with default settings otherwise, use the command:

```
cre sqos met=0 ign=y
```

Related Commands

create sqos trafficclass
destroy sqos meter
set sqos meter
show sqos meter

create sqos policy

Syntax

```
CREate SQOS POLIcy=id-list
   [BWClass3action={DROP|PAUSE|NONE}]
   [DEFaulttrafficclass={0..9999|NONE}]
   [DESCription=description] [DSCPMap={0..9999|NONE}]
   [IGNOREPrenatinfo={YES|NO}][METer={0..9999|NONE}]
   [PAUSEAction={NONE|LOg|TRap|BOth}] [PAUSETIme={1..30}]
   [REMarking={0..63|USEDscpmap|NONE}]
   [REMARKVlanpri={0..7|NONE}] [SYSTEMTraffic={5..50}]
   [VIRTbw={rate[Kbps|Mbps|Gbps]|NONE}]
   [WEIGhtscheduler={WRr|DWrr}]
```

Description This command creates one or more QoS policies. A policy defines the overall QoS processing for an interface. You can create up to 64 policies.

After you have created the policy, use the **set sqos interface** command on page 39-125 to assign it to the required interface.

Parameter	Description	
POLIcy	ID number of the new policy. *id-list* is an integer from 0 to 9999, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once.	
BWClass3action	Action the router takes on Bandwidth Class 3 packets (red coloured packets). These are packets that exceed their allocated bandwidth, as determined at the metering stage. Default: **none**	
	DROP	The router drops non-conformant packets.
	PAUSE	The router drops non-conformant packets and stops dequeuing packets from the flow for **pausetime** seconds.
	NONE	The router sends non-conformant packets to the next processing stage.
DEFaulttrafficclass	Traffic class that the router applies to unclassified traffic on the policy's interface. It must be a leaf traffic class. Default: no default	
	0..9999	The traffic class ID.
	NONE	No user-nominated default traffic class. The router uses the default traffic class that it made when you created the policy.
DEscription	A description of the policy, which has no effect on its operation. A string 1 to 100 characters long. All printable characters are valid. If *description* contains spaces, it must be in double quotes. Default: no default	
DScpmap	The DSCP map to assign to the policy. An integer from 0 to 9999. Default: **none**	
IGNOREPrenatinfo	Whether classifiers attached to the policy use pre-NAT IP settings for classification because these contain the distinguishing information. Default: **no** (uses pre-NAT settings)	

Parameter	**Description** (cont.)	
MEter	Meter from 0 to 9999 to assign to the policy. The meter determines a new bandwidth class (colour) for packets that are processed using this policy. You can configure the policy or a traffic class to drop or queue the packets on the basis of the new bandwidth class. Default: **none**	
PAUSEAction	Notification action the router takes when it pauses a non-conformant traffic flow that belongs to this policy. Only valid if **bwclass3action**=**pause**. Default: **none**	
	LOg	The router generates a log message.
	TRap	The router generates an SNMP trap.
	BOth	The router generates both a log message and an SNMP trap.
	NONE	The router does not generate a notification.
PAUSETIme	The length of time, from 1 to 30 seconds, for which the router does not dequeue packets from a paused flow. Only valid if **bwclass3action**=**pause**. Default: **10**	
REMarking	How the router sets the bandwidth class and/or DSCP value in the packet header's Differentiated Services field after metering. Default: **none**	
	0..63	The router writes the specified value into the DSCP bits in the packet header.
	USEDscpmap	The router uses the metered bandwidth class and current DSCP value, in conjunction with the policy's DSCP map, to determine the new DSCP value and/or bandwidth class. You must also specify the **dscpmap** parameter.
	NONE	The router does not modify the DSCP value or metered bandwidth class.
REMARKVlanpri	Setting for the 802.1p VLAN priority field of the frame's Ethernet header. Default: **none**	
	0..7	The router writes the specified value into the 802.1p VLAN priority field of the Ethernet header.
	NONE	The router does not modify the 802.1p VLAN priority field of the Ethernet header.
SYSTEMTraffic	Percentage of the interface's maximum bandwidth from 5 to 50% that the router reserves for system traffic. Default: **20**	

Parameter	Description (cont.)	
VIRTbw	Maximum bandwidth available to the policy. **Virtbw** determines the maximum rate at which data can leave the internal queues to be transmitted onto the physical media. This rate is not equivalent to the transmission rate for data seen on the line, because the actual transmission rate includes the transmission of bits for the inter-frame-gap and the preamble of the Layer 2 headers. For example, 10 Mbps of data leaving the internal queues is not equivalent to 10 Mbps of data transmitted on the line. *rate* is from 0 to 16000000 kilobits per second, specified in Kbps, Mbps or Gbps (in upper or lower case). If you do not specify a unit, it uses kbps. If you specify Mbps or Gbps, *rate* may contain a decimal fraction with up to 3 decimal places, for example, 1.25 Mbps. Default: **none** (bandwidth is not limited)	
WEIGhtscheduler	Queue scheduling method for weighted traffic classes that belong to the policy. Weighted traffic classes assign weights to flows instead of priorities. Default: **wrr**	
	WRr	The router uses a weighted round robin scheme to empty the queues of weighted traffic classes.
	DWrr	The router uses a deficit weighted round robin scheme to empty the queues of weighted traffic classes. DWRR is less biased towards large packets than WRR.

Example To create a policy that allocates 15% of the available bandwidth to system traffic, use the command:

```
cre sqos poli=0 systemt=15
```

To create a policy that uses meter 1 to measure bandwidth, drops non-conformant packets, and uses DSCP map 1 to re-mark conformant packets, use the command:

```
cre sqos poli=0 bwc=drop dscpm=1 rem=used
```

Related Commands

add sqos policy trafficclass
delete sqos policy trafficclass
destroy sqos policy
set sqos interface
set sqos policy
show sqos policy

create sqos red

Syntax

```
CREate SQOS RED=id-list
   [AVERaging=0..99] [DESCription=description]
   [START1=0..100] [STOP1=0..100] [DROP1=0..100]
   [START2=0..100] [STOP2=0..100] [DROP2=0..100]
   [START3=0..100] [STOP3=0..100] [DROP3=0..100]
```

Description

This command creates one or more sets of RED curves. Red curve sets 0-2 exist by default, and cannot be modified or deleted. You can create up to 61 more RED curve sets. Table 39-3 on page 39-31 shows the properties of the default red curve sets.

Parameter	Description
RED	ID number of the new RED curve. *id-list* is an integer from 3 to 9999, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 3,4-9). An integer cannot appear in the list more than once.
AVERaging	Weight used in the moving averaging estimation of queue length for the RED curve algorithm. The estimated queue length is frequently updated, and is calculated by taking a weighted average of the previous average and the current instantaneous queue length. **Averaging** is the weight given to the previous average in this weighted calculation. If **averaging** is too high, the estimated average queue size responds too slowly to transient congestion. If **averaging** is too low, the estimated average queue size tracks the instantaneous queue size too closely and you lose the benefits of RED. RED works best when the estimated average queue length responds as slowly as possible while preventing the queue from becoming full. To achieve this, set **averaging** to a lower value if the queue constantly becomes full, so that the estimated average queue size more closely tracks the actual queue size. To check how often the queue becomes full, use the **trafficclass** parameter of the **show sqos counters** command on page 39-140 and check the queue counters, or set **qlimitexceedaction** and check the log messages or SNMP traps. Default: **98**
DESCription	An optional description of the RED curve set, which has no effect on its operation. A string from 1 to 100 characters long. All printable characters are valid. If *description* contains spaces, it must be in double quotes. Default: no default
START1 START2 START3	Percentage of the queue length from 1% to 100% at which the RED algorithm starts to drop packets for packets in bandwidth classes 1, 2, and 3 respectively. Default: **35**
STOP1 STOP2 STOP3	Percentage of the queue length from 1% to 100% at which the RED algorithm is dropping **drop** percent of the packets, for packets in bandwidth classes 1, 2, and 3 respectively. Beyond this point, 100% of the packets are dropped. This value must be greater than **start**. Default: **65**
DROP1 DROP2 DROP3	Probability from 1% to 100% that a packet will be dropped at the **stop** queue length for packets in bandwidth classes 1, 2, and 3 respectively. Default: **30**

Example To create a moderately-passive RED curve set, use the command:

```
cre red=3 desc=mod-passive
   start1=65 stop1=85 drop1=20
   start2=40 stop2=65 drop2=30
   start3=30 stop3=45 drop3=40
```

Related Commands

create sqos trafficclass
destroy sqos red
set sqos red
show sqos red

create sqos trafficclass

Syntax

```
CREate SQOS TRafficclass=id-list
   [BWClass3action={DRop|PAuse|NONE}]
   [DESCription=description] [MAXQlen=1..1023]
   [METer={0..9999|NONE}]
   [PAUSEAction={NONE|LOg|TRap|BOth}] [PAUSETIme={1..30}]
   [PREMARKBwcl={1..3|USEDscpmap}]
   [PREMARKDscp={0..63|USEDscpmap|NONE}]
   [{PRIOrity=0..15|WEIght=0..100}]
   [QLIMITExceedaction={NONE|LOg|TRap|BOth}]
   [QUEUEDrop={Head|Tail}] [RED={0..9999|NONE}]
   [REMarking=0..63|USEDscpmap|NONE}]
   [REMARKVlanpri={0..7|NONE}]
   [VIRTbw={bandwidth[Kbps|Mbps|Gbps]|NONE}]
   [WEIGhtscheduler={WRr|DWrr}]
```

Description

This command creates up to 1024 traffic classes. A traffic class specifies the QoS actions for a set of flows.

After you create the traffic class, use the **add sqos trafficclass subclass** command on page 39-91 to assign it to an intermediate traffic class, or the **add sqos policy trafficclass** command on page 39-88 to assign it to a policy.

Parameter	Description	
TRafficclass	ID number of the new traffic class. *id-list* is an integer from 0 to 9999, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once.	
DEscription	Description of the traffic class, which has no effect on its operation. A string 1 to 100 characters long. All printable characters are valid. If *description* contains spaces, it must be in double quotes. Default: no default	
BWClass3action	Action the router takes on Bandwidth Class 3 packets (red coloured packets). These are packets that exceed their allocated bandwidth as determined at the metering stage. Default: **none**	
	DRop	The router drops non-conformant packets.
	PAuse	The router drops non-conformant packets and stops dequeuing packets from the flow for **pausetime** seconds.
	NONE	The router sends non-conformant packets to the next processing stage.
MAxqlen	The maximum queue length, between 1 to 1023 packets, for the traffic class. The router drops packets that would exceed the maximum queue length. If you shape traffic by specifying a virtual bandwidth for a policy or traffic class (intermediate or leaf), give the appropriate leaf traffic classes large maximum queue lengths. This enables them to buffer bursts of packets and avoids packet loss. **maxqlen** is only valid on leaf traffic classes. Default: **64**	

Parameter	Description (cont.)	
MEter	Meter assigned to the traffic class, an integer from 0 to 9999. The meter determines a new bandwidth class (colour) for packets that are processed using this traffic class. You can configure the traffic class, or the policy it is attached to, to drop or queue the packets on the basis of the new bandwidth class. Default: **none**	
PAUSEAction	Notification action the router takes when it pauses a non-conformant traffic flow that belongs to this traffic class. Only valid if **bwclass3action**=**pause**. Default: **none**	
	LOg	The router generates a log message.
	TRap	The router generates an SNMP trap.
	BOth	The router generates both a log message and an SNMP trap.
	NONE	The router does not generate a notification.
PAUSETIme	Pause from 0 to 30 seconds when the router does not dequeue packets from a paused flow. If you specify 0, the router takes the action in **pauseaction**, but does not pause the flow. Only valid if **bwclass3action**=**pause**. Default: **10**	
QLIMITExceedaction	Notification action the router takes when a traffic flow exceeds the maximum queue length of the traffic class. Default: **none**	
	LOg	The router generates a log message.
	TRap	The router generates an SNMP trap.
	BOth	The router generates both a log message and an SNMP trap.
	NONE	The router does not generate a notification.
PREMARKBwcl	How the router assigns the packet to a bandwidth class at the start of the QoS processing (before metering). The router can use the assigned value in metering, marking and RED processing. You can specify only premarking in leaf traffic classes. Default: **1**	
	1..3	The router assigns the packet to the specified bandwidth class.
	USEDscpmap	The router uses the current DSCP value in conjunction with the policy's DSCP map to determine the bandwidth class. You must also specify the DSCP map by using the **dscpmap** parameter in the **create sqos policy** command on page 39-98 or the **set sqos policy** command on page 39-130.

