QoS: Congestion Management Configuration Guide, Cisco IOS XE Release 2

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

© 2011 Cisco Systems, Inc. All rights reserved.

CONTENTS

Congestion Management Overview

Congestion management features allow you to control congestion by determining the order in which packets are sent out an interface based on priorities assigned to those packets. Congestion management entails the creation of queues, assignment of packets to those queues based on the classification of the packet, and scheduling of the packets in a queue for transmission. The congestion management QoS feature offers four types of queueing protocols, each of which allows you to specify creation of a different number of queues, affording greater or lesser degrees of differentiation of traffic, and to specify the order in which that traffic is sent.

During periods with light traffic, that is, when no congestion exists, packets are sent out the interface as soon as they arrive. During periods of transmit congestion at the outgoing interface, packets arrive faster than the interface can send them. If you use congestion management features, packets accumulating at an interface are queued until the interface is free to send them; they are then scheduled for transmission according to their assigned priority and the queueing mechanism configured for the interface. The router determines the order of packet transmission by controlling which packets are placed in which queue and how queues are serviced with respect to each other.

This module discusses the types of queueing and queueing-related features (such as bandwidth management) which constitute the congestion management QoS features:

- Weighted fair queueing (WFQ). Also known as flow-based WFQ in this module.

WFQ offers dynamic, fair queueing that divides bandwidth across queues of traffic based on weights. (WFQ ensures that all traffic is treated fairly, given its weight.) To understand how WFQ works, consider the queue for a series of File Transfer Protocol (FTP) packets as a queue for the collective and the queue for discrete interactive traffic packets as a queue for the individual. Given the weight of the queues, WFQ ensures that for all FTP packets sent as a collective an equal number of individual interactive traffic packets are sent.)

Given this handling, WFQ ensures satisfactory response time to critical applications, such as interactive, transaction-based applications, that are intolerant of performance degradation. For serial interfaces at E1 (2.048 Mbps) and below, flow-based WFQ is used by default.

- Class-based WFQ (CBWFQ)

CBWFQ extends the standard WFQ functionality to provide support for user-defined traffic classes. For CBWFQ, you define traffic classes based on match criteria including protocols, access control lists (ACLs), and input interfaces. Packets satisfying the match criteria for a class constitute the traffic for that class.

- Priority queueing (PQ). With PQ, packets belonging to one priority class of traffic are sent before all lower priority traffic to ensure timely delivery of those packets.

Note You can assign only one queueing mechanism type to an interface.

Note A variety of queueing mechanisms can be configured using multilink, for example, Multichassis Multilink PPP (MMP). However, if only PPP is used on a tunneled interface--for example, virtual private dialup network (VPND), PPP over Ethernet (PPPoE)--no queueing can be configured on the virtual interface.

- Bandwidth Management

CBWFQ and LLQ (as well as other QoS functionality) can al reserve and consume bandwidth, up to a maximum of the reserved bandwidth on an interface. Specific commands can be used to allocate and fineptune bandwidth as needed. For more information, see the Bandwidth Management, page 12.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Why Use Congestion Management

Heterogeneous networks include many different protocols used by applications, giving rise to the need to prioritize traffic in order to satisfy time-critical applications while still addressing the needs of less time-dependent applications, such as file transfer. Different types of traffic sharing a data path through the network can interact with one another in ways that affect their application performance. If your network is designed to support different traffic types that share a single data path between routers, you should consider using congestion management techniques to ensure fairness of treatment across the various traffic types.

Here are some broad factors to consider in determining whether to configure congestion management QoS:

- Traffic prioritization is especially important for delay-sensitive, interactive transaction-based applications--for instance, desktop video conferencing--that require higher priority than do file transfer applications. However, use of WFQ ensures that all traffic is treated fairly, given its weight, and in a dynamic manner. For example, WFQ addresses the requirements of the interactive application without penalizing the FTP application.

- Prioritization is most effective on WAN links where the combination of bursty traffic and relatively lower data rates can cause temporary congestion.
- Depending on the average packet size, prioritization is most effective when applied to links at T1/E1 bandwidth speeds or lower.
- If users of applications running across your network notice poor response time, you should consider using congestion management features. Congestion management features are dynamic, tailoring themselves to the existing network conditions. However, consider that if a WAN link is constantly congested, traffic prioritization may not resolve the problem. Adding bandwidth might be the appropriate solution.
- If there is no congestion on the WAN link, there is no reason to implement traffic prioritization.

The following list summarizes aspects you should consider in determining whether you should establish and implement a queueing policy for your network:

- Determine if the WAN is congested--that is, whether users of certain applications perceive a performance degradation.
- Determine your goals and objectives based on the mix of traffic you need to manage and your network topology and design. In identifying what you want to achieve, consider whether your goal is among the following:
    - To establish fair distribution of bandwidth allocation across all of the types of traffic you identify.
    - To grant strict priority to traffic from special kinds of applications you service--for example, interactive multimedia applications--possibly at the expense of less-critical traffic you also support.
    - To customize bandwidth allocation so that network resources are shared among all of the applications you service, each having the specific bandwidth requirements you have identified.
    - To effectively configure queueing. You must analyze the types of traffic using the interface and determine how to distinguish them. See the "Classification Overview" module for a description of how packets are classified.

After you assess your needs, review the available congestion management queueing mechanisms described in this module and determine which approach best addresses your requirements and goals.

- Configure the interface for the kind of queueing strategy you have chosen, and observe the results.

Traffic patterns change over time, so you should repeat the analysis process described in the second bullet periodically, and adapt the queueing configuration accordingly.

See the following section Deciding Which Queueing Policy to Use, page 3 for more information about the various queueing mechanisms.

Deciding Which Queueing Policy to Use

When deciding which queueing policy to use, note the following points:

- PQ guarantees strict priority in that it ensures that one type of traffic will be sent, possibly at the expense of all others. For PQ, a low priority queue can be detrimentally affected, and, in the worst case, never allowed to send its packets if a limited amount of bandwidth is available or if the transmission rate of critical traffic is high.
- WFQ does not require configuration of access lists to determine the preferred traffic on a serial interface. Rather, the fair queue algorithm dynamically sorts traffic into messages that are part of a conversation.

- Low-volume, interactive traffic gets fair allocation of bandwidth with WFQ, as does high-volume traffic such as file transfers.
- Strict priority queueing can be accomplished with WFQ by using low latency queueing (LLQ). Strict PQ allows delay-sensitive data such as voice to be dequeued and sent before packets in other queues are dequeued.

The table below compares the salient features of flow-based WFQ, CBWFQ, and PQ.

Table 1 Queueing Comparison

	WFQ	CBWFQ/	PQ
Number of Queues	Configurable number of queues (256 user queues, by default)	One queue per class, up to 64 classes	4 queues
Kind of Service	• Ensures fairness among all traffic flows based on weights	• Provides class bandwidth guarantee for user-defined traffic classes • Provides flow-based WFQ support for nonuser-defined traffic classes • Strict priority queueing is available through use of the LLQ.	• High priority queues are serviced first

Weighted Fair Queueing

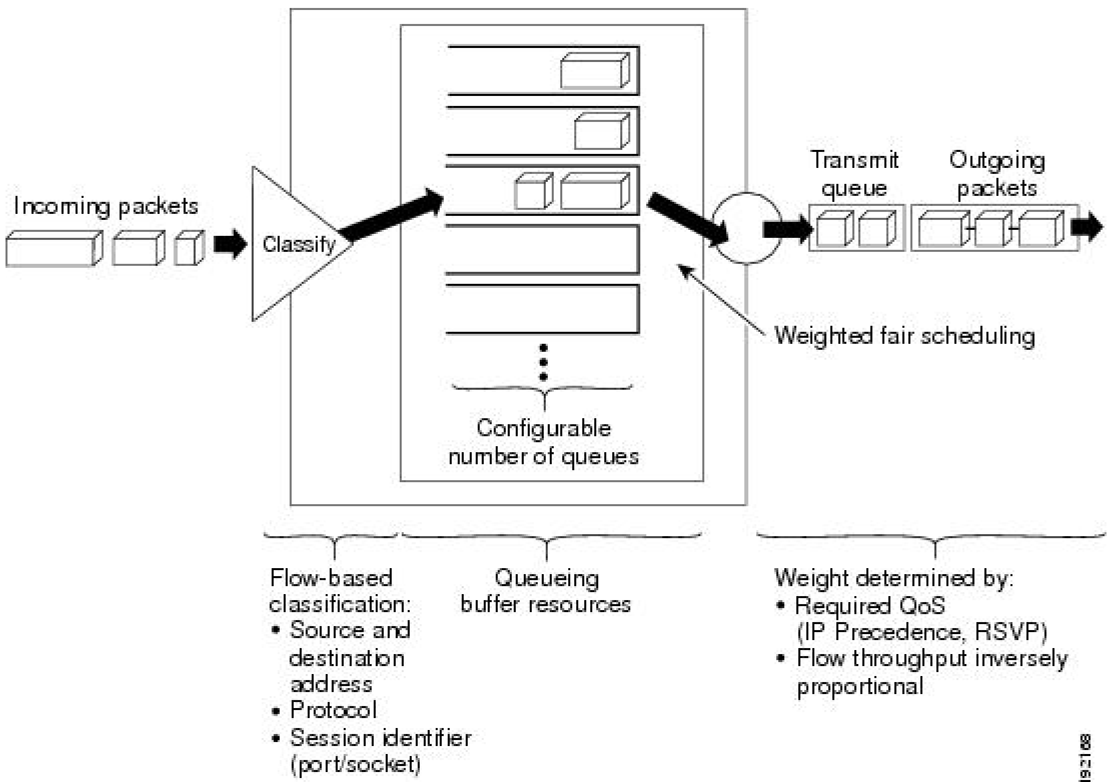
This section contains overview information about WFQ (often referred to as flow-based WFQ).