Parameter	Description (cont.)	
PREMARKDscp	How the router changes the DSCP value in the packet header at the start of the QoS processing (before metering). The router can use the assigned value in metering, marking and RED processing. You can only specify premarking in leaf traffic classes. Default: **none**	
	0..63	The router writes the specified DSCP value into the packet header.
	USEDscpmap	The router uses the current DSCP value in conjunction with the policy's DSCP map to determine the new DSCP. You must also specify the DSCP map by using the **dscpmap** parameter in the **create sqos policy** command on page 39-98 or the **set sqos policy** command on page 39-130.
	NONE	The router does not change the packet DSCP value.
PRIOrity	An integer from 0 to 15 signifying the priority of the traffic class. Specifying **priority** in traffic classes sets their policy (or intermediate traffic class) to schedule queues according to the relative priorities of all its traffic classes. The router services the queue from the traffic class with the highest value for **priority** first. **Priority** and **weight** are mutually exclusive. Use the **priority** parameter to create a hierarchy based on the priority of flows, for strict priority queuing. Use the **weight** parameter to create a hierarchy with weighted flows, for WRR or DWRR queuing. If you create a mixed hierarchy the priority queues are emptied first, giving low latency queuing behaviour. Default: **1**	
QUEUEDrop	Whether packets are dropped from the head or tail of the queue when the queue becomes full. Tail dropping drops the newest packets; head dropping drops the oldest. Default: **tail**	
RED	RED curve set that the router uses for early dropping of packets. An integer from 0 to 9999. Default: **none**	
REMarking	How the router sets the bandwidth class and/or the DSCP value in the packet header's Differentiated Services field after metering. Default: **none**	
	0..63	The router writes the specified value into the DSCP bits in the packet header.
	USEDscpmap	The router uses the metered bandwidth class and current DSCP value, in conjunction with the policy's DSCP map, to determine the new DSCP value and/or bandwidth class.You must also specify the **dscpmap** parameter in the **create sqos policy** command on page 39-98 or the **set sqos policy** command on page 39-130.
	NONE	The router does not modify the DSCP value or metered bandwidth class.

Parameter	Description (cont.)	
REMARKVlanpri	Setting for the 802.1p VLAN priority field of the frame's Ethernet header. Default: **none**	
	0..7	The router writes the specified value into the 802.1p VLAN priority field of the Ethernet header.
	NONE	The router does not modify the 802.1p VLAN priority field of the Ethernet header.
VIRTbw	Maximum bandwidth available to the traffic class. **Virtbw** determines the maximum rate at which data can leave the internal queues to be transmitted onto the physical media. This rate is not equivalent to the transmission rate for data seen on the line, because the actual transmission rate includes the transmission of bits for the inter-frame-gap and the preamble of the Layer 2 headers. For example, 10 Mbps of data leaving the internal queues is not equivalent to 10 Mbps of data transmitted on the line. *rate* is from 1 to 16000000 kilobits per second, specified in Kbps, Mbps or Gbps (in upper or lower case). If you do not specify a unit, it uses kbps. If you specify Mbps or Gbps, *rate* may contain a decimal fraction with up to 3 decimal places, for example, 1.25 Mbps. Default: **none** (bandwidth is not limited)	
WEIght	Weight from 0 to 100 given to the traffic class. Specifying **weight** in traffic classes sets their policy (or intermediate traffic class) to schedule queues according to the relative weights of all its traffic classes. If a traffic class has a weight of 0 (zero), the router only empties its queue once the queues of all its sibling traffic classes are empty. **Priority** and **weight** are mutually exclusive. Use the **priority** parameter to create a hierarchy based on the priority of flows, for strict priority queuing. Use the **weight** parameter to create a hierarchy with weighted flows, for WRR or DWRR queuing. If you create a mixed hierarchy the priority queues are emptied first, giving low latency queuing behaviour. Default: no default, because the default behaviour is priority-based hierarchies.	
WEIGhtscheduler	Queue scheduling method that the router uses to schedule this traffic class' weighted sub traffic classes. This parameter is only valid if the sub traffic classes specify the **weight** parameter. Default: **wrr**	
	WRr	The router uses a weighted round robin scheme to empty the queues of weighted sub traffic classes.
	DWrr	The router uses a deficit weighted round robin scheme to empty the queues of weighted sub traffic classes. DWRR is less biased towards large packets than WRR.

Example To create a trafficclass that uses meter 1 to measure bandwidth, drops non-conformant packets, and uses the policy's DSCP map to re-mark conformant packets, use the command:

```
cre sqos tr=0 bwc=dr rem=used
```

To create a traffic class with a moderately-high priority, and use DWRR to schedule the queues of the traffic class' sub classes, use the command:

```
cre sqos tr=1 prio=10 weig=dw
```

Related Commands

add sqos policy trafficclass
add sqos trafficclass classifier
add sqos trafficclass dar
add sqos trafficclass subclass
delete sqos policy trafficclass
delete sqos trafficclass classifier
delete sqos trafficclass dar
delete sqos trafficclass subclass
destroy sqos trafficclass
set sqos trafficclass
show sqos trafficclass

delete sqos interface dar

Syntax `DELete SQOS INTerface=`*`interface`* `DAR={`*`id-list`*`|ALL}`

Description This command removes one or more DAR (Dynamic Application Recognition) objects from the interface. It does not destroy the DAR objects.

Parameter	Description
INTerface	Interface or tunnel fro which to remove the DAR object. Valid entry types are: Layer 1 and 2 interfaces: ● eth (such as. eth0) ● ATM channel (such as atm0.0) ● frame relay (such as fr0) ● PPP (such as ppp0) ● the switch instance (such as swi0) Layer 3 tunnels: ● GRE (such as gre1) ● IPv6 6-to-4 virtual interface (such as virt9) ● the name of an IPSec policy (such as ipsec-*policyname*) To see a list of current valid Layer 1 and 2 interfaces, use the **show interface** command on page 9-72 of Chapter 9, Interfaces.
DAR	DAR object to remove from the interface. *id-list* is an integer from 0 to 9999, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once. Default: no default

Example To stop using DAR object 0 to identify voice or video sessions that are setup on ppp0, use the command:

```
del sqos int=ppp0 dar=0
```

To stop using any DAR objects to identify voice or video sessions that are setup on ppp0, use the command:

```
del sqos int=ppp0 dar=all
```

To stop using any DAR objects to identify voice or video sessions that are setup over the IPsec policy *central*, use the command:

```
del sqos int=ipsec-central dar=all
```

Related Commands
add sqos interface dar
create sqos dar
delete sqos trafficclass dar
destroy sqos dar
set sqos dar
show sqos dar

delete sqos policy trafficclass

Syntax `DELete SQOS POLIcy=0..9999 TRafficclass={`*`id-list`*`|ALL}`

Description This command removes one or more traffic classes from the specified QoS policy. It does not destroy the traffic classes, or detach sub traffic classes from the traffic classes.

Parameter	Description
POLIcy	Policy from which to remove the traffic class. An integer from 0 to 9999.
TRafficclass	Traffic class to remove from the policy. *id-list* is an integer from 0 to 9999, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once. Default: no default

Example To remove leaf traffic class 1 and intermediate traffic class 2 from policy 1, use the command:

```
del sqos poli=1 tr=1,2
```

To remove all traffic classes from policy 1, use the command:

```
del sqos poli=1 tr=all
```

Related Commands

add sqos policy trafficclass
create sqos policy
create sqos trafficclass
destroy sqos policy
destroy sqos trafficclass
set sqos policy
set sqos trafficclass
show sqos policy
show sqos trafficclass

delete sqos trafficclass classifier

Syntax `DELete SQOS TRafficclass=0..9999 CLASSifier={`*`id-list`*`|ALL}`

Description This command removes one or more classifiers from the specified leaf traffic class. It does not destroy the classifiers.

Parameter	Description
TRafficclass	Leaf traffic class from which to remove the classifier. An integer from 0 to 9999.
CLASSifier	Classifier to remove from the traffic class. *id-list* is an integer from 0 to 9999, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once. Default: no default

Example To remove classifiers 1, 2, 4, 5, and 6 from leaf traffic class 3, use the command:

```
del sqos tr=3 class=1,2,4-6
```

To remove all classifiers from leaf traffic class 3, use the command:

```
del sqos tr=3 class=all
```

Related Commands

add sqos trafficclass classifier
delete sqos policy trafficclass
delete sqos trafficclass subclass
destroy sqos trafficclass
set sqos trafficclass
show sqos trafficclass

delete sqos trafficclass dar

Syntax

```
DELete SQOS TRafficclass=0..9999 DAR={id-list|ALL}
```

Description

This command disassociates one or more DAR (Dynamic Application Recognition) objects from the specified traffic class. It does not destroy the DAR object.

Parameter	Description
TRafficclass	Leaf traffic class from which to remove the DAR object. An integer from 0 to 9999.
DAR	DAR object to remove from the traffic class. *id-list* is an integer from 0 to 9999, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once. Default: no default

Example

To remove DAR objects 1, 2, 4, 5, and 6 from leaf traffic class 3, use the command:

```
del sqos tr=3 dar=1,2,4-6
```

To remove all DAR objects from leaf traffic class 3, use the command:

```
del sqos tr=3 dar=all
```

Related Commands

add sqos interface dar
add sqos trafficclass dar
create sqos dar
create sqos trafficclass
delete sqos interface dar
delete sqos trafficclass classifier
delete sqos trafficclass subclass
destroy sqos dar
destroy sqos trafficclass
set sqos dar
set sqos trafficclass
show sqos dar
show sqos trafficclass

delete sqos trafficclass subclass

Syntax `DELete SQOS TRafficclass=0..9999 SUBClass={`*`id-list`*`|ALL}`

Description This command removes one or more sub traffic classes from the specified intermediate traffic class. It does not destroy the traffic classes.

Parameter	Description
TRafficclass	Intermediate traffic class from which to remove a sub traffic class. An integer from 0 to 9999.
SUBClass	Sub traffic class to remove from the intermediate traffic class. *id-list* is an integer from 0 to 9999, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once. Default: no default

Example To remove leaf traffic classes 2 and 3 from intermediate traffic class 1, use the command:

```
del sqos tr=1 subc=2,3
```

To remove all sub traffic classes from intermediate traffic class 1, use the command:

```
del sqos tr=1 subc=all
```

Related Commands

add sqos trafficclass subclass
create sqos trafficclass
delete sqos trafficclass classifier
destroy sqos trafficclass
set sqos trafficclass
show sqos trafficclass

destroy sqos dar

Syntax `DESTroy SQOS DAR=`*`id-list`*

Description This command destroys one or more DAR objects. You cannot destroy a DAR object if an interface or traffic class uses it.

The *id-list* is an integer from 0 to 9999, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once.

Example To destroy DAR objects 1, 2 and 3, use the command:

```
dest sqos dar=1-3
```

Related Commands
add sqos interface dar
add sqos trafficclass dar
create sqos dar
delete sqos interface dar
delete sqos trafficclass dar
set sqos dar
show sqos dar

destroy sqos dscpmap

Syntax `DESTroy SQOS DSCPMap=`*`id-list`*

Description This command destroys one or more DSCP maps. You cannot destroy a DSCP map if a policy uses it.

The *id-list* is an integer from 1 to 9999, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 1,3,4-9). An integer cannot appear in the list more than once.

Example To destroy DSCP maps 1, 2 and 3, use the command:

```
dest sqos dscpm=1-3
```

Related Commands
create sqos dscpmap
create sqos trafficclass
set sqos dscpmap
show sqos dscpmap

destroy sqos meter

Syntax `DESTroy SQOS METer=`*`id-list`*

Description This command destroys one or more meters. You cannot destroy a meter if a policy or traffic class uses it.

The *id-list* is an integer from 0 to 9999, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once.

Example To destroy meters 1, 2 and 3, use the command:

```
dest sqos met=1-3
```

Related Commands
create sqos trafficclass
create sqos meter
set sqos meter
show sqos meter

destroy sqos policy

Syntax `DESTroy SQOS POLIcy=`*`id-list`*

Description This command destroys one or more policies. You cannot destroy a policy if it is attached to an interface or if any traffic classes are attached to it.

The *id-list* is an integer from 0 to 9999, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once.

Example To destroy policies 1, 2 and 3, use the command:

```
dest sqos poli=1-3
```

Related Commands
add sqos policy trafficclass
create sqos policy
delete sqos policy trafficclass
set sqos policy
show sqos policy

destroy sqos red

Syntax `DESTroy SQOS RED=`*`id-list`*

Description This command destroys one or more RED curve sets. You cannot destroy a RED curve set if a traffic class uses it. You cannot destroy the default RED curve sets (0-2).

The *id-list* is an integer from 3 to 9999, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 3,4-9). An integer cannot appear in the list more than once.

Example To destroy RED curve set 3, use the command:

```
dest sqos red=3
```

Related Commands
create sqos red
create sqos trafficclass
set sqos red
show sqos red

destroy sqos trafficclass

Syntax `DESTroy SQOS TRafficclass=`*`id-list`*

Description This command destroys one or more traffic classes. You cannot destroy a traffic class if it is attached to a policy, or if any sub traffic classes or classifiers are attached to it.

The *id-list* is an integer from 0 to 9999, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once.

Example To destroy traffic classes 1, 2 and 3, use the command:

```
dest sqos tr=1-3
```

Related Commands
delete sqos policy trafficclass
delete sqos trafficclass classifier
delete sqos trafficclass dar
delete sqos trafficclass subclass
create sqos trafficclass
set sqos trafficclass
show sqos trafficclass

disable sqos

Syntax `DISable SQOS`

Description This command disables software QoS. Software QoS is disabled by default.

Example To disable software QoS, use the command:

```
dis sqos
```

Related Commands
disable sqos debug
enable sqos
enable sqos debug
purge sqos
show sqos
show sqos counters

disable sqos debug

Syntax
```
DISable SQOS
   DEBug={ALL|DAR|DARDATA|ENGine|ERRor|INFo|MARk|PKT}
```

Description This command disables software QoS debugging. Debugging is disabled by default.

Parameter	Description	
DEBug	The debug mode to disable. Default: no default	
	ALL	All debugging modes.
	DAR	Notifications when DAR objects and instances are created or destroyed.
	DARDATA	More detailed information about voice and video data.
	ENGine	Debugging information related to the packet conditioning engine.
	ERRor	Critical error debugging information, including a stack trace.
	INFo	General debugging information.
	MARk	Packet marking debugging information.
	PKT	Packet debugging.

Example To disable packet debugging, use the command:

```
dis sqos deb=pkt
```

Related Commands
disable sqos
enable sqos debug
enable sqos
purge sqos
show sqos
show sqos counters

enable sqos

Syntax

```
ENAble SQOS
```

Description

This command enables software QoS. Software QoS is disabled by default.

Example

To enable software QoS, use the command:

```
ena sqos
```

Related Commands

disable sqos
disable sqos debug
enable sqos debug
purge sqos
show sqos
show sqos counters

enable sqos debug

Syntax

```
ENAble SQOS
    DEBug={ALL|DAR|DARDATA|ENGine|ERRor|INFo|MARk|PKT}
```

Description

This command enables software QoS debugging. Debugging is disabled by default.

Parameter	Description	
DEBug	The debug mode to enable. Default: no default	
	ALL	All debugging modes.
	DAR	Notifications when DAR objects and instances are created or destroyed.
	DARDATA	More detailed information about SIP and RTSP data.
	ENGine	Debugging information related to the packet conditioning engine.
	ERRor	Critical error debugging information, including a stack trace.
	INFo	General debugging information.
	MARk	Packet marking debugging information.
	PKT	Packet debugging.

Example

To enable packet debugging, use the command:

```
ena sqos deb=pkt
```

Related Commands

disable sqos
disable sqos debug
enable sqos
purge sqos
show sqos
show sqos counters

purge sqos

Syntax `PURge SQOS {DAR|INTerface|POLIcy|TRafficclass}`

Description This command destroys all or a section of software QoS configuration. If you specify **purge sqos** with no other parameters, all software QoS configuration is destroyed.

Parameter	Description
DAR	Destroy all DAR objects.
INTerface	Remove software QoS policies from all interfaces. The interfaces and policies are not destroyed.
POLIcy	Destroy all software QoS policies. This parameter destroys the traffic class hierarchy by also detaching all sub traffic classes from intermediate traffic classes.
TRafficclass	Destroy all traffic classes.

Example To destroy all software QoS configuration, use the command:

```
pur sqos
```

To destroy all software QoS policies and the traffic class hierarchy, while leaving the traffic classes intact, use the command:

```
pur sqos poli
```

Related Commands
disable sqos
disable sqos debug
enable sqos
enable sqos debug
show sqos
show sqos counters

reset sqos counters

Syntax

```
RESET SQOS COUnters POLIcy[=id-list]
   [DIrection={In|OUT|TUNnel}] [INTerface=interface]

RESET SQOS COUnters TRafficclass[=id-list]
   [DIrection={In|OUT|TUNnel|ALL}] [INTerface=interface]

RESET SQOS COUnters CLASSifier[=id-list]
   [DIrection={In|OUT|TUNnel|ALL}] [INTerface=interface]
   [TRafficclass=id-list]

RESET SQOS COUnters DAR[=id-list]
   [DIrection={In|OUT|TUNnel}] [INTerface=interface]
```

Description This command resets counter information for the specified software QoS objects.

Parameter	Description	
CLASSifier	Classifier for which to reset the counters. An integer from 0 to 9999, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once. Default: no default (classifier counters are not reset)	
DAR	DAR object for which to reset the counters. An integer from 0 to 9999, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once. Default: no default (DAR object counters are not reset)	
DIrection	Filter that restricts the command, so that the router only resets counters for software QoS objects with this direction. Default: no default (not filtered by direction)	
	IN	Reset counters for software QoS objects that act on the packet at ingress.
	OUT	Reset counters for software QoS objects that act on the packet at egress.
	TUNnel	Reset counters for software QoS objects that act on tunnelled packets.
	ALL	Reset counters for software QoS objects that act on packets in any direction. **All** is only valid with **trafficclass** and **classifier**.

Parameter	Description (cont.)
INTerface	A filter that restricts the command, so that the router only resets counters for software QoS objects that are associated with this interface or tunnel. Valid entry types are: Layer 1 and 2 interfaces: ● eth (such as. eth0) ● ATM channel (such as atm0.0) ● frame relay (such as fr0) ● PPP (such as ppp0) ● the switch instance (such as swi0) Layer 3 tunnels: ● GRE (such as gre1) ● IPv6 6-to-4 virtual interface (such as virt9) ● the name of an IPSec policy (such as ipsec-*policyname*) To see a list of current valid Layer 1 and 2 interfaces, use the **show interface** command on page 9-72 of Chapter 9, Interfaces. Default: no default (not filtered by interface)
POLIcy	Policy for which to reset the counters. An integer from 0 to 9999, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once. Default: no default (policy counters are not reset)
TRafficclass	Traffic class. An integer from 0 to 9999, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once. If you specify both **classifier** and **trafficclass** the router resets the classifier counters for that traffic class; otherwise it resets the traffic class counters. Default: no default

Example To reset all classifier counters for traffic class 1, use the command:

```
reset sqos cou class tr=1
```

To reset ingress software QoS policy counters on ppp0, use the command:

```
reset sqos cou poli di=in int=ppp0
```

Related Commands

disable sqos debug
enable sqos debug
show sqos
show sqos counters

set sqos dar

Syntax

```
SET SQOS DAR=id-list [CODEC={AUDio|VIDeo|ANY}]
   [DESCription=description] [DSTIp={ipadd[/
   0..32]|ipv6add[/0..128]}] [SRCIp={ipadd[/
   0..32]|ipv6add[/0..128]}]
   [INACtivetimeout={1..3600|NONE}]
   [PROTocol={SIP|RTSp|H323|ALL}]
   [H323Port=1..65535] [RTSPPort=1..65535]
   [SIPPort=1..65535]
```

Description This command modifies one or more Dynamic Application Recognition objects.

Parameter	Description	
DAR	ID number of the DAR object. *id-list* is an integer from 0 to 9999, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once.	
CODEC	Coder/decoder for the DAR object to use to match packets. Default: **any**	
	AUDio	Matches traffic flows that use any audio codec.
	VIDeo	Matches traffic flows that use any video codec.
	ANY	The DAR object ignores the codec. A DAR object with **codec=any** matches any sessions set up by the specified protocol, not just voice and video sessions.
DESCription	Description of the DAR object, which has no effect on its operation. A string 1 to 100 characters long. All printable characters are valid. If *description* contains spaces, it must be in double quotes. Default: no default	
DSTIp	Destination IPv4 or IPv6 address or subnet. The DAR object only matches voice or video traffic flows to that address or network. IPv4 addresses are specified in dotted decimal notation. IPv6 addresses are specified as eight pairs of octets, separated by colons. The CIDR mask for IPv4 and prefix length for IPv6 are optional. For IPv4, if you specify a subnet address without specifying a mask, the default mask for that subnet is used. For IPv6, the default prefix length is 128. If you also specify **srcip**, either both must be IPv4 addresses or both must be IPv6 addresses. Default: no default (ignores destination IP address)	
H323Port	TCP port that H.323 session control messages are received on. Default: **1720**	
INACtivetimeout	Time from 1 to 3600 seconds (up to 60 minutes). If a classified flow is idle for this length of time, its entry is deleted. Default: **600**	
PROTocol	Protocol for the DAR object to use to match packets. Default: **all** (ignores protocol)	

Parameter	Description
RTSPPort	TCP port that RTSP session control messages are received on. Default: **554**
SIPPort	UDP port that SIP messages are received on. Default: **5060**
SRCIp	Source IPv4 or IPv6 address or subnet. The DAR object only matches voice or video traffic flows from that address or network. IPv4 addresses are specified in dotted decimal notation. IPv6 addresses are specified as eight pairs of octets, separated by colons. The CIDR mask for IPv4 and prefix length for IPv6 are optional. For IPv4, if you specify a subnet address without specifying a mask, the default mask for that subnet is used. For IPv6, the default prefix length is 128. If you also specify **dstip**, either both must be IPv4 addresses or both must be IPv6 addresses. Default: no default (ignores source IP address)

Example To modify DAR object 0 so that it identifies and classifies voice packets destined for the 192.168.1.0 subnet, use the command:

```
set sqos dar=0 codec=audio dsti=192.168.1.0/24
```

Related Commands

add sqos interface dar
add sqos trafficclass dar
create sqos dar
delete sqos interface dar
delete sqos trafficclass dar
destroy sqos dar
show sqos dar

set sqos dscpmap

Syntax To set the premarking table:

```
SET SQOS DSCPMap=id-list TAble=PREmark
   [DESCription=description] [DSCP=dscp-list]
   [NEWBwclass=1..3] [NEWDscp=0..63]
```

To set the re-marking table:

```
SET SQOS DSCPMap=id-list TAble=REMark
   [DESCription=description] [BWClass=bwclass-list]
   [DSCP=dscp-list] [NEWBwclass=1..3] [NEWDscp=0..63]
```

To change only the description:

```
SET SQOS DSCPMap=id-list [DESCription=description]
```

Description This command configures one or more DSCP maps. Each map consists of a premarking table and a re-marking table. To modify both, use separate commands.

Parameter	Description
DSCPMap	ID number of the DSCP map. An integer from 0 to 9999, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 1,4-9). An integer cannot appear in the list more than once.
BWClass	Bandwidth class to use as an index into the re-marking table. An integer from 1 to 3, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 1,3). The router writes the **newdscp** and/or **newbwclass** into the entries in the table that have this bandwidth class and the specified **dscp** (if any). If you specify neither **dscp** or **bwclass**, the router gives the **newdscp** and/or **newbwclass** to all re-marking table entries**.** Default: no default
DESCription	Description of the DSCP map, which has no effect on its operation. A string from 1 to 100 characters long. All printable characters are valid. If *description* contains spaces, it must be in double quotes. Default: no default
DSCP	DSCP value to use as an index into the premarking or re-marking table. An integer from 0 to 63, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 1,4-9). The router writes the **newdscp** and/or **newbwclass** into the entries in the table that have this DSCP and for the re-marking table the specified **bwclass** (if any). If you specify neither **dscp** or, for the re-marking table, **bwclass**, the router gives the **newdscp** and/or **newbwclass** to all entries**.**
NEWBwclass	New bandwidth class for entries with the specified **dscp** and/or, for the re-marking table, **bwclass**. An integer from 1 to 3. Default: no default
NEWDscp	New DSCP value for entries with the specified **dscp** and/or, for the re-marking table, **bwclass**. An integer from 0 to 63. Default: no default

Parameter	Description (cont.)	
TAble	Table to configure in the DSCP map. Default: no default	
	PREmark	Used before the metering stage.
	REMark	Used after metering has taken place, after the packet has been dequeued from the leaf traffic class.

Example To set DSCP map 1 so that the re-mark table gives non-conformant traffic a DSCP of 10, use the command:

```
set dscpm=1 table=rem bwc=3 newd=10
```

Related Commands

create sqos dscpmap
create sqos trafficclass
destroy sqos dscpmap
show sqos dscpmap

set sqos interface

Syntax

```
SET SQOS INTerface=interface [INpolicy={0..9999|NONE}]
   [OUtpolicy={0..9999|NONE}]
   [TUNnelpolicy={0..9999|NONE}]
```

Description This command associates one or more software QoS policies with a Layer 1 or 2 interface or Layer 3 tunnel.

Parameter	Description
INTerface	Interface or tunnel with which to associate the policy. Valid entry types are: Layer 1 and 2 interfaces: ● eth (such as. eth0) ● ATM channel (such as atm0.0) ● frame relay (such as fr0) ● PPP (such as ppp0) ● the switch instance (such as swi0) Layer 3 tunnels: ● GRE (such as gre1) ● IPv6 6-to-4 virtual interface (such as virt9) ● the name of an IPSec policy (such as ipsec-*policyname*) To see a list of current valid Layer 1 and 2 interfaces, use the **show interface** command on page 9-72 of Chapter 9, Interfaces.
INpolicy	Software QoS policy that the router applies to ingress traffic on the interface. **Inpolicy** is valid only for interfaces at Layers 1 and 2. Default: no default 0..9999 The policy ID. NONE No policy. You can use this option to deactivate software QoS on ingress traffic for the interface.
OUtpolicy	Software QoS policy that the router applies to egress traffic on the interface. **Outpolicy** is valid only for interfaces at Layers 1 and 2. Default: no default 0..9999 The policy ID. NONE No policy. You can use this option to deactivate software QoS on egress traffic for the interface.
TUNnelpolicy	Software QoS policy that the router applies to tunnelled traffic on the interface. **Tunnelpolicy** is only valid for tunnels. The router performs QoS processing on traffic before it enters the tunnel (in other words, before it is encapsulated by the tunnelling protocol). Default: no default 0..9999 The policy ID. NONE No policy. You can use this option to deactivate software QoS on tunnelled traffic for the interface.