WFQ Functionality

WFQ is a dynamic scheduling method that provides fair bandwidth allocation to all network traffic. WFQ applies priority, or weights, to identified traffic to classify traffic into conversations and determine how much bandwidth each conversation is allowed relative to other conversations. WFQ is a flow-based algorithm that simultaneously schedules interactive traffic to the front of a queue to reduce response time and fairly shares the remaining bandwidth among high-bandwidth flows. In other words, WFQ allows you to give low-volume traffic, such as Telnet sessions, priority over high-volume traffic, such as FTP sessions.

WFQ gives concurrent file transfers balanced use of link capacity; that is, when multiple file transfers occur, the transfers are given comparable bandwidth. The figure below shows how WFQ works.

Figure 1 Weighted Fair Queueing



WFQ provides traffic priority management that dynamically sorts traffic into messages that make up a conversation. WFQ breaks up the train of packets within a conversation to ensure that bandwidth is shared fairly between individual conversations and that low-volume traffic is transferred in a timely fashion.

WFQ classifies traffic into different flows based on packet header addressing, including such characteristics as source and destination network or MAC address, protocol, source and destination port and socket numbers of the session, Frame Relay data-link connection identifier (DLCI) value, and ToS value. There are two categories of flows: high-bandwidth sessions and low-bandwidth sessions. Low-bandwidth traffic has effective priority over high-bandwidth traffic, and high-bandwidth traffic shares the transmission service proportionally according to assigned weights. Low-bandwidth traffic streams, which comprise the majority of traffic, receive preferential service, allowing their entire offered loads to be sent in a timely fashion. High-volume traffic streams share the remaining capacity proportionally among themselves.

WFQ places packets of the various conversations in the fair queues before transmission. The order of removal from the fair queues is determined by the virtual time of the delivery of the last bit of each arriving packet.

New messages for high-bandwidth flows are discarded after the congestive-messages threshold has been met. However, low-bandwidth flows, which include control-message conversations, continue to enqueue data. As a result, the fair queue may occasionally contain more messages than are specified by the threshold number.

WFQ can manage duplex data streams, such as those between pairs of applications, and simplex data streams such as voice or video.

The WFQ algorithm also addresses the problem of round-trip delay variability. If multiple high-volume conversations are active, their transfer rates and interarrival periods are made much more predictable. WFQ greatly enhances algorithms such as Systems Network Architecture (SNA) Logical Link Control (LLC) and TCP congestion control and slow start features.

WFQ is used as the default queueing mode on most serial interfaces configured to run at E1 speeds (2.048 Mbps) or below.

WFQ provides the solution for situations in which it is desirable to provide consistent response time to heavy and light network users alike without adding excessive bandwidth. WFQ automatically adapts to changing network traffic conditions.

Restrictions

WFQ is not supported with tunneling and encryption because these features modify the packet content information required by WFQ for classification.

Although WFQ automatically adapts to changing network traffic conditions, it does not offer the degree of precision control over bandwidth allocation that CQ and CBWFQ offer.

WFQ and IP Precedence

WFQ is IP precedence-aware. It can detect higher priority packets marked with precedence by the IP Forwarder and can schedule them faster, providing superior response time for this traffic. Thus, as the precedence increases, WFQ allocates more bandwidth to the conversation during periods of congestion.

WFQ assigns a weight to each flow, which determines the transmit order for queued packets. In this scheme, lower weights are served first. For standard Cisco IOS WFQ, the IP precedence serves as a divisor to this weighting factor.

Like CQ, WFQ sends a certain number of bytes from each queue. With WFQ, each queue corresponds to a different flow. For each cycle through all flows, WFQ effectively sends a number of bytes equal to the precedence of the flow plus one. This number is only used as a ratio to determine how many bytes per packets to send. However, for the purposes of understanding WFQ, using this number as the byte count is sufficient. For instance, traffic with an IP Precedence value of 7 gets a lower weight than traffic with an IP Precedence value of 3, thus, the priority in transmit order. The weights are inversely proportional to the IP Precedence value.

To determine the bandwidth allocation for each queue, divide the byte count for the flow by the total byte count for all flows. For example, if you have one flow at each precedence level, each flow will get precedence + 1 parts of the link:

1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 = 36

Thus, precedence 0 traffic will get 1/36 of the bandwidth, precedence 1 traffic will get 2/36, and precedence 7 traffic will get 8/36.

However, if you have 18 precedence 1 flows and one of each of the rest, the total is now:

1 + 2(18) + 3 + 4 + 5 + 6 + 7 + 8 = 70

Precedence 0 traffic will get 1/70, each of the precedence 1 flows will get 2/70, and so on.

As flows are added or ended, the actual allocated bandwidth will continuously change.

WFQ and RSVP

RSVP uses WFQ to allocate buffer space and schedule packets, and to guarantee bandwidth for reserved flows. WFQ works with RSVP to help provide differentiated and guaranteed QoS services.

RSVP is the Internet Engineering Task Force (IETF) Internet Standard (RFC 2205) protocol for allowing an application to dynamically reserve network bandwidth. RSVP enables applications to request a specific QoS for a data flow. The Cisco implementation allows RSVP to be initiated within the network using configured proxy RSVP.

RSVP is the only standard signalling protocol designed to guarantee network bandwidth from end to end for IP networks. Hosts and routers use RSVP to deliver QoS requests to the routers along the paths of the data stream and to maintain router and host state to provide the requested service, usually bandwidth and latency. RSVP uses a mean data rate, the largest amount of data the router will keep in queue, and minimum QoS to determine bandwidth reservation.

WFQ or Weighted Random Early Detection (WRED) acts as the preparer for RSVP, setting up the packet classification and scheduling required for the reserved flows. Using WFQ, RSVP can deliver an Integrated Services Guaranteed Service.

Class-Based Weighted Fair Queueing

CBWFQ extends the standard WFQ functionality to provide support for user-defined traffic classes. For CBWFQ, you define traffic classes based on match criteria including protocols, access control lists (ACLs), and input interfaces. Packets satisfying the match criteria for a class constitute the traffic for that class.

Once a class has been defined according to its match criteria, you can assign it characteristics. To characterize a class, you assign it bandwidth, weight, and maximum packet limit. The bandwidth assigned to a class is the guaranteed bandwidth delivered to the class during congestion.

To characterize a class, you also specify the queue limit for that class, which is the maximum number of packets allowed to accumulate in the queue for the class. Packets belonging to a class are subject to the bandwidth and queue limits that characterize the class.

After a queue has reached its configured queue limit, enqueueing of additional packets to the class causes tail drop or packet drop to take effect, depending on how class policy is configured.

Tail drop is used for CBWFQ classes unless you explicitly configure policy for a class to use WRED to drop packets as a means of avoiding congestion. Note that if you use WRED packet drop instead of tail drop for one or more classes comprising a policy map, you must ensure that WRED is not configured for the interface to which you attach that service policy.

If a default class is configured with the **bandwidth** policy-map class configuration command, all unclassified traffic is put into a single queue and given treatment according to the configured bandwidth. If a default class is configured with the **fair-queue** command, all unclassified traffic is flow classified and given best-effort treatment. If no default class is configured, then by default the traffic that does not match any of the configured classes is flow classified and given best-effort treatment. Once a packet is classified, all of the standard mechanisms that can be used to differentiate service among the classes apply.

Flow classification is standard WFQ treatment. That is, packets with the same source IP address, destination IP address, source TCP or UDP port, or destination TCP or UDP port are classified as belonging to the same flow. WFQ allocates an equal share of bandwidth to each flow. Flow-based WFQ is also called fair queueing because all flows are equally weighted.

For CBWFQ, the weight specified for the class becomes the weight of each packet that meets the match criteria of the class. Packets that arrive at the output interface are classified according to the match criteria filters you define, then each one is assigned the appropriate weight. The weight for a packet belonging to a specific class is derived from the bandwidth you assigned to the class when you configured it; in this sense the weight for a class is user-configurable.

After the weight for a packet is assigned, the packet is enqueued in the appropriate class queue. CBWFQ uses the weights assigned to the queued packets to ensure that the class queue is serviced fairly.

Configuring a class policy--thus, configuring CBWFQ--entails these three processes:

- Defining traffic classes to specify the classification policy (class maps).

This process determines how many types of packets are to be differentiated from one another.

- Associating policies--that is, class characteristics--with each traffic class (policy maps).

This process entails configuration of policies to be applied to packets belonging to one of the classes previously defined through a class map. For this process, you configure a policy map that specifies the policy for each traffic class.

- Attaching policies to interfaces (service policies).

This process requires that you associate an existing policy map, or service policy, with an interface to apply the particular set of policies for the map to that interface.

CBWFQ Bandwidth Allocation

The sum of all bandwidth allocation on an interface cannot exceed 75 percent of the total available interface bandwidth. The remaining 25 percent is used for other overhead, including Layer 2 overhead, routing traffic, and best-effort traffic. Bandwidth for the CBWFQ class-default class, for instance, is taken from the remaining 25 percent. However, under aggressive circumstances in which you want to configure more than 75 percent of the interface bandwidth to classes, you can override the 75 percent maximum sum allocated to all classes or flows using the **max-reserved-bandwidth** command. If you want to override the default 75 percent, exercise caution and ensure that you allow enough remaining bandwidth to support best-effort and control traffic, and Layer 2 overhead.

Why Use CBWFQ?

Here are some general factors you should consider in determining whether you need to configure CBWFQ:

- Bandwidth allocation. CBWFQ allows you to specify the exact amount of bandwidth to be allocated for a specific class of traffic. Taking into account available bandwidth on the interface, you can configure up to 64 classes and control distribution among them, which is not the case with flow-based WFQ. Flow-based WFQ applies weights to traffic to classify it into conversations and determine how much bandwidth each conversation is allowed relative to other conversations. For flow-based WFQ, these weights, and traffic classification, are dependent on and limited to the seven IP Precedence levels.
- Coarser granularity and scalability. CBWFQ allows you to define what constitutes a class based on criteria that exceed the confines of flow. CBWFQ allows you to use ACLs and protocols or input interface names to define how traffic will be classified, thereby providing coarser granularity. You need not maintain traffic classification on a flow basis. Moreover, you can configure up to 64 discrete classes in a service policy.