Example To apply policy 1 to egress traffic on ppp0 use the command:

```
set sqos int=ppp0 ou=1
```

To apply software QoS policy 2 to traffic that the IPsec policy *central* processes, use the command:

```
set sqos int=ipsec-central tun=2
```

Related Commands

create sqos policy
delete sqos policy trafficclass
destroy sqos policy
set sqos policy
show sqos interface
show sqos policy

set sqos meter

Syntax

```
SET SQOS METer=id-list [DESCription=description]
   [IGNorebwclass={Yes|No}]
   [MINbandwidth=rate[Kbps|Mbps|Gbps]]
   [MAXbandwidth=rate[Kbps|Mbps|Gbps]]
   [MINBUrstsize=burstsize[Bytes|Kbytes|Mbytes|Gbytes]]
   [MAXBUrstsize=burstsize[Bytes|Kbytes|Mbytes|Gbytes]]
   [TYPE={SRtcm|TRtcm}]
```

Description

This command modifies one or more Three Colour Marker meters, as described in RFC 2697, *A Single Rate Three Color Marker* and RFC 2698, *A Two Rate Three Color Marker*. The meter measures how much bandwidth packets in a traffic flow use, and how well bandwidth usage conforms with specifications for the traffic class to which the flow belongs. It assigns the packet to a bandwidth class depending on its conformance. For more information, see Bandwidth Class on page 20.

Parameter	Description	
METer	ID number of the meter. *id-list* is an integer from 0 to 9999, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once.	
DESCription	Description of the meter, which has no effect on its operation. A string 1 to 100 characters long. All printable characters are valid. If *description* contains spaces, it must be in double quotes. Default: no default	
IGNorebwclass	Whether the meter acknowledges any previous bandwidth class assigned to packets. A meter that acknowledges previous conformance is called *colour aware*. Default: **no** (the meter is colour aware)	
	Yes	The metering function is colour blind and ignores any bandwidth class previously assigned to packets. It sets the meter bandwidth class according to only the metered conformance level of the flow.
	No	The metering function is colour aware and uses any bandwidth class previously assigned to packets, as well as the metered conformance level, to set the bandwidth class. Packets previously labelled non-conformant (bandwidth class 3, red) remain in bandwidth class 3. Packets previously labelled partially-conformant (bandwidth class 2, yellow) are assigned to bandwidth class 2 or 3, depending on metered conformance.

Parameter	Description (cont.)
MAXbandwidth	For the single rate meter of RFC 2697, the highest rate at which a steady stream of packets can arrive at the meter and be assigned to bandwidth class 1 (conformant, green). This is the Committed Information Rate (CIR) of the RFC. For the two rate meter of RFC 2698, the Peak Information Rate (PIR) of the RFC. See "Metering: Bandwidth Conformance" on page 39-21 for a description of PIR. It must equal or exceed **minbandwidth**. *rate* is from 0 to 16000000 kilobits per second, specified in Kbps, Mbps or Gbps (in upper or lower case). If you do not specify a unit, it uses kbps. If you specify Mbps or Gbps, *rate* may contain a decimal fraction with up to 3 decimal places, for example, 1.25 Mbps. Default: **1Mbps**
MAXBUrstsize	For the single rate meter of RFC 2697, the amount by which a packet can exceed **maxbandwidth** plus **minburstsize** and still possibly be assigned to bandwidth class 2. This is the Excess Burst Size (EBS) of the RFC. For the two rate meter of RFC 2698, the amount by which a packet can exceed **maxbandwidth** and still possibly be assigned to bandwidth class 2. This is the Peak Burst Size (PBS) of the RFC. *burstsize* is from 0 to 16777216 bytes (16 MB), specified in bytes, kbytes, Mbytes, or Gbytes (in upper or lower case). If you do not specify a unit, it uses bytes. If you specify kB, MB or GB, *burstsize* may contain a decimal fraction, for example, 1.25M. For a single rate meter, at least one of **minburstsize** and **maxburstsize** should equal or exceed the size of the largest IP packet you expect on the metered flow. For a two rate meter, both **minburstsize** and **maxburstsize** must exceed 0 (zero), and the RFC recommends that both values equal or exceed the size of the largest IP packet you expect on the metered flow. To create a single rate 2-colour meter (green and red), set this to 0 (zero). Default: **10kbytes**
MINbandwidth	For the two rate meter of RFC 2698, the Committed Information Rate (CIR) of the RFC. See "Metering: Bandwidth Conformance" on page 39-21 for a description of CIR. It must not exceed **maxbandwidth**. *rate* is from 0 to 16000000 kilobits per second, specified in Kbps, Mbps or Gbps (in upper or lower case). If you do not specify a unit, it uses kbps. If you specify Mbps or Gbps, *rate* may contain a decimal fraction with up to 3 decimal places, for example, 1.25 Mbps. Only valid if **type=trtcm.** Default: **1Mbps**

Parameter	Description (cont.)
MINBUrstsize	For the single rate meter of RFC 2697, the amount by which a packet can exceed **maxbandwidth** and still possibly be assigned to bandwidth class 1. This is the Committed Burst Size (CBS) of the RFC. For the two rate meter of RFC 2698, the amount by which a packet can exceed **minbandwidth** and still possibly be assigned to bandwidth class 1. This is the Committed Burst Size (CBS) of the RFC. *burstsize* is from 0 to 16777216 bytes (16 MB), specified in bytes, kbytes, Mbytes, or Gbytes (in upper or lower case). If you do not specify a unit, it uses bytes. If you specify kB, MB or GB, *burstsize* may contain a decimal fraction, for example, 1.25M. For a single rate meter, at least one of **minburstsize** and **maxburstsize** should equal or exceed the size of the largest IP packet you expect on the metered flow. For a two rate meter, both **minburstsize** and **maxburstsize** must exceed 0 (zero), and the RFC recommends that both values equal or exceed the size of the largest IP packet you expect on the metered flow. For a single rate meter, if you set **minburstsize** to 0 (zero) the meter assigns all packets to bandwidth class 2 or 3. Default: **10kbytes**
TYPE	The type of meter. "Metering: Bandwidth Conformance" on page 39-21 describes the meters. Default: **srtcm** SRtcm The Single Rate Three Colour Marker of RFC 2697. TRtcm The Two Rate Three Colour Marker of RFC 2698.

Example To make meter 0 a colour-blind meter, use the command:

```
set sqos met=0 ign=y
```

Related Commands

create sqos meter
create sqos trafficclass
destroy sqos meter
show sqos meter

set sqos policy

Syntax

```
SET SQOS POLIcy=id-list
   [BWClass3action={DROP|PAUSE|NONE}]
   [DEFaulttrafficclass={0..9999|NONE}]
   [DESCription=description] [DSCPMap={0..9999|NONE}]
   [IGNOREPrenatinfo={YES|NO}] [METer={0..9999|NONE}]
   [PAUSEAction={NONE|LOg|TRap|BOth}] [PAUSETIme={1..30}]
   [REMarking={0..63|USEDscpmap|NONE}]
   [REMARKVlanpri={0..7|NONE}] [SYSTEMTraffic={5..50}]
   [VIRTbw={rate[Kbps|Mbps|Gbps]|NONE}]
   [WEIGhtscheduler={WRr|DWrr}]
```

Description This command modifies one or more QoS policies. A policy defines the overall QoS processing for an interface.

Parameter	Description	
POLIcy	ID number of the policy. *id-list* is an integer from 0 to 9999, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once.	
BWClass3action	Action the router takes on Bandwidth Class 3 packets (red coloured packets). These are packets that exceed their allocated bandwidth, as determined at the metering stage. Default: **none**	
	DROP	The router drops non-conformant packets.
	PAUSE	The router drops non-conformant packets and stops dequeuing packets from the flow for **pausetime** seconds.
	NONE	The router sends non-conformant packets to the next processing stage.
DEFaulttrafficclass	Traffic class that the router applies to unclassified traffic on the policy's interface. It must be a leaf traffic class. Default: no default	
	0..9999	The traffic class ID.
	NONE	No user-nominated default traffic class. The router uses the default traffic class that it made when you created the policy.
DEscription	Description of the policy, which has no effect on its operation. A string 1 to 100 characters long. All printable characters are valid. If *description* contains spaces, it must be in double quotes. Default: no default	
DScpmap	DSCP map to assign to the policy. An integer from 0 to 9999. Default: **none**	
IGNOREPrenatinfo	Whether classifiers attached to the policy use pre-NAT IP settings for classification because these contain the distinguishing information. Default: **no** (uses pre-NAT settings)	

Parameter	**Description** (cont.)	
MEter	Meter to assign to the policy. An integer from 0 to 9999. The meter determines a new bandwidth class (colour) for packets that are processed using this policy. You can configure the policy or a traffic class to drop or queue the packets on the basis of the new bandwidth class. Default: **none**	
PAUSEAction	Notification action the router takes when it pauses a non-conformant traffic flow that belongs to this policy. Only valid if **bwclass3action** is **pause**. Default: **none**	
	LOg	The router generates a log message.
	TRap	The router generates an SNMP trap.
	BOth	The router generates both a log message and an SNMP trap.
	NONE	The router does not generate a notification.
PAUSETIme	Pause from 1 to 30 seconds when the router does not dequeue packets from a paused flow. Only valid if **bwclass3action** is **pause**. Default: **10**	
REMarking	How the router sets the bandwidth class and/or DSCP value in the packet header's Differentiated Services field after metering. Default: **none**	
	0..63	The router writes the specified value into the DSCP bits in the packet header.
	USEDscpmap	The router uses the metered bandwidth class and current DSCP value, in conjunction with the policy's DSCP map, to determine the new DSCP value and/or bandwidth class. You must also specify the **dscpmap** parameter.
	NONE	The router does not modify the DSCP value or metered bandwidth class.
REMARKVlanpri	Setting for the 802.1p VLAN priority field of the frame's Ethernet header. Default: **none**	
	0..7	The router writes the specified value into the 802.1p VLAN priority field of the Ethernet header.
	NONE	The router does not modify the 802.1p VLAN priority field of the Ethernet header.
SYSTEMTraffic	Percentage of the interface's maximum bandwidth from 5% to 50% that the router reserves for system traffic. Default: **20**	

Parameter	Description (cont.)	
VIRTbw	Maximum bandwidth available to the policy. **Virtbw** determines the maximum rate at which data can leave the internal queues to be transmitted onto the physical media. This rate is not equivalent to the transmission rate for data seen on the line, because the actual transmission rate includes the transmission of bits for the inter-frame-gap and the preamble of the Layer 2 headers. For example, 10 Mbps of data leaving the internal queues is not equivalent to 10 Mbps of data transmitted on the line. *rate* is from 0 to 16000000 kilobits per second, specified in Kbps, Mbps or Gbps (in upper or lower case). If you do not specify a unit, it uses kbps. If you specify Mbps or Gbps, *rate* may contain a decimal fraction with up to 3 decimal places, for example, 1.25 Mbps. Default: **none** (bandwidth is not limited)	
WEIGhtscheduler	Queue scheduling method for weighted traffic classes that belong to the policy. Weighted traffic classes assign weights to flows instead of priorities. Default: **wrr**	
	WRr	The router uses a weighted round robin scheme to empty the queues of weighted traffic classes.
	DWrr	The router uses a deficit weighted round robin scheme to empty the queues of weighted traffic classes. DWRR is less biased towards large packets than WRR.

Example To modify policy 0 so that it allocates 15% of the available bandwidth to system traffic, use the command:

```
set sqos poli=0 systemt=15
```

Related Commands

add sqos policy trafficclass
create sqos policy
delete sqos policy trafficclass
destroy sqos policy
set sqos interface
show sqos policy

set sqos red

Syntax

```
SET SQOS RED=id-list
   [AVERaging=0..99] [DESCription=description]
   [START1=0..100] [STOP1=0..100] [DROP1=0..100]
   [START2=0..100] [STOP2=0..100] [DROP2=0..100]
   [START3=0..100] [STOP3=0..100] [DROP3=0..100]
```

Description

This command modifies one or more sets of RED curves. Red curve sets 0-2 exist by default, and cannot be modified or deleted. Table 39-3 on page 39-31 shows the properties of the default red curve sets.

Parameter	Description
RED	ID number of the RED curve set. *id-list* is an integer from 3 to 9999, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 3,4-9). An integer cannot appear in the list more than once.
AVERaging	Weight used in the moving averaging estimation of queue length for the RED curve algorithm. The estimated queue length is frequently updated, and is calculated by taking a weighted average of the previous average and the current instantaneous queue length. **Averaging** is the weight given to the previous average in this weighted calculation. If **averaging** is too high, the estimated average queue size responds too slowly to transient congestion. If **averaging** is too low, the estimated average queue size tracks the instantaneous queue size too closely and you lose the benefits of RED. RED works best when the estimated average queue length responds as slowly as possible while preventing the queue from becoming full. To achieve this, set **averaging** to a lower value if the queue constantly becomes full, so that the estimated average queue size more closely tracks the actual queue size. To check how often the queue becomes full, use the **trafficclass** parameter of the **show sqos counters** command on page 39-140 and check the queue counters, or set **qlimitexceedaction** and check the log messages or SNMP traps. Default: **98**
DESCription	Description of the RED curve set, which has no effect on its operation. A string from 1 to 100 characters long. All printable characters are valid. If *description* contains spaces, it must be in double quotes. Default: no default
START1 START2 START3	Percentage of the queue length from 1% to 100% when the RED algorithm starts to drop packets, for packets in bandwidth classes 1, 2, and 3 respectively. Default: **35**
STOP1 STOP2 STOP3	Percentage of the queue length from 1% to 100% when the RED algorithm is dropping **drop** percent of the packets, for packets in bandwidth classes1, 2, and 3 respectively. Beyond this point, 100% of the packets are dropped. This value must be greater than **start**. Default: **65**
DROP1 DROP2 DROP3	Probability from 1% to 100% that a packet will be dropped at the **stop** queue length for packets in bandwidth classes 1, 2 and 3 respectively. Default: **30**

Example To set the drop probability to 20% for bandwidth class 1 packets in RED curve set 3, use the command:

```
set red=3 drop1=20
```

Related Commands

create sqos red
create sqos trafficclass
destroy sqos red
show sqos red

set sqos trafficclass

Syntax

```
SET SQOS TRafficclass=id-list
   [BWClass3action={DRop|PAuse|NONE}]
   [DESCription=description] [MAXQlen=1..1023]
   [METer={0..9999|NONE}]
   [PAUSEAction={NONE|LOg|TRap|BOth}] [PAUSETIme={1..30}]
   [PREMARKBwcl={1..3|USEDscpmap}]
   [PREMARKDscp={0..63|USEDscpmap|NONE}]
   [{PRIOrity=0..15|WEIght=0..100}]
   [QLIMITExceedaction={NONE|LOg|TRap|BOth}]
   [QUEUEDrop={Head|Tail}] [RED={0..9999|NONE}]
   [REMarking=0..63|USEDscpmap|NONE}]
   [REMARKVlanpri={0..7|NONE}]
   [VIRTbw={bandwidth[Kbps|Mbps|Gbps]|NONE}]
   [WEIGhtscheduler={WRr|DWrr}]
```

Description This command modifies one or more traffic classes. A traffic class specifies the QoS actions for a set of flows.

Parameter	Description	
TRafficclass	ID number of the traffic class. *id-list* is an integer from 0 to 9999, a range of integers separated by a hyphen, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once.	
DEscription	Description of the traffic class, which has no effect on its operation. A string 1 to 100 characters long. All printable characters are valid. If *description* contains spaces, it must be in double quotes. Default: no default	
BWClass3action	Action the router takes on Bandwidth Class 3 packets (red coloured packets). These are packets that exceed their allocated bandwidth, as determined at the metering stage. Default: **none**	
	DRop	The router drops non-conformant packets.
	PAuse	The router drops non-conformant packets and stops dequeuing packets from the flow for **pausetime** seconds.
	NONE	The router sends non-conformant packets to the next processing stage.
MAxqlen	Maximum combined queue length, from 1 to 1023 packets, for the traffic class. The router drops packets that would exceed the maximum queue length. If you shape traffic by specifying a virtual bandwidth for a policy or traffic class (intermediate or leaf), give the appropriate leaf traffic classes large maximum queue lengths. This enables them to buffer bursts of packets and avoids packet loss. **maxqlen** is only valid on leaf traffic classes. Default: **64**	
MEter	Meter to assign to the traffic class. An integer from 0 to 9999. The meter determines a new bandwidth class (colour) for packets that are processed using this traffic class. You can configure the traffic class, or the policy it is attached to, to drop or queue the packets on the basis of the new bandwidth class. Default: **none**	

Parameter	Description (cont.)	
PAUSEAction	Notification action the router takes when it pauses a non-conformant traffic flow that belongs to this traffic class. Only valid if **bwclass3action**=**pause**. Default: **none**	
	LOg	The router generates a log message.
	TRap	The router generates an SNMP trap.
	BOth	The router generates both a log message and an SNMP trap.
	NONE	The router does not generate a notification.
PAUSETIme	Pause from 0 to 30 seconds when the router does not dequeue packets from a paused flow. If you specify 0, the router takes the action in **pauseaction**, but not pause the flow. Only valid if **bwclass3action**=**pause**. Default: **10**	
QLIMITExceedaction	Notification action the router takes when a traffic flow exceeds the maximum queue length of the traffic class. Default: **none**	
	LOg	The router generates a log message.
	TRap	The router generates an SNMP trap.
	BOth	The router generates both a log message and an SNMP trap.
	NONE	The router does not generate a notification.
PREMARKBwcl	How the router assigns the packet to a bandwidth class at the start of the QoS processing (before metering). The router can use the assigned value in metering, marking and RED processing. You can specify premarking only in leaf traffic classes. Default: **1**	
	1..3	The router assigns the packet to the specified bandwidth class.
	USEDscpmap	The router uses the current DSCP value in conjunction with the policy's DSCP map to determine the bandwidth class. You must also specify the DSCP map by using the **dscpmap** parameter in the **create sqos policy** command on page 39-98 or the **set sqos policy** command on page 39-130.
PREMARKDscp	How the router changes the DSCP value in the packet header at the start of the QoS processing (before metering). The router can use the assigned value in metering, marking and RED processing. You can specify premarking only in leaf traffic classes. Default: **none**	
	0..63	The router writes the specified DSCP value into the packet header.
	USEDscpmap	The router uses the current DSCP value in conjunction with the policy's DSCP map to determine the new DSCP. You must also specify the DSCP map by using the **dscpmap** parameter in the **create sqos policy** command on page 39-98 or the **set sqos policy** command on page 39-130.
	NONE	The router does not change the packet DSCP value.

Parameter	Description (cont.)	
PRIOrity	Integer from 0 to 15 for the priority of the traffic class. Specifying **priority** in traffic classes sets their policy (or intermediate traffic class) to schedule queues according to the relative priorities of all its traffic classes. The router services the queue from the traffic class with the highest value for **priority** first. **Priority** and **weight** are mutually exclusive. Use the **priority** parameter to create a hierarchy based on the priority of flows, for strict priority queuing. Use the **weight** parameter to create a hierarchy with weighted flows, for WRR or DWRR queuing. If you create a mixed hierarchy the priority queues are emptied first, giving low latency queuing behaviour. Default: **1**	
QUEUEDrop	Whether packets are dropped from the head or tail of the queue when the queue becomes full. Tail dropping drops the newest packets; head dropping drops the oldest. Default: **tail**	
RED	Integer from 0 to 9999 for the RED curve set that the router uses for early dropping of packets. Default: **none**	
REMarking	How the router sets the bandwidth class and/or the DSCP value in the packet header's Differentiated Services field after metering. Default: **none**	
	0..63	The router writes the specified value into the DSCP bits in the packet header.
	USEDscpmap	The router uses the metered bandwidth class and current DSCP value, in conjunction with the policy's DSCP map, to determine the new DSCP value and/or bandwidth class.You must also specify the **dscpmap** parameter in the **create sqos policy** command on page 39-98 or the **set sqos policy** command on page 39-130.
	NONE	The router does not modify the DSCP value or metered bandwidth class.
REMARKVlanpri	Setting for the 802.1p VLAN priority field of the frame's Ethernet header. Default: **none**	
	0..7	The router writes the specified value into the 802.1p VLAN priority field of the Ethernet header.
	NONE	The router does not modify the 802.1p VLAN priority field of the Ethernet header.

Parameter	Description (cont.)	
VIRTbw	Maximum bandwidth available to the traffic class. **Virtbw** determines the maximum rate at which data can leave the internal queues to be transmitted onto the physical media. This rate is not equivalent to the transmission rate for data seen on the line, because the actual transmission rate includes the transmission of bits for the inter-frame-gap and the preamble of the Layer 2 headers. For example, 10 Mbps of data leaving the internal queues is not equivalent to 10 Mbps of data transmitted on the line. *rate* is from 1 to 16000000 kilobits per second, specified in Kbps, Mbps or Gbps (in upper or lower case). If you do not specify a unit, it uses kbps. If you specify Mbps or Gbps, *rate* may contain a decimal fraction with up to 3 decimal places, for example, 1.25 Mbps. Default: **none** (bandwidth is not limited)	
WEIght	Weight given to the traffic class, from 0 to 100. Specifying **weight** in traffic classes sets their policy (or intermediate traffic class) to schedule queues according to the relative weights of all its traffic classes. If a traffic class has a weight of 0 (zero), the router only empties its queue once the queues of all its sibling traffic classes are empty. **Priority** and **weight** are mutually exclusive. Use the **priority** parameter to create a hierarchy based on the priority of flows, for strict priority queuing. Use the **weight** parameter to create a hierarchy with weighted flows, for WRR or DWRR queuing. If you create a mixed hierarchy the priority queues are emptied first, giving low latency queuing behaviour. Default: no default, because the default behaviour is priority-based hierarchies.	
WEIGhtscheduler	Queue scheduling method that the router uses to schedule this traffic class' weighted sub traffic classes. This parameter is only valid if the sub traffic classes specify the **weight** parameter. Default: **wrr**	
	WRr	The router uses a weighted round robin scheme to empty the queues of weighted sub traffic classes.
	DWrr	The router uses a deficit weighted round robin scheme to empty the queues of weighted sub traffic classes. DWRR is less biased towards large packets than WRR.

Example

To modify traffic class 1 so that it has a moderately-high priority, and use DWRR to schedule the queues of the traffic class' sub classes, use the command:

```
set sqos tr=1 prio=10 weig=dw
```

Related Commands

add sqos policy trafficclass
add sqos trafficclass classifier
add sqos trafficclass dar
add sqos trafficclass subclass
create sqos trafficclass
delete sqos policy trafficclass
delete sqos trafficclass classifier
delete sqos trafficclass dar
delete sqos trafficclass subclass
destroy sqos trafficclass
show sqos trafficclass

show sqos

Syntax `SHow SQOS`

Description This command displays general information about software QoS.

Figure 39-31: Example output from the **show sqos** command

```
Software QoS Module
------------------------------------------------------------
  Status:
    SQoS Module Enabled.. YES
  Number of:
    Policies.............. 3
    Traffic Classes...... 7
    Meters................ 3
    RED Curves............ 3
    DSCP Maps............. 3
    Interfaces............ 2
    DAR Objects........... 3
  Debug Information:
    Debug Device......... 16
    Debug Flags.......... ERROR
```

Table 39-23: Parameters in output of the **show sqos** command

Parameter	Meaning
SQoS Module Enabled	Whether software QoS is enabled.
Number of:	Total number of each type of software QoS object.
Policies	Total number of software QoS policies.
Traffic Classes	Total number of software QoS traffic classes.
Meters	Total number of software QoS meters.
RED Curves	Total number of software QoS RED curve sets, including the 3 default RED curve sets.
DSCP Maps	Total number of software QoS DSCP maps, including the default map.
Interfaces	Total number of interfaces at Layers 1 and 2, and Layer 3 tunnels that have software QoS policies and/or DAR objects attached.
DAR Objects	Total number of software QoS Dynamic Application Recognition objects.
Debug information:	Information about debugging settings, if debugging is enabled.
Debug Device	Device to which debug messages are sent.
Debug Flags	Types of debugging that are enabled.

Example To find out how many software QoS policies you have created, and other general information, use the command:

```
sh sqos
```

Related Commands

disable sqos
disable sqos debug
enable sqos
enable sqos debug
show sqos counters

show sqos counters

Syntax

```
SHow SQOS COUnters CLASSifier[=id-list]
   [DIrection={In|OUT|TUNnel|ALL}] [INTerface=interface]
   [TRafficclass=id-list]

SHow SQOS COUnters DAR[=id-list]
   [DIrection={In|OUT|TUNnel}] [INTerface=interface]

SHow SQOS COUnters POLIcy[=id-list]
   [DIrection={In|OUT|TUNnel}] [INTerface=interface]

SHow SQOS COUnters TRafficclass[={id-list|DEFault|SYStem}]
   [DIrection={In|OUT|TUNnel|ALL}] [INTerface=interface]
```

Description This command displays counter information for the specified software QoS objects.

Parameter	Description	
CLASSifier	Classifier for which to display counters. An integer from 0 to 9999, a range of integers separated by hyphens, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once. If the classifier is used in more than one traffic class, the command shows a set of counters for each use of the classifier, unless you limit the display to one traffic class. Default: no default (classifier counters are not displayed)	
DAR	DAR object for which to display counters. An integer from 0 to 9999, a range of integers separated by hyphens, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once. Default: no default (DAR object counters are not displayed)	
DIrection	Filter that restricts the command so that the router displays only counters for software QoS objects with this direction. Default: no default (not filtered by direction)	
	In	Display counters for software QoS objects that act on the packet at ingress.
	OUT	Display counters for software QoS objects that act on the packet at egress.
	TUNnel	Display counters for software QoS objects that act on tunnelled packets.
	ALL	Display counters for software QoS objects that act on packets in any direction. **All** is only valid with **trafficclass** and **classifier**.