CBWFQ and RSVP

RSVP can be used in conjunction with CBWFQ. When both RSVP and CBWFQ are configured for an interface, RSVP and CBWFQ act independently, exhibiting the same behavior that they would if each were running alone. RSVP continues to work as it does when CBWFQ is not present, even in regard to bandwidth availability assessment and allocation.

Restrictions

Configuring CBWFQ on a physical interface is only possible if the interface is in the default queueing mode. Serial interfaces at E1 (2.048 Mbps) and below use WFQ by default--other interfaces use FIFO by default. Enabling CBWFQ on a physical interface overrides the default interface queueing method.

If you configure a class in a policy map to use WRED for packet drop instead of tail drop, you must ensure that WRED is not configured on the interface to which you intend to attach that service policy.

Low Latency Queueing

The LLQ feature brings strict PQ to CBWFQ. Strict PQ allows delay-sensitive data such as voice to be dequeued and sent before packets in other queues are dequeued.

Without LLQ, CBWFQ provides WFQ based on defined classes with no strict priority queue available for real-time traffic. CBWFQ allows you to define traffic classes and then assign characteristics to that class. For example, you can designate the minimum bandwidth delivered to the class during congestion.

For CBWFQ, the weight for a packet belonging to a specific class is derived from the bandwidth you assigned to the class when you configured it. Therefore, the bandwidth assigned to the packets of a class determines the order in which packets are sent. All packets are serviced fairly based on weight; no class of packets may be granted strict priority. This scheme poses problems for voice traffic that is largely intolerant of delay, especially variation in delay. For voice traffic, variations in delay introduce irregularities of transmission manifesting as jitter in the heard conversation.

LLQ provides strict priority queueing for CBWFQ, reducing jitter in voice conversations. Configured by the **priority** command, LLQ enables use of a single, strict priority queue within CBWFQ at the class level, allowing you to direct traffic belonging to a class to the CBWFQ strict priority queue. To enqueue class traffic to the strict priority queue, you specify the named class within a policy map and then configure the **priority** command for the class. (Classes to which the **priority** command is applied are considered priority classes.) Within a policy map, you can give one or more classes priority status. When multiple classes within a single policy map are configured as priority classes, all traffic from these classes is enqueued to the same, single, strict priority queue.

One of the ways in which the strict PQ used within CBWFQ differs from its use outside CBWFQ is in the parameters it takes. Outside CBWFQ, you can use the **ip rtp priority** command to specify the range of UDP ports whose voice traffic flows are to be given priority service. Using the **priority** command, you are no longer limited to a UDP port number to stipulate priority flows because you can configure the priority status for a class within CBWFQ. Instead, all of the valid match criteria used to specify traffic for a class now apply to priority traffic. These methods of specifying traffic for a class include matching on access lists, protocols, and input interfaces. Moreover, within an access list you can specify that traffic matches are allowed based on the IP differentiated services code point (DSCP) value that is set using the first six bits of the ToS byte in the IP header.

Although it is possible to enqueue various types of real-time traffic to the strict priority queue, we strongly recommend that you direct only voice traffic to it because voice traffic is well-behaved, whereas other types of real-time traffic are not. Moreover, voice traffic requires that delay be nonvariable in order to avoid jitter. Real-time traffic such as video could introduce variation in delay, thereby thwarting the steadiness of delay required for successful voice traffic transmission.

For information on how to configure LLQ, see the "Configuring Weighted Fair Queueing" module.

LLQ Bandwidth Allocation

When you specify the **priority** command for a class, it takes a *bandwidth* argument that gives maximum bandwidth in kbps. You use this parameter to specify the maximum amount of bandwidth allocated for packets belonging to the class configured with the **priority** command. The bandwidth parameter both guarantees bandwidth to the priority class and restrains the flow of packets from the priority class.

In the event of congestion, policing is used to drop packets when the bandwidth is exceeded. Voice traffic enqueued to the priority queue is UDP-based and therefore not adaptive to the early packet drop characteristic of WRED. Because WRED is ineffective, you cannot use the WRED **random-detect** command with the **priority** command. In addition, because policing is used to drop packets and a queue limit is not imposed, the **queue-limit** command cannot be used with the **priority** command.

When congestion occurs, traffic destined for the priority queue is metered to ensure that the bandwidth allocation configured for the class to which the traffic belongs is not exceeded.

Priority traffic metering has the following qualities:

- Priority traffic metering is only performed under congestion conditions. When the device is not congested, the priority class traffic is allowed to exceed its allocated bandwidth. When the device is congested, the priority class traffic above the allocated bandwidth is discarded.
- It is performed on a per-packet basis, and tokens are replenished as packets are sent. If not enough tokens are available to send the packet, it is dropped.
- It restrains priority traffic to its allocated bandwidth to ensure that nonpriority traffic, such as routing packets and other data, is not starved.

With metering, the classes are policed and rate-limited individually. That is, although a single policy map might contain four priority classes, all of which are enqueued in a single priority queue, they are each treated as separate flows with separate bandwidth allocations and constraints.

It is important to note that because bandwidth for the priority class is specified as a parameter to the **priority** command, you cannot also configure the **bandwidth** policy-map class configuration command for a priority class. To do so is a configuration violation that would only introduce confusion in relation to the amount of bandwidth to allocate.

The bandwidth allocated for a priority queue always includes the Layer 2 encapsulation header. However, it does not include other headers. When you calculate the amount of bandwidth to allocate for a given priority class, you must account for the fact that Layer 2 headers are included. You must also allow bandwidth for the possibility of jitter introduced by routers in the voice path.

Note The sum of all bandwidth allocation on an interface cannot exceed 75 percent of the total available interface bandwidth. However, under aggressive circumstances in which you want to configure more than 75 percent of the interface bandwidth to classes, you can override the 75 percent maximum sum allocated to all classes or flows using the **max-reserved-bandwidth** command. The **max-reserved-bandwidth** command is intended for use on main interfaces only.

Why Use LLQ?

Here are some general factors you should consider in determining whether you need to configure LLQ:

- LLQ provides strict priority service serial interfaces.
- LLQ is not limited to UDP port numbers. Because you can configure the priority status for a class within CBWFQ, you are no longer limited to UDP port numbers to stipulate priority flows. Instead, all of the valid match criteria used to specify traffic for a class now apply to priority traffic.
- By configuring the maximum amount of bandwidth allocated for packets belonging to a class, you can avoid starving nonpriority traffic.

Restrictions

The following restrictions apply to LLQ:

- The **random-detect** command, **queue-limit** command, and **bandwidth** policy-map class configuration command cannot be used while the **priority** command is configured.
- The **priority** command can be configured in multiple classes, but it should only be used for voice-like, constant bit rate (CBR) traffic.

Priority Queueing

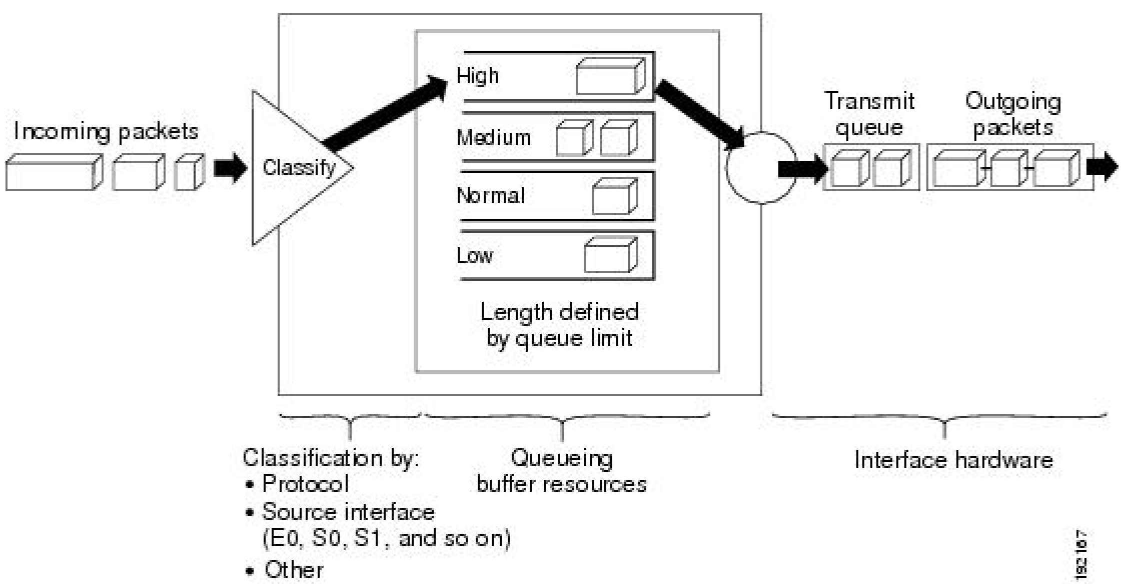
PQ allows you to define how traffic is prioritized in the network. You configure four traffic priorities. You can define a series of filters based on packet characteristics to cause the router to place traffic into these four queues; the queue with the highest priority is serviced first until it is empty, then the lower queues are serviced in sequence.

For information on how to configure PQ, see the "Configuring Priority Queueing" module.

How It Works

During transmission, PQ gives priority queues absolute preferential treatment over low priority queues; important traffic, given the highest priority, always takes precedence over less important traffic. Packets are classified based on user-specified criteria and placed into one of the four output queues--high, medium, normal, and low--based on the assigned priority. Packets that are not classified by priority fall into the normal queue. The figure below illustrates this process.

Figure 2 Priority Queueing



When a packet is to be sent out an interface, the priority queues on that interface are scanned for packets in descending order of priority. The high priority queue is scanned first, then the medium priority queue, and so on. The packet at the head of the highest queue is chosen for transmission. This procedure is repeated every time a packet is to be sent.

The maximum length of a queue is defined by the length limit. When a queue is longer than the queue limit, all additional packets are dropped.

Note The priority output queueing mechanism can be used to manage traffic from all networking protocols. Additional fine-tuning is available for IP and for setting boundaries on the packet size.

How Packets Are Classified for Priority Queueing

A priority list is a set of rules that describe how packets should be assigned to priority queues. A priority list might also describe a default priority or the queue size limits of the various priority queues.

Packets can be classified by the following criteria:

- Protocol or subprotocol type
- Incoming interface
- Packet size
- Fragments
- Access list

Keepalives sourced by the network server are always assigned to the high priority queue; all other management traffic (such as Interior Gateway Routing Protocol (IGRP) updates) must be configured. Packets that are not classified by the priority list mechanism are assigned to the normal queue.

Why Use Priority Queueing?

PQ provides absolute preferential treatment to high priority traffic, ensuring that mission-critical traffic traversing various WAN links gets priority treatment. In addition, PQ provides a faster response time than do other methods of queueing.

Although you can enable priority output queueing for any interface, it is best used for low-bandwidth, congested serial interfaces.

Restrictions

When choosing to use PQ, consider that because lower priority traffic is often denied bandwidth in favor of higher priority traffic, use of PQ could, in the worst case, result in lower priority traffic never being sent. To avoid inflicting these conditions on lower priority traffic, you can use traffic shaping to rate-limit the higher priority traffic.

PQ introduces extra overhead that is acceptable for slow interfaces, but may not be acceptable for higher speed interfaces such as Ethernet. With PQ enabled, the system takes longer to switch packets because the packets are classified by the processor card.

PQ uses a static configuration and does not adapt to changing network conditions.

PQ is not supported on any tunnels.

Bandwidth Management

RSVP, CBWFQ and LLQ can all reserve and consume bandwidth, up to a maximum of the reserved bandwidth on an interface.

To allocate bandwidth, you can use one of the following commands:

- For RSVP, use the **ip rsvp bandwidth** command.
- For CBWFQ, use the **bandwidth** policy-map class configuration command.
- For LLQ, you can allocate bandwidth using the **priority** command.

When you configure these commands, be aware of bandwidth limitations and configure bandwidth according to requirements in your network. Remember, the sum of all bandwidths cannot exceed the maximum reserved bandwidth. The default maximum bandwidth is 75 percent of the total available bandwidth on the interface. The remaining 25 percent of bandwidth is used for overhead, including Layer 2 overhead, routing traffic, and best-effort traffic.

If you find that it is necessary to change the maximum reserved bandwidth, you can change the maximum bandwidth by using the **max-reserved-bandwidth** command. The **max-reserved-bandwidth** command can be used only on interfaces; it cannot be used on VCs.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Low Latency Queueing with Priority Percentage Support

This feature allows you to configure bandwidth as a percentage within low latency queueing (LLQ). Specifically, you can designate a percentage of the bandwidth to be allocated to an entity (such as a physical interface, a shaped ATM permanent virtual circuit [PVC], or a shaped Frame Relay PVC to which a policy map is attached). Traffic associated with the policy map will then be given priority treatment.

This feature also allows you to specify the percentage of bandwidth to be allocated to nonpriority traffic classes. It modifies two existing commands--**bandwidth** and **priority**--and provides additional functionality to the way that bandwidth can be allocated using these two commands.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for LLQ with Priority Percentage Support

Dropping Excess Traffic

If the incoming high priority traffic exceeds the bandwidth percentage calculated by the **priority percent**command, and there is congestion in the network, the excess traffic is dropped. This is identical to the behavior demonstrated when the **priority**command uses bandwidth in kbps. In both cases, if the high priority traffic exceeds the bandwidth, and there is congestion in the network, excess traffic is dropped.

Exceeding the Configured Bandwidth Percentage Calculated by the bandwidth percent and priority percent Commands

By default, when the **bandwidth percent**and **priority percent** commands are used to allocate bandwidth, the sum of the bandwidth percentage allocated to the high priority traffic and the bandwidth percentage allocated to the nonpriority traffic cannot exceed 99 percent of the total bandwidth available on the interface.

The remaining 1 percent of the total bandwidth available on the interface is kept in reserve for the unclassified traffic and routing traffic, if any, and is proportionally divided among the defined traffic classes.

Information About LLQ with Priority Percentage Support

Benefits of LLQ with Priority Percentage Support

This feature allows the Cisco IOS XE Software to accommodate networks with a large number of interfaces, all with differing bandwidths. This feature is useful when all of those interfaces with differing bandwidths need to be associated with a policy map that allocates proportional bandwidths to multiple classes.

Additionally, configuring bandwidth in percentages is most useful when the underlying link bandwidth is unknown or the relative class bandwidth distributions are known. For interfaces that have adaptive shaping rates (such as available bit rate [ABR] virtual circuits), CBWFQ can be configured by configuring class bandwidths in percentages.

Changes to the bandwidth Command for LLQ with Priority Percentage Support

This feature adds a new keyword to the **bandwidth** command--**remaining percent**. The feature also changes the functionality of the existing **percent** keyword. These changes result in the following commands for bandwidth: **bandwidth percent**and **bandwidth remaining percent**.

The **bandwidth percent** command configures bandwidth as an absolute percentage of the total bandwidth on the interface.

The **bandwidth remaining percent**command allows you to allocate bandwidth as a relative percentage of the total bandwidth available on the interface. This command allows you to specify the relative percentage of the bandwidth to be allocated to the classes of traffic. For instance, you can specify that 30 percent of the available bandwidth be allocated to class1, and 60 percent of the bandwidth be allocated to class2. Essentially, you are specifying the ratio of the bandwidth to be allocated to the traffic class. In this case, the ratio is 1 to 2 (30 percent allocated to class1 and 60 percent allocated to class2). The sum of the numbers used to indicate this ratio cannot exceed 100 percent. This way, you need not know the total amount of bandwidth available, just the relative percentage you want to allocate for each traffic class.

Each traffic class gets a minimum bandwidth as a relative percentage of the remaining bandwidth. The remaining bandwidth is the bandwidth available after the priority queue, if present, is given its required bandwidth, and after any Resource Reservation Protocol (RSVP) flows are given their requested bandwidth.

Because this is a relative bandwidth allocation, the packets for the traffic classes are given a proportionate weight only, and no admission control is performed to determine whether any bandwidth (in kbps) is actually available. The only error checking that is performed is to ensure that the total bandwidth percentages for the classes do not exceed 100 percent.

For more information about how this feature defines and calculates bandwidth, see the Bandwidth Calculations in LLQ with Priority Percentage Support, page 17 section of this document. For the **bandwidth**command syntax description and usage guidelines, see the Cisco IOS Quality of Service Solutions Command Reference.

Changes to the priority Command for LLQ with Priority Percentage Support

This feature also adds the **percent** keyword to the **priority** command. The **priority percent** command indicates that the bandwidth will be allocated as a percentage of the total bandwidth of the interface. You can then specify the percentage (that is, a number from 1 to 100) to be allocated by using the *percentage* argument with the **priority percent**command.

Unlike the **bandwidth** command, the **priority** command provides a strict priority to the traffic class, which ensures low latency to high priority traffic classes.

Bandwidth Calculations in LLQ with Priority Percentage Support

When the **bandwidth** and **priority** commands calculate the total amount of bandwidth available on an entity, the following guidelines are invoked:

- If the entity is a physical interface, the total bandwidth is the bandwidth on the physical interface.
- If the entity is a shaped ATM PVC, the total bandwidth is calculated as follows:
    - For a variable bit rate (VBR) VC, the average shaping rate is used in the calculation.
    - For an available bit rate (ABR) VC, the minimum shaping rate is used in the calculation.

How to Configure LLQ with Priority Percentage Support

Specifying the Bandwidth Percentage

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map*
4. **class** {*class-name***class-default**}
5. **priority** {*bandwidth-kbps* | **percent** *percentage*} [*burst*]
6. **bandwidth** {*bandwidth-kbps* | **percent** *percentage* | **remaining percent** *percentage*}
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map* **Example:** `Router(config)# policy-map policy1`	Specifies the name of the policy map to be created or modified. Enters policy-map configuration mode. • Enter the policy map name.
Step 4	**class** {*class-name***class-default**} **Example:** `Router(config-pmap)# class class1`	Specifies the class so that you can configure or modify its policy. Enters policy-map class configuration mode. • Enter the class name.

Step 5	**priority** {*bandwidth-kbps* | **percent** *percentage*} [*burst*]
	
	Example:
	
	`Router(config-pmap-c)#`
	`priority percent 10`
		Gives priority to a class of traffic belonging to the policy map.
		• Enter the priority percentage.

	Command or Action	Purpose

Step 6	**bandwidth** {*bandwidth-kbps* | **percent** *percentage* | **remaining percent** *percentage*}
	
	Example:
	
	`Router(config-pmap-c)#`
	
	`bandwidth percent 30`
		Specifies the bandwidth for a class of traffic belonging to the policy map.
		
		• Enter the bandwidth percentage.

Step 7	**end**
	
	Example:
	
	Example:
	
	`Router(config-pmap-c)#`
	
	`end`
		(Optional) Exits policy-map class configuration mode and returns to privileged EXEC mode.

Verifying the Bandwidth Percentage

SUMMARY STEPS

1. **enable**
2. **show policy-map** *policy-map*
3. **show policy-map** *policy-map* **class** *class-name*
4. **show policy-map interface** *type number*
5. **exit**

DETAILED STEPS

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

Step 2	**show policy-map** *policy-map*
	
	Example:
	
	`Router#`
	`show policy-map policy1`
		(Optional) Displays the configuration of all classes for a specified service policy map or the configuration of all classes for all existing policy maps
		
		• Enter the name of the policy map whose complete configuration is to be displayed.

	Command or Action	Purpose

Step 3	**show policy-map** *policy-map* **class** *class-name*
	
	Example:
	
	`Router#`
	`show policy-map policy1 class class1`
		(Optional) Displays the configuration for the specified class of the specified policy map.
		
		• Enter the policy map name and the class name.

Step 4	**show policy-map interface** *type number*
	
	Example:
	
	`Router#`
	`show policy-map interface serial4/0/0`
		(Optional) Displays the packet statistics of all classes that are configured for all service policies either on the specified interface or subinterface or on a specific PVC on the interface.
		
		• Enter the interface type and number.