Parameter	**Description** (cont.)	
INTerface	Filter that restricts the command so that the router displays only counters for software QoS objects that are associated with this interface or tunnel. Valid entry types are Layer 1 and 2 interfaces: ● eth (such as. eth0) ● ATM channel (such as atm0.0) ● frame relay (such as fr0) ● PPP (such as ppp0) ● the switch instance (such as swi0) Layer 3 tunnels: ● GRE (such as gre1) ● IPv6 6-to-4 virtual interface (such as virt9) ● the name of an IPSec policy (such as ipsec-*policyname*) To see a list of current valid Layer 1 and 2 interfaces, use the **show interface** command on page 9-72 of Chapter 9, Interfaces. Default: no default (not filtered by interface)	
POLIcy	Policy for which to display counters. An integer from 0 to 9999, a range of integers separated by hyphens, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once. If the policy is used on more than one interface, the command shows a set of counters for each use of the policy, unless you limit the display to one interface. Default: no default (policy counters are not displayed)	
TRafficclass	Traffic class for which to display counters. If you specify both **classifier** and **trafficclass** the router displays the classifier counters for that traffic class; otherwise, it displays traffic class counters. If the traffic class is used on a policy that is attached to more than one interface, the command shows a set of counters for each use of the traffic class, unless you limit the display to one interface. Default: no default	
	id-list	An integer from 0 to 9999, a range of integers separated by hyphens, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once.
	DEFault	Default traffic class.
	SYStem	System traffic class.

Figure 39-32: Example output from the **show sqos counters classifiers** command

```
Classifier 8: (Interface=eth0 TC=6 Direction=Egress)
-------------------------------------------------------------
  Hit Counters
    Packets Matched.. 0
    Bytes Matched.... 0
```

Table 39-24: Parameters in output of the **show sqos counters classifiers** command

Parameter	Meaning
Classifier	ID of the classifier to which the counters apply.
Interface	Interface to which this set of classifier counters applies.
Traffic Class	Traffic class to which this set of classifier counters applies.
Direction	Whether this set of classifier counters applies to ingress traffic (In), egress traffic (Out) or tunnelled traffic (Tunnel).
Hit Counters	Information about the classifier matches since the router last restarted or the counters were last reset.
Packets Matched	Number of packets that the classifier classified.
Bytes Matched	Number of bytes of data that the classifier classified.

Figure 39-33: Example output from the **show sqos counters dar** command

```
DAR Object 1
-------------------------------------------------------------
  Session Counters (by protocol)
    Total Sessions Recognised.. 3
    RTSP Sessions Recognised... 0
    SIP Sessions Recognised.... 3
    H323 Sessions Recognised... 0
  Session Counters (by media)
    Active Sessions............ 3
    Voice Sessions Started..... 1
    Video Sessions Started..... 1
  Dynamic Classifiers
    Classifer=10002 tc=2 ip=10.33.25.17/32 port=69-69
    Classifer=10000 tc=2 ip=10.33.25.17/32 port=49170-49171
    Classifer=10001 tc=2 ip=10.33.25.17/32 port=51372-51373
```

Table 39-25: Parameters in output of the **show sqos counters dar** command

Parameter	Meaning
DAR Object	ID of the DAR object to which the counters apply.
Session Counters (by protocol)	Information about the number of sessions the DAR object recognised since the router last restarted or the counters were last reset, sorted according to the protocol they used for session setup.
Total Sessions Recognised	Total number of sessions that were recognised by the DAR object.
RTSP Sessions Recognised	Number of Real Time Streaming Protocol sessions that were recognised by the DAR object.
SIP Sessions Recognised	Number of Session Initiation Protocol sessions that were recognised by the DAR object.

Table 39-25: Parameters in output of the **show sqos counters dar** command

Parameter	Meaning
H323 Sessions Recognised	Number of H.323 sessions that were recognised by the DAR object.
Session Counters (by media)	Information about the number of sessions the DAR object recognised since the router last restarted or the counters were last reset, sorted according to the type of data they carried.
Active Sessions	Number of sessions that are currently successfully connected.
Voice Sessions Started	Total number of successful VoIP connections.
Video Sessions Started	Total number of successful video connections.
Dynamic Classifiers	Information about the dynamic classifiers created by the DAR object.
Classifier	ID number of the classifier. Within a traffic class, classifiers with a lower number take precedence.
tc	Traffic class to which the classifier is assigned.
ip	Source or destination IP address and mask that the classifier uses to identify traffic.
port	UDP or TCP port that the classifier uses to identify traffic.

Figure 39-34: Example output from the **show sqos counters policy** command

```
Policy 1: (Interface=eth0 Direction=Egress)
-------------------------------------------------------------------------------
  Packets Processed
    Passed (Packets)......... 59839                (Bytes).. 15076488
    Dropped (Packets)........ 164144               (Bytes).. 41364288
    Total Dropped (Packets).. 306888               (Bytes).. 77335776
  Meter Counters
    Meter.................... 0
    BWC 1 (Packets).......... 51517                (Bytes).. 12979344
    BWC 2 (Packets).......... 8322                 (Bytes).. 2097144
    BWC 3 (Packets).......... 2                    (Bytes).. 504
```

Table 39-26: Parameters in output of the **show sqos counters policy** command

Parameter	Meaning
Policy	ID of the policy to which the counters apply.
Interface	Interface to which this set of policy counters applies.
Direction	Whether this set of policy counters applies to ingress traffic, egress traffic, or tunnelled traffic.
Packets Processed	**Information about the packets the policy processed since the router last restarted or the counters were last reset.**
Passed (Packets) ... (Bytes)	Number of packets and bytes of data processed and forwarded by the policy tree.
Dropped (Packets) ... (Bytes)	Number of packets and byte of data dropped by the policy's queueing and metering at the last stage before transmission. This does **not** include packets or bytes dropped by the root traffic class, the system traffic class, the default traffic class, or any user-defined traffic classes.
Total Dropped (Packets) ... (Bytes)	Number of packets and bytes of data dropped by the policy tree. This includes packets and bytes dropped by the root traffic class, the system traffic class, the default traffic class, any user-defined traffic classes, any meter, and the policy queueing mechanism.

Table 39-26: Parameters in output of the **show sqos counters policy** command

Parameter	Meaning
Meter Counters	**Information about the packets metered by the policy to measure their bandwidth use and conformance, since the router last restarted or the counters were last reset.**
Meter	ID of the meter that the policy uses.
BWC 1 (Packets) ... (Bytes)	Number of packets and bytes of data that the meter assigned to bandwidth class 1 (green).
BWC 2 (Packets) ... (Bytes)	Number of packets and bytes of data that the meter assigned to bandwidth class 2 (yellow).
BWC 3 (Packets) ... (Bytes)	Number of packets and bytes of data that the meter assigned to bandwidth class 3 (red).

Figure 39-35: Example output from the **show sqos counters trafficclass** command

```
Traffic Class 4: (Interface=eth0 Direction=Egress)
------------------------------------------------------------------------------
  Packets Processed
    Passed (Packets)..................... 0            (Bytes).. 0
    Dropped (Packets).................... 0            (Bytes).. 0
    Classifiers.......................... 2
  Queue Counters
    Current Queue Length (Packets)....... 0            (Bytes).. 0
    Avg Queue Length (Last Sec) (Packets)... 0         (Bytes).. 0
    Avg Queue Length (Last Min) (Packets)... 0         (Bytes).. 0
    Avg Queue Length (Last Hour) (Packets).. 0         (Bytes).. 0
    Avg Latency (microseconds)........... 0
  Meter Counters
    Meter................................ 1
    BWC 1 (Packets)...................... 0            (Bytes).. 0
    BWC 2 (Packets)...................... 0            (Bytes).. 0
    BWC 3 (Packets)...................... 0            (Bytes).. 0
  RED Curve Counters
    Red Curve............................ 0
    BWC 1 Dropped (Packets).............. 0            (Bytes).. 0
    BWC 2 Dropped (Packets).............. 0            (Bytes).. 0
    BWC 3 Dropped (Packets).............. 0            (Bytes).. 0
```

Table 39-27: Parameters in output of the **show sqos counters trafficclass** command

Parameter	Meaning
Traffic Class	ID of the traffic class to which the counters apply.
Interface	Interface to which this set of traffic class counters applies.
Direction	Whether this set of traffic class counters applies to ingress traffic, egress traffic or tunnelled traffic.
Packets Processed	Information about the packets processed by the traffic class since the router last restarted or the counters were last reset.
Packets Passed	Number of packets the traffic class processed and forwarded.
Packets Dropped	Number of packets the traffic class dropped for any reason.
Bytes Passed	Number of bytes of data the traffic class processed and forwarded.
Bytes Dropped	Number of bytes of data the traffic class dropped for any reason.
Classifiers	Classifiers attached to the traffic class.

Table 39-27: Parameters in output of the **show sqos counters trafficclass** command

Parameter	Meaning
Queue Counters	Information about the traffic class queue since the router last restarted or the counters were last reset.
Current Queue Length	Number of packets currently queued by the traffic class.
Avg Queue Length (Last Sec)	Average number of packets and bytes queued by the traffic class at any one time, averaged over the last second.
Avg Queue Length (Last Min)	Average number of packets and bytes queued by the traffic class at any one time, averaged over the last minute.
Avg Queue Length (Last Hour)	Average number of packets and bytes queued by the traffic class at any one time, averaged over the last hour.
Avg Latency	Average length of time that packets spend in software QoS queues, in microseconds. Dropped packets are not counted. Latency is averaged since the router last restarted or the counters were last reset.
Meter Counters	Information about the packets metered by the traffic class to measure their bandwidth use and conformance, since the router last restarted or the counters were last reset.
Meter	ID of the meter that the traffic class uses.
BWC 1 (Packets) ... (Bytes)	Number of packets and bytes of data that the meter assigned to bandwidth class 1 (green).
BWC 2 (Packets) ... (Bytes)	Number of packets and bytes of data that the meter assigned to bandwidth class 2 (yellow).
BWC 3 (Packets) ... (Bytes)	Number of packets and bytes of data that the meter assigned to bandwidth class 3 (red).
RED Curve Counters	Information about the packets dropped by the traffic class' RED curve, since the router last restarted or the counters were last reset.
RED Curve	The ID of the RED curve that the traffic class uses.
BWC 1 Dropped (Packets) ... (Bytes)	Number of packets and bytes of data in bandwidth class 1 (green) that were dropped.
BWC 2 Dropped (Packets) ... (Bytes)	Number of packets and bytes of data in bandwidth class 2 (yellow) that were dropped.
BWC 3 Dropped (Packets) ... (Bytes)	Number of packets and bytes of data in bandwidth class 3 (red) that were dropped.

Example To display the number of packets and bytes of data processed by policy 1 on ppp0, use the command:

```
sh sqos cou poli=1 int=ppp0
```

Related Commands

disable sqos debug
enable sqos debug
reset sqos counters
show sqos

show sqos dar

Syntax `SHow SQOS DAR[={id-list|ALL}] [FULl|SUMmary] [SHOwunused={Yes|No}]`

Description This command displays information about one or more DAR objects.

Parameter	Description
DAR	DAR object for which to display information. An integer from 0 to 9999, a range of integers separated by hyphens, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once. Default: **all**
FULl	Detailed information about the DAR object. This is the default if you specify a single DAR object.
SUMmary	Summary table of information about the DAR object. This is the default if you specify multiple DAR objects.
SHOwunused	Whether the output displays an entry for a parameter if the DAR object has no value for that parameter (for example, displays **Dst IP** when you have not specified a destination IP address in the DAR object). Default: **no**

Figure 39-36: Summary example output from the **show sqos dar summary** command

```
Id     Src IP             Dst IP             Protocol   Codec   Interfaces
-------------------------------------------------------------------------------
   1   192.168.1.0/24     192.168.100.0/24        ALL     ANY   eth0
   2   192.168.2.0/24     192.168.200.0/24        ALL     ANY   eth1
```

Figure 39-37: Full example output from the **show sqos dar=1** command

```
Id..................... 1
    Src IP.............. 2001:0DB8::1/32
    Protocol............ ALL
    Codec............... ANY
    Inactivity Timeout.. 600
    SIP Port............ 5060
    RTSP Port........... 554
    H323 Port........... 1720
    Interfaces.......... eth0
    Policies............ 1
    Traffic Classes..... 1
```

Table 39-28: Parameters in output of the **show sqos dar** command

Parameter	Meaning
ID	ID number of the DAR object.
Description	Description of the DAR object, if it has one.
Src IP	Source IPv4 or IPv6 address, or subnet if it includes a CIDR mask or prefix length. The DAR object matches traffic flows from only that address or network.
Dst IP	Destination IPv4 or IPv6 address, or subnet if it includes a CIDR mask or prefix length. The DAR object only matches traffic flows destined for that address or network.
Protocol	Protocol that the DAR object uses to match packets. "All" indicates that the DAR object ignores the protocol.
Codec	Coder/decoder that the DAR object uses to match packets; one of "Any" (ignores the codec), "Audio" (matches traffic flows that use any audio codec), or "Video" (matches traffic flows that use any video codec).
Inactivity Timeout	Idle period in seconds before which an entry for a classified flow is deleted. "None" indicates that the flow entry is never deleted.
SIP Port	UDP port over which SIP messages are received.
RTSP Port	TCP port over which the RTSP session control messages are received
H323 Port	TCP port over which H.323 session control messages are received.
Interfaces	Interfaces to which the DAR object is attached. The DAR object recognises session initiation packets on this interface and uses them to create classifiers for packets from those sessions.
Policies	Policies to which the DAR object is attached through its traffic classes.
Traffic Classes	Traffic classes to which the DAR object is attached.