Step 5	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

Configuration Examples for LLQ with Priority Percentage Support

Example Specifying the Bandwidth Percentage

The following example uses the **priority percent** command to specify a bandwidth percentage of 10 percent for the class called voice-percent. Then the **bandwidth remaining percent** command is used to specify a bandwidth percentage of 30 percent for the class called data1 and a bandwidth percentage of 20 percent for the class called data2.

```
Router> enable
Router# configure terminal
Router(config)# policy-map policy1
Router(config-pmap)# class voice-percent
Router(config-pmap-c)# priority percent 10
Router(config-pmap-c)# exit
Router(config-pmap)# class data1
Router(config-pmap-c)# bandwidth remaining percent 30
Router(config-pmap-c)# exit
Router(config-pmap)# class data2
Router(config-pmap-c)# bandwidth remaining percent 20
Router(config-pmap-c)# end
```

As a result of this configuration, 10 percent of the interface bandwidth is guaranteed for the class called voice-percent. The classes called data1 and data2 get 30 percent and 20 percent of the remaining bandwidth, respectively.

Example Mixing the Units of Bandwidth for Nonpriority Traffic

If a particular unit (that is, kbps or percentages) is used when specifying the bandwidth for a specific class of nonpriority traffic, the same bandwidth unit must be used when specifying the bandwidth for the other nonpriority classes in that policy map. The bandwidth units within the same policy map must be identical. However, the unit for the **priority** command in the priority class can be different from the bandwidth unit of the nonpriority class. The same configuration can contain multiple policy maps, however, which in turn can use different bandwidth units.

The following sample configuration contains three policy maps--policy1, policy2, and policy3. In the policy map called policy1 and the policy map called policy2, the bandwidth is specified by percentage. However, in the policy map called policy3, bandwidth is specified in kbps.

```
Router> enable
Router# configure terminal
Router(config)# policy-map policy1
Router(config-pmap)# class voice-percent
Router(config-pmap-c)# priority percent 10
Router(config-pmap-c)# exit
Router(config-pmap)# class data1
Router(config-pmap-c)# bandwidth percent 30
Router(config-pmap-c)# exit
Router(config-pmap)# class data2
Router(config-pmap-c)# bandwidth percent 20
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# policy-map policy2
Router(config-pmap)# class voice-percent
Router(config-pmap-c)# priority percent 10
Router(config-pmap-c)# exit
Router(config-pmap)# class data1
Router(config-pmap-c)# bandwidth remaining percent 30
Router(config-pmap-c)# exit
Router(config-pmap)# class data2
Router(config-pmap-c)# bandwidth remaining percent 20
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# policy-map policy3
Router(config-pmap)# class voice-percent
Router(config-pmap-c)# priority 500
Router(config-pmap-c)# exit
Router(config-pmap)# class data1
Router(config-pmap-c)# bandwidth 30
Router(config-pmap-c)# exit
Router(config-pmap)# class data2
Router(config-pmap-c)# bandwidth 20
Router(config-pmap-c)# end
```

Example Verifying the Bandwidth Percentage

The following sample output from the **show policy-map interface**command shows that 50 percent of the interface bandwidth is guaranteed for the class called class1 and that 25 percent is guaranteed for the class called class2. The output displays the amount of bandwidth as both a percentage and a number of kbps.

```
Router# show policy-map interface
 serial3/2/0
 Serial3/2/0
  Service-policy output:policy1
    Class-map:class1 (match-all)
```

```
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match:none
      Weighted Fair Queueing
        Output Queue:Conversation 265
        Bandwidth 50 (%)
        Bandwidth 772 (kbps) Max Threshold 64 (packets)
        (pkts matched/bytes matched) 0/0
        (depth/total drops/no-buffer drops) 0/0/0
Class-map:class2 (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match:none
      Weighted Fair Queueing
        Output Queue:Conversation 266
        Bandwidth 25 (%)
        Bandwidth 386 (kbps) Max Threshold 64 (packets)
        (pkts matched/bytes matched) 0/0
        (depth/total drops/no-buffer drops) 0/0/0
    Class-map:class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match:any
```

In this example, serial interface s3/2/0 has a total bandwidth of 1544 kbps. During periods of congestion, 50 percent (or 772 kbps) of the link bandwidth is guaranteed to the class called class1, and 25 percent (or 386 kbps) of the link bandwidth is guaranteed to the class called class2.

Additional References

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
LLQ	"Applying QoS Features Using the MQC" module

Standards

Standards	Title
No new or modified standards are supported, and support for existing standards has not been modified.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE Software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported, and support for existing RFCs has not been modified.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for LLQ with Priority Percentage Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 2 Feature Information for Low Latency Queueing with Priority Percentage Support

Feature Name	Releases	Feature Information
Low Latency Queueing with Priority Percentage Support	Cisco IOS XE Release 2.1	This feature allows you to configure bandwidth as a percentage within low latency queueing (LLQ). Specifically, you can designate a percentage of the bandwidth to be allocated to an entity (such as a physical interface, a shaped ATM permanent virtual circuit [PVC], or a shaped Frame Relay PVC to which a policy map is attached). Traffic associated with the policy map will then be given priority treatment. This feature was implemented on the Cisco ASR 1000 Series Routers. The following commands were introduced or modified: **bandwidth**(policy-map class), **priority**.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Low Latency Queueing for IPsec Encryption Engines

This feature module describes the LLQ for IPsec encryption engines feature and includes the following sections:

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Feature Overview

LLQ for IPsec encryption engines helps reduce packet latency by introducing the concept of queueing before crypto engines. Prior to this, the crypto processing engine gave data traffic and voice traffic equal status. Administrators now designate voice traffic as priority. Data packets arriving at a router interface are directed into a data packet inbound queue for crypto engine processing. This queue is called the best effort queue. Voice packets arriving on a router interface are directed into a priority packet inbound queue for crypto engine processing. This queue is called the priority queue. The crypto engine undertakes packet processing in a favorable ratio for voice packets. Voice packets are guaranteed a minimum processing bandwidth on the crypto engine.

Benefits of the LLQ for IPSec Encryption Engines

The LLQ for IPsec encryption engines feature guarantees a certain level of crypto engine processing time for priority designated traffic.

Better Voice Performance

Voice packets can be identified as priority, allowing the crypto engine to guarantee a certain percentage of processing bandwidth. This feature impacts the end user experience by assuring voice quality if voice traffic is directed onto a congested network.

Improved Latency and Jitters

Predictability is a critical component of network performance. The LLQ for IPsec encryption engines feature delivers network traffic predictability relating to VPN. With this feature disabled, an end user employing an IP phone over VPN might experience jitter or latency, both symptoms of overall network latency and congestion. With this feature enabled, these undesirable characteristics are dissipated.

Restrictions

- No per-tunnel QoS policy. An interface QoS policy represents all tunnels.
- Assume the same IP precedence/DSCP marking for inbound and outbound voice packets.
- Assume that the IP precedence/DSCP marking for voice packets is done at the source.
- Limited match criteria for voice traffic in the interface QoS policy.
- Assume that call admission control is enforced within the enterprise.
- No strict error checking when aggregate policy's bandwidth exceeds crypto engine bandwidth. Only a warning is displayed, but configuration is allowed.
- Assume that voice packets are either all encrypted or unencrypted.

Related Documents

- Cisco IOS Quality of Service Solutions Command Reference
- "Applying QoS Features Using the MQC" module

Supported Standards MIBs and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

No new or modified standards are supported by this feature.

To locate and download MIBs for selected platforms, Cisco IOS XE Software releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://www.cisco.com/go/mibs

RFCs

No new or modified RFCs are supported by this feature.

Prerequisites

To use this feature, you should be familiar with the following:

- Access control lists
- Bandwidth management
- CBWFQ

Configuration Tasks

To configure LLQ for IPsec encryption engines, perform the tasks described in the following sections.

- Defining Class Maps, page 27
- Configuring Class Policy in the Policy Map, page 28
- Configuring Class Policy for a Priority Queue, page 29
- Configuring Class Policy Using a Specified Bandwidth, page 29
- Configuring the Class-Default Class Policy, page 30
- Attaching the Service Policy, page 31
- Verifying Configuration of Policy Maps and Their Classes, page 31

Defining Class Maps

SUMMARY STEPS

1. Router(config)# class-map *class-map-name*
2. Do one of the following:
   - Router(config-cmap)# match access-group {access-group | name access-group-name}
   -
   -
   -

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# class-map *class-map-name*	Specifies the name of the class map to be created.

	Command or Action	Purpose

Step 2	Do one of the following:
	• Router(config-cmap)# match access-group {access-group | name access-group-name}
	•
	•
	•
	
	Example:
	`Router(config-cmap)# match input-interface interface-name`
	
	Example:
	
	Example:
	`or`
	
	Example:
	
	Example:
	`Router(config-cmap)# match protocol protocol`
		Specifies the name of the access control list (ACL) against whose contents packets are checked to determine if they belong to the class. Specifies the name of the input interface used as a match criterion against which packets are checked to determine if they belong to the class. Specifies the name of the protocol used as a match criterion against which packets are checked to determine if they belong to the class.

Configuring Class Policy in the Policy Map

To configure a policy map and create class policies that make up the service policy, begin with the policy-map command to specify the policy map name. Then use one or more of the following commands to configure the policy for a standard class or the default class:

- priority
- bandwidth
- queue-limit or random-detect
- fair-queue (for class-default class only)

For each class that you define, you can use one or more of the commands listed to configure the class policy. For example, you might specify bandwidth for one class and both bandwidth and queue limit for another class.

The default class of the policy map (commonly known as the class-default class) is the class to which traffic is directed if that traffic does not satisfy the match criteria of the other classes defined in the policy map.

You can configure class policies for as many classes as are defined on the router, up to the maximum of 64. However, the total amount of bandwidth allocated for all classes in a policy map must not exceed the minimum committed information rate (CIR) configured for the virtual circuit (VC) minus any bandwidth reserved by the frame-relay voice bandwidth and frame-relay ip rtp priority commands. If the minimum CIR is not configured, the bandwidth defaults to one half of the CIR. If all of the bandwidth is not allocated, the remaining bandwidth is allocated proportionally among the classes on the basis of their configured bandwidth.

Configuring Class Policy for a Priority Queue

SUMMARY STEPS

1. Router(config)# policy-map *policy-map*
2. Router(config-cmap)# class *class-name*
3. Router(config-pmap-c)# priority *bandwidth-kbps*

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# policy-map *policy-map*	Specifies the name of the policy map to be created or modified.
Step 2	Router(config-cmap)# class *class-name*	Specifies the name of a class to be created and included in the service policy.
Step 3	Router(config-pmap-c)# priority *bandwidth-kbps*	Creates a strict priority class and specifies the amount of bandwidth, in kbps, to be assigned to the class.

Configuring Class Policy Using a Specified Bandwidth

SUMMARY STEPS

1. Router(config)# policy-map *policy-map*
2. Router(config-cmap)# class *class-name*
3. Router(config-pmap-c)# bandwidth *bandwidth-kbps*

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# policy-map *policy-map*	Specifies the name of the policy map to be created or modified.
Step 2	Router(config-cmap)# class *class-name*	Specifies the name of a class to be created and included in the service policy.
Step 3	Router(config-pmap-c)# bandwidth *bandwidth-kbps*	Specifies the amount of bandwidth to be assigned to the class, in kbps, or as a percentage of the available bandwidth. Bandwidth must be specified in kbps or as a percentage consistently across classes. (Bandwidth of the priority queue must be specified in kbps.) **Note** To configure more than one class in the same policy map, repeat Configuring Class Policy Using a Specified Bandwidth, page 29 and Configuring Class Policy Using a Specified Bandwidth, page 29.

Configuring the Class-Default Class Policy

SUMMARY STEPS

1. Router(config)# policy-map *policy-map*
2. Router(config-cmap)# class class-default default-class-name
3. Router(config-pmap-c)# bandwidth *bandwidth-kbps*

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# policy-map *policy-map*	Specifies the name of the policy map to be created or modified.
Step 2	Router(config-cmap)# class class-default default-class-name	Specifies the default class so that you can configure or modify its policy. **Note** The class-default class is used to classify traffic that does not fall into one of the defined classes. Even though the class-default class is predefined when you create the policy map, you still have to configure it. If a default class is not configured, then traffic that does not match any of the configured classes is given best-effort treatment, which means that the network will deliver the traffic if it can, without any assurance of reliability, delay prevention, or throughput.

Step 3	Router(config-pmap-c)# bandwidth *bandwidth-kbps*
	Example:
	Example:
	`or`
	Example:
	Example:
	`Router(config-pmap-c)# fair-queue [number-of-dynamic-queues]`
		Specifies the amount of bandwidth, in kbps, to be assigned to the class. Specifies the number of dynamic queues to be reserved for use by flow-based WFQ running on the default class. The number of dynamic queues is derived from the bandwidth of the interface.

Attaching the Service Policy

SUMMARY STEPS

1. Router(config)# interface *type number*
2. Router(config-if)# service-policy output *policy-map*

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# interface *type number*	Specifies the interface using the LLQ for IPsec encryption engines.
Step 2	Router(config-if)# service-policy output *policy-map*	Attaches the specified service policy map to the output interface and enables LLQ for IPsec encryption engines.

Verifying Configuration of Policy Maps and Their Classes

SUMMARY STEPS

1. Router# show frame-relay pvc dlci
2. Router# show policy-map interface type number
3. Router# show policy-map interface interface-name dlci dlci

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router# show frame-relay pvc dlci	Displays statistics about the PVC and the configuration of classes for the policy map on the specified data-link connection identifier (DLCI).
Step 2	Router# show policy-map interface type number	When LLQ is configured, displays the configuration of classes for all policy maps.
Step 3	Router# show policy-map interface interface-name dlci dlci	When LLQ is configured, displays the configuration of classes for the policy map on the specified DLCI.

Monitoring and Maintaining LLQ for IPsec Encryption Engines

SUMMARY STEPS

1. Router# show crypto eng qos

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router# show crypto eng qos	Displays quality of service queueing statistics for LLQ for IPsec encryption engines.

Configuration Examples

LLQ for IPsec Encryption Engines Example

In the following example, a strict priority queue with a guaranteed allowed bandwidth of 50 kbps is reserved for traffic that is sent from the source address 10.10.10.10 to the destination address 10.10.10.20, in the range of ports 16384 through 20000 and 53000 through 56000.

First, the following commands configure access list 102 to match the desired voice traffic:

```
Router(config)# access-list 102 permit udp host 10.10.10.10 host 10.10.10.20 range 16384
20000
Router(config)# access-list 102 permit udp host 10.10.10.10 host 10.10.10.20 range 53000
56000
```

Next, the class map voice is defined, and the policy map called policy1 is created; a strict priority queue for the class voice is reserved, a bandwidth of 20 kbps is configured for the class bar, and the default class is configured for WFQ. The service-policy command then attaches the policy map to the fas0/0.

```
Router(config)# class-map voice
Router(config-cmap)# match access-group 102
Router(config-cmap)# exit
Router(config)# policy-map policy1
Router(config-pmap)# class voice
Router(config-pmap-c)# priority 50
Router (config-cmap-c)# exit
Router(config-pmap)# class bar
Router(config-pmap-c)# bandwidth 20
Router(config-cmap-c)# exit
Router(config-pmap)# class class-default
Router(config-pmap-c)# fair-queue
Router(config-cmap-c)# exit
Router(config-cmap)# exit
Router(config)# interface fastethernet0/0/0
Router(config-if)# service-policy output policy1
```

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Configurable Queue Depth

This feature allows you to configure (resize) the depth of the packet queues on your network. That is, you can set the maximum number (the depth) of packets that a class queue can hold, which in turn controls when the router drops packets. Configuring the depth of the packet queues helps alleviate packet queue congestion.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About Configuring Queue Depth

Queue Limit

Each queue has a limit on the number of packets that the router can place into the queue. This limit, referred to as the depth, is a user-configurable limit. During periods of high traffic, a queue fills with packets that are waiting for transmission. When a queue reaches its queue limit and becomes full, by default, the router drops packets until the queue is no longer full.

For the Cisco ASR 1000 Series Router in Cisco IOS XE Software Release 2.1, the packets-per-queue range is 1 to 2,000,000.

When a packet queue temporarily experiences congestion, increasing the depth of the queue using the queue-limit command reduces the number of packets dropped. However, setting the queue limit to a high value might reduce the number of packet buffers available to other interfaces.

If you do not specify a queue limit, the router calculates the default buffer size for each class queue as follows:

- Class queues with weighted random early detection (WRED)--The router uses the default queue limit of two times the largest WRED maximum threshold value, rounded to the nearest power of 2.
- Class queues without WRED--The router uses 50 ms of 1500-byte packets but never less than 64 packets.
- Priority queues with WRED--The router uses a queue limit of 512 packets.

Note Priority queues without WRED are not allowed.

Note When setting the queue limit, decide how many users will be active at any given time and tune the queue limits accordingly. This will allow individual interfaces to handle traffic bursts and not deplete the available memory. For assistance, contact the Cisco Support website at http://www.cisco.com/techsupport.

How to Configure Queue Depth

This section contains the following tasks:

- Setting the Depth of a Traffic Class Queue, page 36
- Verifying the Depth of the Traffic Class Queue, page 38

Setting the Depth of a Traffic Class Queue

The traffic classes, class maps, and policy maps must exist.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** *class-map-name*
5. **bandwidth** {*bandwidth-kbps* | **percent** *percent*
6. **queue-limit** *number-of-packets*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map-name* **Example:** `Router(config)# policy-map Policy1`	Specifies the name of the policy map and enters policy-map configuration mode. • Enter the policy map name.
Step 4	**class** *class-map-name* **Example:** `Router(config-pmap)# class Class1`	Assigns the traffic class you specify to the policy map. Enters policy-map class configuration mode. • Enter the name of a previously configured class map. This is the traffic class for which you want to enable QoS features.
Step 5	**bandwidth** {*bandwidth-kbps* | **percent** *percent* **Example:** `Router(config-pmap-c)# bandwidth 3000`	Specifies the amount of bandwidth (in kbps or as a percentage of available bandwidth) to be assigned to the class. • Enter the amount of bandwidth. The amount of bandwidth configured should be large enough to also accommodate Layer 2 overhead.
Step 6	**queue-limit** *number-of-packets* **Example:** `Router(config-pmap-c)# queue-limit 32` **Example:**	Specifies or modifies the maximum number of packets that the queue can hold for this class. • Enter the maximum number of packets as applicable.

	Command or Action	Purpose
Step 7	**end** **Example:** `Router(config-pmap-c)# end`	(Optional) Exits policy-map class mode.

Verifying the Depth of the Traffic Class Queue

SUMMARY STEPS

1. **enable**
2. **show policy-map interface** *type number*
3. **exit**

DETAILED STEPS

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

Step 2	**show policy-map interface** *type number*
	
	Example:
	`Router#`
	`show policy-map interface serial4/0/0`
		Displays the packet statistics of all classes that are configured for all service policies either on the specified interface or subinterface or on a specific PVC on the interface.
		• Enter the interface type and number.

Step 3	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

Configuration Examples for Configuring Queue Depth

- Example Setting the Queue Size, page 39
- Example Verifying the Queue Size, page 39

Example Setting the Queue Size

The following example shows how to create a policy map named Policy1 that contains two classes named Class1 and Class2. The Class1 configuration enable a specific bandwidth allocation and specifies the maximum number of packets that can be queued for the class. Because Class1 limits the number of packets that can be held in the queue to 32, the router uses tail drop to drop packets when that limit is reached. Class2 enables bandwidth allocation only.

```
Router(config)# policy-map Policy1
Router(config-pmap)# class Class1
Router(config-pmap-c)# bandwidth 3000
Router(config-pmap-c)# queue-limit 32
Router(config-pmap-c)# exit
Router(config-pmap)# class Class2
Router(config-pmap-c)# bandwidth 2000
Router(config-pmap-c)# end
```

Example Verifying the Queue Size

Use the **show policy-map interface** command to display traffic statistics for the class maps, policy maps, and traffic queues on your network.