Example To display summary information about all DARs, use the command:

```
sh sqos dar
```

To display detailed information about all DARs, use the command:

```
sh sqos dar ful
```

Related Commands

add sqos interface dar
add sqos trafficclass dar
create sqos dar
delete sqos interface dar
delete sqos trafficclass dar
destroy sqos dar
set sqos dar

show sqos dscpmap

Syntax `SHow SQOS DSCPMap[={`*`id-list`*`|ALL}] [FULl|SUMmary]`

Description This command displays information about one or more DSCP maps.

Parameter	Description
DSCPMap	DSCP map for which to display information. An integer from 0 to 9999, a range of integers separated by hyphens, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once. Default: **all**
FULl	Detailed information about the DSCP map. This is the default if you specify a single DSCP map.
SUMmary	Summary table of information about the DSCP map. This is the default if you specify multiple DSCP maps.

Figure 39-38: Summary example output from the **show sqos dscpmap summary** command

```
Id
Map  Description                    Policy Refs
---------------------------------------------
   1 Primary DSCP Map                1
   2 Secondary DSCP Map              2
   3 Auxilary DSCP Map
```

Figure 39-39: Full example output from the **show sqos dscpmap=1 full** command

```
                              Remarking
                               ----------------------------------------
                 Premarking    BW Class 1     BW Class 2     BW Class 3
                 ------------  ------------   ------------   ------------
Id      Old      New      New  New      New   New      New   New      New
Map     DSCP     BWClass  DSCP BWClass  DSCP BWClass  DSCP BWClass  DSCP
-------------------------------------------------------------------------
   1
          1          1   63         1    1         2    1         3    1
          2          1   63         1    2         2    2         3    2
          3          1   63         1    3         2    3         3    3
          4          1   63         1    4         2    4         3    4
          5          1   63         1    5         2    5         3    5
          6          1   63         1    6         2    6         3    6
          7          1   63         1    7         2    7         3    7
          8          1   63         1    8         2    8         3    8
          9          1   63         1    9         2    9         3    9
         10          1   63         1   10         2   10         3   10
```

Table 39-29: Parameters in output of the **show sqos dscpmap** command

Parameter	Meaning
ID Map	ID number of the DSCP map.
Description	Description of the DSCP map, if it has one.
Policy Refs	Policies that use this DSCP map.
Old DSCP	The packet's existing DSCP value. For premarking, this is the DSCP the packet had at ingress. For re-marking, it is the DSCP the packet had at ingress unless premarking or earlier re-marking changed it.
Premarking	The premarking table, which uses the ingress (old) DSCP to determine a bandwidth class and/or DSCP for the packet. Premarking happens before metering.
New BW Class	Bandwidth class to which the map assigns packets that have the given "Old DSCP" at ingress.
New DSCP	DSCP which the map writes into packets that have the given "Old DSCP" at ingress.
Remarking	The re-marking table, which uses the current (old) DSCP and bandwidth class to determine a new DSCP and/or bandwidth class for the packet. Re-marking happens after metering, after the packet is dequeued from the leaf traffic class.
New BW Class	For each bandwidth class and old DSCP, the bandwidth class to which the packet is assigned.
New DSCP	For each bandwidth class and old DSCP, the DSCP value that is written into the packet header before egress.
BW Class 1 BW Class 2 BW Class 3	Current bandwidth class. Generally, this is the bandwidth class to which the meter assigned the packet.

Example To display the information about DSCP map 1 that Figure 39-39 on page 39-148 illustrates, use the command:

```
show sqos dscpmap=1 full
```

This map is used to premark the DSCP value of all incoming packets to 63 and to assign them to bandwidth class 1.

Related Commands

create sqos dscpmap
create sqos trafficclass
destroy sqos dscpmap
set sqos dscpmap

show sqos interface

Syntax `SHow SQOS INTerface[={`*interface*`|ALL}]`

Description This command displays information about the software QoS policies attached to one or more interfaces at Layers 1 and 2, or Layer 3 tunnels.

Parameter	Description
INTerface	Interface or tunnel for which to display information. Valid entry types are: Layer 1 and 2 interfaces: ● eth (such as. eth0) ● ATM channel (such as atm0.0) ● frame relay (such as fr0) ● PPP (such as ppp0) ● the switch instance (such as swi0) Layer 3 tunnels: ● GRE (such as gre1) ● IPv6 6-to-4 virtual interface (such as virt9) ● the name of an IPSec policy (such as ipsec-*policyname*) To see a list of current valid Layer 1 and 2 interfaces, use the **show interface** command on page 9-72 of Chapter 9, Interfaces. Default: **all**

Figure 39-40: Example output from the **show sqos interface** command

```
Interface          In Policy     Out Policy    Tunnel Policy   DAR Objects
-------------------------------------------------------------------------
eth0                       1             21                        2
ppp0                       2             22                        3,5
ipsec-central                                              41    4
-------------------------------------------------------------------------
```

Table 39-30: Parameters in output of the **show sqos interface** command

Parameter	Meaning
Interface	Interface at Layer 1 or 2, Layer 3 tunnel, or IPsec policy that the policies and DAR objects are acting on.
In Policy	Policy that the router applies to ingress traffic on this Layer 1 or 2 interface.
Out Policy	Policy that the router applies to egress traffic on this Layer 1 or 2 interface.
Tunnel Policy	Policy that the router applies to traffic processed by this Layer 3 tunnel or IPsec policy.
DAR Objects	DAR objects that are attached to the interface. The DAR objects recognise session initiation packets on this interface and use them to create classifiers for packets from those sessions.

Example To find out which DAR object and policies are attached to each interface, use the command:

```
sh sqos int
```

Related Commands

create sqos policy
delete sqos policy trafficclass
destroy sqos policy
set sqos interface
set sqos policy
show sqos policy

show sqos meter

Syntax `SHow SQOS METer[={`*id-list*`|ALL}] [FULl|SUMmary]`

Description This command displays information about one or more meters. Meters measure bandwidth usage and conformance.

Parameter	Description
METer	Meter for which to display information. An integer from 0 to 9999, a range of integers separated by hyphens, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once. Default: **all**
FULl	Detailed information about the meter. This is the default if you specify a single meter.
SUMmary	Summary table of information about the meter. This is the default if you specify multiple meters.

Figure 39-41: Summary example output from the **show sqos meter summary** command

```
                Bandwidth       Burst Size
Id      Type    Min     Max     Min     Max     Traffic Classes
-------------------------------------------------------------------------------
   1    SRTCM           1Mbps   10kB    10kB    4
   2    SRTCM           1Mbps   10kB    10kB
   3    TRTCM   1Mbps   1Mbps   10kB    10kB
```

Figure 39-42: Full example output from the **show sqos meter full** command

```
Id................... 1
    Meter Type....... SRTCM
    Max Bandwidth.... 1Mbps
    Min Burst Size... 10kB
    Max Burst Size... 10kB
    Traffic Classes.. 4

Id................... 2
    Meter Type....... SRTCM
    Max Bandwidth.... 1Mbps
    Min Burst Size... 10kB
    Max Burst Size... 10kB

Id................... 3
    Meter Type....... TRTCM
    Min Bandwidth.... 1Mbps
    Max Bandwidth.... 1Mbps
    Min Burst Size... 10kB
    Max Burst Size... 10kB
```

Table 39-31: Parameters in the output of the **show sqos meter** command

Parameter	Meaning
ID	ID number of the meter.
Meter Type	Whether the meter is a Single Rate Three Colour Marker of RFC 2697 (SRTCM) or a Two Rate Three Colour Marker of RFC 2698 (TRTCM).
Description	Description of the meter, if it has one.
Min Bandwidth	For the two rate meter of RFC 2698, the Committed Information Rate (CIR) of the RFC. See "Metering: Bandwidth Conformance" on page 39-21 for a description of CIR.
Max Bandwidth	For the single rate meter of RFC 2697, the highest rate at which a steady stream of packets can arrive at the meter and be assigned to bandwidth class 1 (conformant, green). This is the Committed Information Rate (CIR) of the RFC. For the two rate meter of RFC 2698, the Peak Information Rate (PIR) of the RFC. See "Metering: Bandwidth Conformance" on page 39-21 for a description of PIR.
Min Burst Size	For the single rate meter of RFC 2697, the amount by which a packet can exceed **maxbandwidth** and still possibly be assigned to bandwidth class 1. This is the Committed Burst Size (CBS) of the RFC. For the two rate meter of RFC 2698, the amount by which a packet can exceed **minbandwidth** and still possibly be assigned to bandwidth class 1. This is the Committed Burst Size (CBS) of the RFC.
Max Burst Size	For the single rate meter of RFC 2697, the amount by which a packet can exceed **maxbandwidth** plus **minburstsize** and still possibly be assigned to bandwidth class 2. This is the Excess Burst Size (EBS) of the RFC. For the two rate meter of RFC 2698, the amount by which a packet can exceed **maxbandwidth** and still possibly be assigned to bandwidth class 2. This is the Peak Burst Size (PBS) of the RFC.
Traffic Classes	Traffic classes that use the meter.
Ignore Bandwidth Class	Whether the meter acknowledges any previous bandwidth class assigned to packets. "Yes" indicates that the metering function is colour blind and ignores any bandwidth class previously assigned to packets. It sets the meter bandwidth class according to only the metered conformance level of the flow. "No" indicates that the metering function is colour aware and uses any bandwidth class previously assigned to packets, as well as the metered conformance level, to set the bandwidth class.

Example To get a list of the available meters, use the command:

```
sh sqos met
```

Related Commands

create sqos trafficclass
create sqos meter
destroy sqos meter
set sqos meter

show sqos policy

Syntax

```
SHow SQOS POLIcy[={id-list|ALL}] [FULl|SUMmary|TREE]
   [SHOwunused={Yes|No}]
```

Description This command displays information about one or more software QoS policies.

Parameter	Description
POLIcy	Policy for which to display information. An integer from 0 to 9999, a range of integers separated by hyphens, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once. Default: **all**
FULl	Detailed information about the policy. This is the default if you specify a single policy.
SUMmary	Summary table of information about the policy. This is the default if you specify multiple policies.
TREE	Tree diagram of the traffic class hierarchy that is attached to this policy.
SHOwunused	Whether the output displays an entry for a parameter if the policy has no value for that parameter (for example, displays **DSCP Map** when you have not specified a DSCP map for the policy). Default: **no**

Figure 39-43: Summary example output from the **show sqos policy summary** command

```
                DSCP   Virt    Wt
Id      Mtr     Map    BW      Schd   Interfaces
-------------------------------------------------
   1              1    -       WRR    eth0
   2              1    -       DWRR   eth1,eth2
```

Table 39-32: Parameters in the summary output of the **show sqos policy** command

Parameter	Meaning
ID	ID number of the policy.
Mtr	Meter that the policy uses.
DSCP map	DSCP map that the policy uses.
Virt BW	Maximum bandwidth available to the policy. Virtual BW determines the maximum rate at which data can leave the internal queues to be transmitted onto the physical media.
Wt Schd	Queue scheduling method for weighted traffic classes that belong to the policy, either weighted round robin (WRR) or deficit weighted round robin (DWRR).
Interfaces	Layer 1 and 2 interfaces, layer 3 tunnels and IPsec policies that use this software QoS policy.