The following is sample output for the show policy-map interface command. In this example, the policy map named Traffic-5-PR is attached to serial interface 1/0/0 and includes three traffic classes. The Voice-5-PR class has a configured queue limit of 32 packets with 0 packets dropped. The Gold-5-PR class also indicates that no packets dropped. The Silver-5-PR class has a configured queue limit of 64 packets with 0 packets dropped.

```
Router# show policy-map interface serial 1/0/0
 Serial1/0/0
  Service-policy output: Traffic-Parent (1051)
    Class-map: class-default (match-any) (1068/0)
      2064335 packets, 120273127 bytes
      5 minute offered rate 1000 bps, drop rate 0 bps
      Match: any  (1069)
        126970 packets, 3982597 bytes
        5 minute rate 0 bps
      Shape : 6000 kbps
      Service-policy : Traffic-5-PR (1052)
        Class-map: Voice-5-PR (match-all) (1053/1)
          82310 packets, 4938600 bytes
          5 minute offered rate 0 bps, drop rate 0 bps
          Match: ip precedence 5  (1054)
          Output queue: 0/32; 82310/4938600 packets/bytes output, 0 drops
          Absolute priority
          Queue-limit: 32 packets
          Police:
            304000 bps, 1536 limit, 0 extended limit
            conformed 82312 packets, 4938720 bytes; action: transmit
            exceeded 0 packets, 0 bytes; action: drop
            violated 0 packets, 0 bytes; action: drop
        Class-map: Gold-5-PR (match-any) (1058/2)
          1125476 packets, 67528560 bytes
          5 minute offered rate 0 bps, drop rate 0 bps
          Match: ip precedence 3  4  (1059)
            1125476 packets, 67528560 bytes
            5 minute rate 0 bps
          Output queue: 0/128; 1125503/67530180 packets/bytes output, 0 drops
          Bandwidth : 188 kbps (Weight 3)
        Class-map: Silver-5-PR (match-any) (1061/3)
          697908 packets, 41874480 bytes
          5 minute offered rate 0 bps, drop rate 0 bps
          Match: ip precedence 0  1  2  (1062)
            697908 packets, 41874480 bytes
```

```
    5 minute rate 0 bps
  Output queue: 0/64; 697919/41875140 packets/bytes output, 0 drops
  Bandwidth : 71 kbps (Weight 1)
  Random-detect (precedence-based):
    Exponential weight: 9 (1/512)
    Current average queue length: 0 packets
    -------------------------------------------------
              Min   Max Prob    Rand-Drops Tail-Drops
    -------------------------------------------------
          0    16    32 1/10            0          0
          1    18    32 1/10            0          0
          2    20    32 1/10            0          0
          3    22    32 1/10            0          0
          4    24    32 1/10            0          0
          5    26    32 1/10            0          0
          6    28    32 1/10            0          0
          7    30    32 1/10            0          0
  Queue-limit: 64 packets
Class-map: class-default (match-any) (1066/0)
  158641 packets, 5931487 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: any  (1067)
    158641 packets, 5931487 bytes
    5 minute rate 0 bps
  Output queue: 0/128; 31672/1695625 packets/bytes output, 0 drops
```

Additional References

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
Packet classification	"Classifying Network Traffic" module
Creating classes, class maps, and policy maps	"Applying QoS Features Using the MQC" module

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE Software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring Queue Depth

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 3 Feature Information for Configuring Queue Depth

Feature Name	Releases	Feature Information
Configurable Queue Depth	Cisco IOS XE Release 2.1	This feature allows you to configure (resize) the depth of the packet queues on your network. That is, you can set the maximum number (the depth) of packets that a class queue can hold, which in turn controls when the router drops packets. Configuring the depth of the packet queues helps alleviate packet queue congestion. The following command was introduced or modified: **queue-limit**.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Multi-Level Priority Queues

The Multi-Level Priority Queues (MPQ) feature allows you to configure multiple priority queues for multiple traffic classes by specifying a different priority level for each of the traffic classes in a single service policy map. You can configure multiple service policy maps per router. Having multiple priority queues enables the router to place delay-sensitive traffic (for example, voice) on the outbound link before delay-insensitive traffic. As a result, high-priority traffic receives the lowest latency possible on the router.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Multi-Level Priority Queues

You must configure traffic classes using the class-map command.

Restrictions for Multi-Level Priority Queues

- You cannot configure both the priority command and the priority level command for two different classes in the same policy map.
- You cannot specify the same priority level for two different classes in the same policy map.

- You cannot configure the default queue as a priority queue at any level. For example, the router rejects the following configuration:

```
policy-map P1
 class class-default
  priority level 1
```

- You cannot configure the bandwidth command and multi-level priority queueing on the same class. For example, the router rejects the following configuration:

```
policy-map P1
 class C1
  priority level 1
  bandwidth 200
```

- You cannot configure the shape command and multi-level priority queueing on the same class. For example, the router rejects the following configuration:

```
policy-map P1
 class C1
  priority level 1
  shape average 56000
```

- To convert a one-level (flat) service policy with multiple priority queueing configured to a hierarchical multi-level priority queueing service policy, you must first detach the flat service policy from the interface using the no service-policy command and then add a child policy map to it.

Information About Multi-Level Priority Queues

Benefits of Multi-Level Priority Queues

The Multi-Level Priority Queues (MPQ) feature allows you to configure multiple priority queues for multiple traffic classes by specifying a different priority level for each of the traffic classes in a single service policy map. You can configure multiple service policy maps per router.

Previously, routers based on Cisco IOS XE Software could have only one strict priority queue per policy map for all delay-sensitive traffic--the router associated all priority traffic with this one single priority queue. However, having only one priority queue can cause significant delay in delivering traffic, especially if the router sends high-priority traffic (for example, voice) behind low-priority traffic (for example, video). Using class-based weighted fair queueing (CBWFQ) to reduce delay by heavily weighting one queue can affect the granularity of bandwidth allocations to the other queues. The MPQ feature addresses these issues and improves latency.

Multi-Level Priority Queues Functionality

The priority command is used to specify that a class of traffic has latency requirements with respect to other classes. For multiple priority queues, you can use the priority level command to configure a level of priority service on a class in a policy map. Currently, the router supports two priority levels: level 1 (high) and level 2 (low). The router places traffic with a high-priority level on the outbound link ahead of traffic with a low-priority level. High-priority packets, therefore, are not delayed behind low-priority packets.

The router services the high-level priority queues until empty before servicing the next-level priority queues and non-priority queues. While the router services a queue, the service rate is as fast as possible and is constrained only by the rate of the underlying link or parent node in a hierarchy. If a rate is configured and the router determines that a traffic stream has exceeded the configured rate, the router drops the exceeding packets during periods of congestion. If the link is currently not congested, the router places the exceeding packets onto the outbound link.

When configuring MPQ on different traffic classes in a policy map, you must specify different priority levels for the traffic classes. For example, configure one traffic class to have priority level 2 and another class to have priority level 1.

Note In a hierarchical MPQ configuration in which *all* traffic is sent through the level 2 priority queue only, the traffic sent through the level 2 priority queue receives the same treatment as the traffic sent through the level 1 priority queue.

If high-priority traffic is not policed appropriately, bandwidth starvation of low-priority traffic can occur. Therefore, though not required, we recommend that you configure a policer for high-priority traffic using the police command. If you configure the police command for priority queues, the traffic rate is policed to the police rate for each of the priority queues.

You cannot configure the priority command and the priority level command on different classes in the same policy map.

Traffic Policing and Multi-Level Priority Queues

Bandwidth guarantees can be given to other classes only if traffic policing is enabled on the priority queue.

Using the **priority** and **police** commands, multi-level priority queues can be configured to police traffic in one of the following ways:

- Conditional traffic policing, as illustrated below:

```
policy-map my_policy
  class voice
    priority 400000   <<< Priority queue conditionally policed to 400M
  class gold
    bandwidth 400000 <<<< 400M min guaranteed to class gold
```

With conditional traffic policing on the queue, you run the risk of sudden degradation in priority service when an interface becomes congested. You can go from an instance of a priority class using the entire link to suddenly traffic being policed to configured value. You need to know the available bandwidth and use some form of admission control to ensure your offered loads do not exceed the available bandwidth.

Note With the conditional policing, traffic policing does not engage unless the interface is congested.

- Unconditional traffic policing, as illustrated below:

```
policy-map my_policy
  class voice
    priority           <<< Indicates priority scheduling
    police 400000000   <<< Traffic policed to 400M
```

```
class gold
  bandwidth 400000  <<<400M min guaranteed to class gold
```

The priority class is configured with an "always on" (unconditional) policer. The priority class is always policed to the configured value regardless of whether the interface is congested.

The advantage of an unconditional policer is that you always know how much priority traffic will be offered to the downstream devices, thus making your bandwidth planning much simpler.

This is the recommended choice.

- Absolute priority queue (no traffic policing)

If traffic policing is not configured, the priority traffic may consume the entire interface bandwidth.