Figure 39-44: Full example output from the **show sqos policy full** command

```
Id.............................. 1
    Pause Time................... 10
    Maximum Queue Length (pkts).. 64
    Weight Scheduler............. WRR
    DSCP Map..................... 1
    System Traffic Weight........ 20
    Traffic Classes.............. 1 2 3
    DAR Objects.................. 1,2,3

Id.............................. 2
    Pause Time................... 10
    Maximum Queue Length (pkts).. 64
    Weight Scheduler............. WRR
    DSCP Map..................... 2
    Default Traffic Class........ 7
    System Traffic Weight........ 20
    Traffic Classes.............. 4 5 6 7*
    Interfaces (out)............. eth0,eth1

Id.............................. 3
    Pause Time................... 10
    Maximum Queue Length (pkts).. 64
    Weight Scheduler............. WRR
    System Traffic Weight........ 20
```

Table 39-33: Parameters in the full output of the **show sqos policy** command

Parameter	Meaning
ID	ID number of the policy.
Meter	Meter that the policy uses.
Description	Description of the policy, if it has one.
Pause Time	Length of time, in seconds, for which the router does not dequeue packets from a paused flow.
Pause Action	Notification action the router takes when it pauses a non-conformant traffic flow that belongs to this policy: Log: The router generates a log message Trap: The router generates an SNMP trap Both: The router generates both a log message and an SNMP trap None: The router does not generate a notification.
BW Class 3 Action	Action the router takes on Bandwidth Class 3 packets (red coloured packets): ● Drop: The router drops non-conformant packets ● Pause: The router drops non-conformant packets and stops dequeuing packets from the flow for Pause Time seconds. ● None: The router sends non-conformant packets to the next processing stage.
Ignore Pre-NAT Information	Whether classifiers attached to the policy use pre-NAT IP settings for classification because these contain the distinguishing information, one of No (uses pre-NAT settings) or Yes (uses post-NAT settings).
Remark	How the router sets the bandwidth class and/or DSCP value in the packet header's Differentiated Services field after metering.

Table 39-33: Parameters in the full output of the **show sqos policy** command (cont.)

Parameter	Meaning
Remark VLAN Priority	Setting for the 802.1p VLAN priority field of the frame's Ethernet header. "None" indicates that the router does not reset the VLAN priority.
Virtual BW	Maximum bandwidth available to the policy. Virtual BW determines the maximum rate at which data can leave the internal queues to be transmitted onto the physical media.
Weight scheduler	Queue scheduling method for weighted traffic classes that belong to the policy, either weighted round robin (WRR) or deficit weighted round robin (DWRR).
DSCP Map	DSCP map assigned to the policy.
System traffic weight	Percentage of the interface's maximum bandwidth from 5% to 50% that the router reserves for system traffic.
Traffic classes	Traffic classes that are attached to this policy. Parentheses show the hierarchy of sub traffic classes. If you have set a default class, it is indicated by an *.
Interfaces (in)	Layer 1 and 2 interfaces that use this policy for ingress traffic.
Interfaces (out)	Layer 1 and 2 interfaces that use this policy for egress traffic.
Interfaces (tunnel)	Layer 3 tunnels or IPsec policies that use this policy.
Default traffic class	Traffic class that the router applies to unclassified traffic on the policy's interface. The class policies use by default is called "Default".

Figure 39-45: Tree example output from the **show sqos policy tree** command

```
                Actual                          Virt
Tree            Scheduler   Priority   Weight   BW      Classifiers
-------------------------------------------------------------------------------
1               PQ
  1             FIFO               15                   1
  2             FIFO               14
  3             FIFO               13

2               PQ
  4             FIFO               15                   2
  5             FIFO               14                   8
  7*            FIFO               1

3               FIFO
```

Table 39-34: Parameters in the tree output of the **show sqos policy** command

Parameter	Meaning
Tree	The policy ID and its traffic class hierarchy. The first entry is the policy. The sub traffic classes belonging to a traffic class are shown indented below that traffic class (so in the example above, traffic classes 3 and 4 are attached to traffic class 2).
Actual Scheduler	Scheduling algorithm that the router uses to schedule queues at this level of the hierarchy. For the top level of the tree, PQ indicates that only priority traffic classes are attached to the policy (apart from the system and default classes). WRR and DWRR indicate that only weighted classes are attached. PQ+WRR and PQ+WDRR indicate that a mix of priority and weighted traffic classes are attached.
Priority	For priority queue based traffic classes, the priority. The range is 1 to 15, and a higher value means a higher priority.
Weight	For weighted traffic classes, the weight. The range is 0 to 100, and a higher value means a higher weight.
Virt BW	Maximum bandwidth available to the policy. Virtual BW determines the maximum rate at which data can leave the internal queues to be transmitted onto the physical media.
Classifiers	Classifiers that each leaf traffic class uses.

Example To see the traffic class hierarchy that makes up policy 1 and information about the order in which it will dequeue packets, use the command:

```
sh sqos poli=1 tree
```

Related Commands

add sqos policy trafficclass
create sqos policy
delete sqos policy trafficclass
destroy sqos policy
set sqos policy

show sqos red

Syntax SHow SQOS RED[={*id-list*|ALL}] [FULl|SUMmary] [SHOwunused={Yes|No}]

Description This command displays information about one or more RED curve sets.

Parameter	Description
RED	RED curve set for which to display information. An integer from 0 to 9999, a range of integers separated by hyphens, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once. Default: **all**
FULl	Detailed information about the RED curve set. This is the default if you specify a single curve set.
SUMmary	Summary table of information about the RED curve set. This is the default if you specify multiple curve sets.
SHOwunused	Whether the output displays an entry for a parameter if the RED curve set has no value for that parameter (for example, displays **Traffic Classes** when no traffic class uses the RED curve set). Default: **no**

Figure 39-46: Summary example output from the **show sqos red summary** command

```
                 BW Class 1        BW Class 2        BW Class 3
                 --------------    --------------    --------------
                 Start Stop Drop   Start Stop Drop   Start Stop Drop
Id     Average             Prob              Prob              Prob
-------------------------------------------------------------------
   0         98      35   50   20      20   35   30      10   20   40
   1         98      50   70   20      30   50   30      15   30   40
   2         98      80   95   20      60   80   30      40   60   40
```

Figure 39-47: Full example output from the **show sqos red=0,2 full** command

```
Id..................... 0
    Description......... Aggressive
    Averaging........... 98
    1 Start............. 35
      Stop.............. 50
      Drop Probability.. 20
    2 Start............. 20
      Stop.............. 35
      Drop Probability.. 30
    3 Start............. 10
      Stop.............. 20
      Drop Probability.. 40
    Traffic Classes..... 2

Id..................... 2
    Description......... Passive
    Averaging........... 98
    1 Start............. 80
      Stop.............. 95
      Drop Probability.. 20
    2 Start............. 60
      Stop.............. 80
      Drop Probability.. 30
    3 Start............. 40
      Stop.............. 60
      Drop Probability.. 40
    Traffic Classes..... 1,3
```

Table 39-35: Parameters in output of the **show sqos red** command

Parameter	Meaning
ID	ID number of the RED curve set.
Description	Description of the RED curve set, if it has one.
Averaging	Weight used in the moving averaging estimation of queue length for the RED curve algorithm, expressed as a percentage of the current average. If Averaging is too high, the estimated average queue size responds too slowly to transient congestion. If Averaging is too low, the estimated average queue size tracks the instantaneous queue size too closely and you lose the benefits of RED.
For each of bandwidth class 1 (green), 2 (yellow), and 3 (red):	
Start	Percentage of the queue length at which the RED algorithm starts to drop packets.
Stop	Percentage of the queue length at which the RED algorithm is dropping Drop Probability percent of the packets. Beyond this point, 100% of the packets are dropped.
Drop Probability	Probability that a packet will be dropped at the Stop queue length.
Traffic Classes	Traffic classes that use this RED curve set

Example To see the cut-off values for each RED curve and their descriptions, use the command:

```
sh sqos red ful
```

Related Commands

create sqos red
create sqos trafficclass
destroy sqos red
set sqos red

show sqos trafficclass

Syntax

```
SHow SQOS TRafficclass[={id-list|ALL}] [FULl|SUMmary]
   [SHOwunused={Yes|No|ON|OFf}]
```

Description This command displays information about one or more traffic classes.

Parameter	Description
TRafficclass	Traffic class for which to display information. An integer from 0 to 9999, a range of integers separated by hyphens, or a comma-separated list of integers and/or ranges (for example 0,3,4-9). An integer cannot appear in the list more than once. Default: **all**
FULl	Detailed information about the traffic class. This is the default if you specify a single traffic class.
SUMmary	Summary table of information about the traffic class. This is the default if you specify multiple traffic classes.
SHOwunused	Whether the output displays an entry for a parameter if the traffic class has no value for that parameter (for example, displays **Pause Mode** when you have not configured the traffic class to pause traffic). Default: **no**

Figure 39-48: Summary example output from the **show sqos trafficclass summary** command

```
                                 Max
                Red       Virt   QLen     Wt
Id     Mtr      Curve     BW     (pkts)   Schd   Policy   Sub-classes
-------------------------------------------------------------------------
   1       1         0    -          64   WRR         3        13,18
  13       2         1    -          64   WRR
  18       3         2    -          64   WRR
  23       4         3    -          64   WRR         6
```

Table 39-36: Parameters in the summary output of the **show sqos trafficclass** command

Parameter	Meaning
ID	ID number of the traffic class.
Mtr	Meter that the traffic class uses.
RED Curve	RED curve that the traffic class uses.
Virt BW	Maximum bandwidth available to the traffic class. Virtual BW determines the maximum rate at which data can leave the internal queues to be transmitted onto the physical media.
Max Qlen (pkts)	Maximum queue length, in packets, for the traffic class. The router drops packets that would exceed the maximum queue length.
Wt Schd	Queue scheduling method for weighted sub classes that belong to the traffic class, either weighted round robin (WRR) or deficit weighted round robin (DWRR).
Policy	Policy that uses the traffic class.
Sub-classes	Sub traffic classes that are attached to this traffic class.

Figure 39-49: Full example output from the **show sqos trafficclass=2** command

```
Id.............................. 2
    Description................. VoIP
    Pause Time.................. 10
    Pause Action................ None
    Premark BW Class............ 1
    Remark...................... USEDSCPMAP
    Remark VLAN Priority........ 6
    BW Class 3 Action........... Drop
    Virtual BW.................. None
    Maximum Queue Length (pkts).. 64
    Queue Limit Exceed Action.... None
    Weight Scheduler............ WRR
    Policy...................... 1
    Parent Class................ None
    Priority.................... 15
    Weight...................... None
    Sub Classes................. None
    Classifiers................. none
    DAR Objects.................1,2
```

Table 39-37: Parameters in the full output of the **show sqos trafficclass** command

Parameter	Meaning
ID	ID number of the traffic class.
Description	Description of the traffic class, if it has one.
Pause Time	Length of time, in seconds, for which the router does not dequeue packets from a paused flow.
Pause Action	Notification action the router takes when it pauses a non-conformant traffic flow that belongs to this traffic class: Log: The router generates a log message Trap: The router generates an SNMP trap Both: The router generates both a log message and an SNMP trap None: The router does not generate a notification.
Premark BW Class	How the router assigns the packet to a bandwidth class at the start of the QoS processing (before metering).
Premark DSCP	How the router changes the DSCP value in the packet header at the start of the QoS processing (before metering).
Remark	How the router sets the bandwidth class and/or DSCP value in the packet header's Differentiated Services field after metering.
Remark VLAN Priority	Setting for the 802.1p VLAN priority field of the frame's Ethernet header. "None" indicates that the router does not reset the VLAN priority.
BW Class 3 Action	Action the router takes on Bandwidth Class 3 packets (red coloured packets): ● Drop: The router drops non-conformant packets ● Pause: The router drops non-conformant packets and stops dequeuing packets from the flow for Pause Time seconds. ● None: The router sends non-conformant packets to the next processing stage.
Virtual BW	Maximum bandwidth available to the traffic class. Virtual BW determines the maximum rate at which data can leave the internal queues to be transmitted onto the physical media.

Table 39-37: Parameters in the full output of the **show sqos trafficclass** command (cont.)

Parameter	Meaning
Max Queue Length (pkts)	Maximum queue length in packets for the traffic class. The router drops packets that would exceed the maximum queue length.
Queue Limit Exceed Action	Notification action the router takes when a traffic flow exceeds the maximum queue length of the traffic class, one of: Log: The router generates a log message Trap: The router generates an SNMP trap Both: The router generates both a log message and an SNMP trap None: The router does not generate a notification.
Weight scheduler	Queue scheduling method for weighted sub classes that belong to the traffic class, either weighted round robin (WRR) or deficit weighted round robin (DWRR).
Policy	Policy that uses the traffic class.
Parent Class	Intermediate traffic class to which this traffic class is attached.
Priority	Priority from 0 to 15 of the traffic class. The policy (or intermediate traffic class) schedules queues of priority-based traffic classes according to the relative priorities of all its traffic classes. The router services queues from the traffic class with the highest value for Priority first.
Weight	Weight given to the traffic class, from 0 to 100. The policy (or intermediate traffic class) schedules queues of weighted traffic classes according to the relative weights of all its traffic classes, using WRR or DWRR. If a traffic class has a weight of 0 (zero), the router only empties its queue once the queues of all its sibling traffic classes are empty.
Sub Classes	Sub traffic classes that are attached to this traffic class.
Classifiers	Classifiers that are attached to this traffic class.
DARs	DAR objects that are attached to this traffic class.

Related Commands

add sqos policy trafficclass
add sqos trafficclass classifier
add sqos trafficclass dar
add sqos trafficclass subclass
create sqos trafficclass
delete sqos policy trafficclass
delete sqos trafficclass classifier
delete sqos trafficclass dar
delete sqos trafficclass subclass
destroy sqos trafficclass
set sqos trafficclass