How to Configure Multi-Level Priority Queues

- Configuring Multi-Level Priority Queues in a Policy Map, page 46
- Verifying Multi-Level Priority Queues, page 48

Configuring Multi-Level Priority Queues in a Policy Map

The traffic classes, class maps, and policy maps must exist.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. policy-map policy-name
4. class class-name
5. priority level level
6. **police cir** *bps*
7.
8. police cir percent percent
9. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	policy-map policy-name **Example:** `Router(config)# policy-map Premium`	Creates or modifies a policy map and enters policy-map configuration mode. • Enter the name of the policy map.
Step 4	class class-name **Example:** `Router(config-pmap)# class business`	Specifies a traffic class and enters policy-map class configuration mode. • Enter the name of a previously configured traffic class.
Step 5	priority level level **Example:** `Router(config-pmap-c)# priority level 2`	Assigns priority to a traffic class at the priority level specified. • Enter the level of priority assigned to the priority class. Valid values are 1 (high priority) and 2 (low priority). The default is 1. **Note** Do not specify the same priority level for two different classes in the same policy map.
Step 6	**police cir** *bps* **Example:** `Router(config-pmap-c)# police cir 8000` **Example:**	(Optional) Configures traffic policing based on a bits per second (bps) rate. • Enter the **cir** keyword and a value for the *bps* argument. Note the following: ◦ cir is the committed information rate and is based on the interface shape rate. This keyword indicates an average rate at which the policer meters traffic. ◦ bps specifies the average rate in bits per second (bps). Valid values are from 8000 to 2488320000 bps.
Step 7		

	Command or Action	Purpose
Step 8	police cir percent percent **Example:** `Router(config-pmap-c)# police cir percent 20` **Example:**	(Optional) Configures traffic policing based on a percentage of bandwidth available on the interface. • Enter the **cir** keyword, the **percent** keyword, and a value for the *percent* argument. Note the following: ◦ cir is the committed information rate and is based on the interface shape rate. Indicates an average rate at which the policer meters traffic. ◦ percent percent indicates to use the percentage of available bandwidth specified in percent to calculate the CIR. Valid values are from 1 to 100.
Step 9	**end** **Example:** `Router(config-pmap-c)# end`	(Optional) Exits policy-map class mode.

Verifying Multi-Level Priority Queues

SUMMARY STEPS

1. **enable**
2. **show policy-map interface** *type number*
3. **exit**

DETAILED STEPS

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

Step 2	**show policy-map interface** *type number*
	
	Example:
	
	`Router#`
	`show policy-map interface serial4/0/0`
		Displays the packet statistics of all classes that are configured for all service policies either on the specified interface or subinterface or on a specific PVC on the interface.
		
		• Enter the interface type and number.

	Command or Action	Purpose
Step 3	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

Configuration Examples for Multi-Level Priority Queues

Example Configuring Multi-Level Priority Queues

The following example shows how to configure multiple priority queues. The policy map named Business has two traffic classes: Bronze and Gold. Bronze traffic has a level 2 (low) priority, while Gold traffic has level 1 (high) priority. To prevent bandwidth starvation of Bronze traffic, the Gold traffic is policed at 30 percent of the interface bandwidth.

```
Router> enable
Router# configure terminal
Router(config)# policy-map Business
Router(config-pmap)# class Bronze
Router(config-pmap-c)# priority level 2
Router(config-pmap-c)# police cir 1000
Router(config-pmap-c)# exit
Router(config-pmap)# class Gold
Router(config-pmap-c)# priority level 1
Router(config-pmap-c)# police cir percent 30
Router(config-pmap-c)# end
```

Note Although a policer is not required, configure policing for priority traffic to prevent bandwidth starvation of low priority traffic. When policing is configured, the traffic rate is policed to the police rate for each of the priority queues.

Example Unacceptable MPQ Configurations

You cannot specify both the priority command and the priority level command for two different classes in the same policy map. For example, the router does not accept the following configuration:

```
Router> enable
Router# configure terminal
Router(config)# policy-map Map1
Router(config-pmap)# class Bronze
Router(config-pmap-c)# priority level 1
Router(config-pmap-c)# exit
Router(config-pmap)# class Gold
```

```
Router(config-pmap-c)# priority rate 1000
Router(config-pmap-c)# end
```

You cannot specify the same priority level for two different classes in the same policy map. For example, the router does not accept the following configuration:

```
Router> enable
Router# configure terminal
Router(config)# policy-map Map1
Router(config-pmap)# class Bronze
Router(config-pmap-c)# priority level 1
Router(config-pmap-c)# police cir percent 30
Router(config-pmap-c)# exit
Router(config-pmap)# class Gold
Router(config-pmap-c)# priority level 1
Router(config-pmap-c)# police cir 10000
Router(config-pmap-c)# end
```

Example Verifying Multi-Level Priority Queues

The following is partial sample output from the show policy-map interface command.

```
Router# show policy-map interface serial2/1/0
Serial2/1/0
Service-policy output: P1
Queue statistics for all priority classes:
.
.
.
Class-map: Gold (match-all)
0 packets, 0 bytes                /*Updated for each priority level configured.*/
5 minute offered rate 0 bps, drop rate 0 bps
Match: ip precedence 2
Priority: 0 kbps, burst bytes 1500, b/w exceed drops: 0
Priority Level 2:
0 packets, 0 bytes
```

Additional References

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
Priority queues	"Applying QoS Features Using the MQC" module
Creating classes, class maps, and policy maps	"Applying QoS Features Using the MQC" module

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS XE Software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported, and support for existing RFCs has not been modified.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Multi-Level Priority Queues

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 4 Feature Information for Multi-Level Priority Queues

Feature Name	Releases	Feature Information
Multi-Level Priority Queues	Cisco IOS XE 2.1	The Multi-Level Priority Queues (MPQ) feature allows you to configure multiple priority queues for multiple traffic classes by specifying a different priority level for each of the traffic classes in a single service policy map. You can configure multiple service policy maps per router. Having multiple priority queues enables the router to place delay-sensitive traffic (for example, voice) on the outbound link before delay-insensitive traffic. As a result, high-priority traffic receives the lowest latency possible on the router. This feature was implemented on the Cisco ASR 1000 Series Routers. The following commands were introduced or modified: **priority level, show policy-map interface**.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

QoS Hierarchical Queueing for Ethernet DSLAMs

This feature module describes how to configure quality of service (QoS) hierarchical queueing policy maps on sessions and subinterfaces in Ethernet Digital Subscriber Line Access Multiplexer (E-DSLAM) applications on a Cisco ASR 1000 series router. The QoS Hierarchical Queueing for Ethernet DSLAMs feature supports IEEE 802.1 QinQ VLAN tag termination to configure inner VLAN identifiers on E-DSLAMs.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for QoS Hierarchical Queueing for Ethernet DSLAMs

You must configure traffic classes using the class-map command.

Restrictions for QoS Hierarchical Queueing for Ethernet DSLAMs

This feature is not supported in combination with load balancing when a session service policy is routed to a Layer 2 Tunnel Protocol (L2TP) tunnel. Do not configure load balancing on an L2TP tunnel if per-session queueing is enabled.

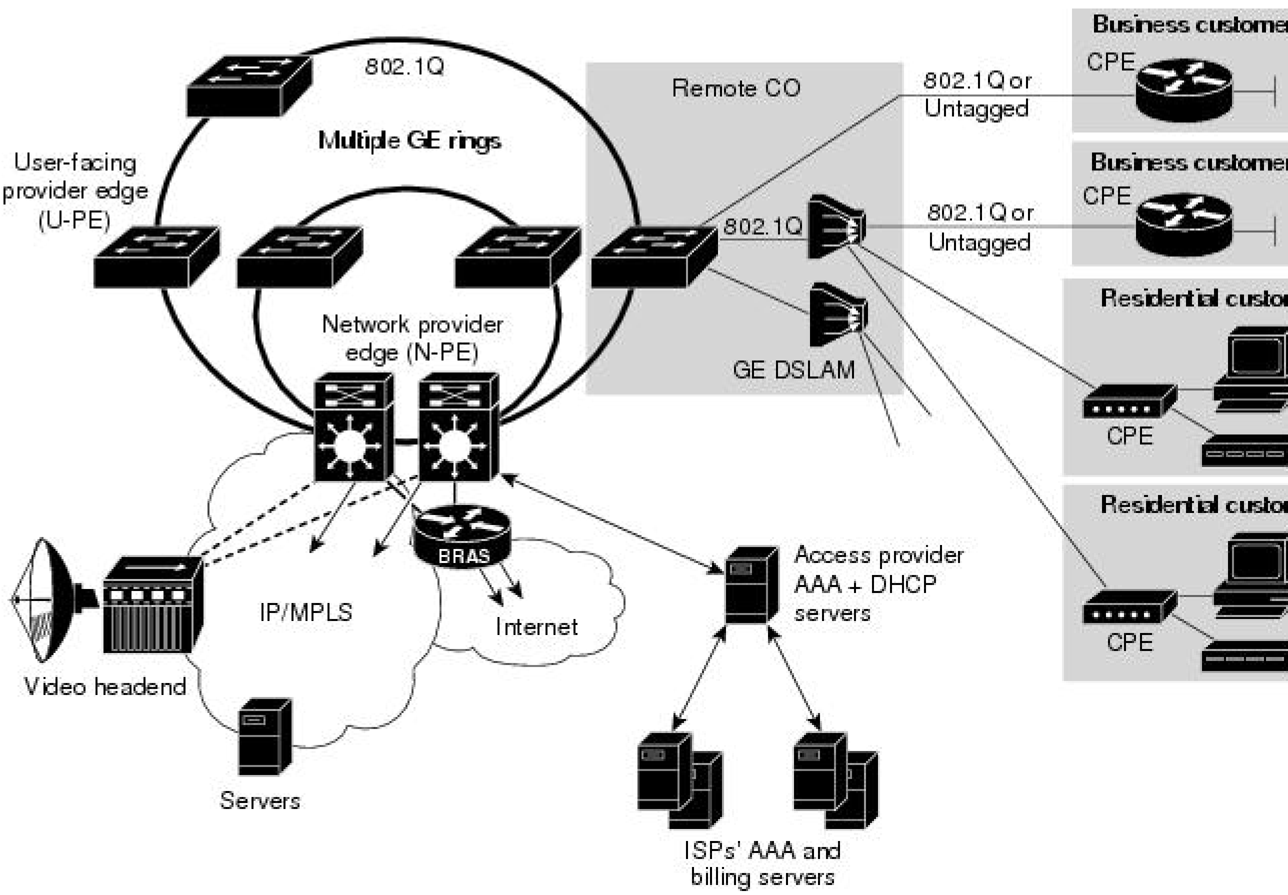
Information About QoS Hierarchical Queueing for Ethernet DSLAMs

- Different Levels of QoS Provisioning, page 54
- Configuration Guidelines for Hierarchical Queueing on Ethernet DSLAMs, page 55

Different Levels of QoS Provisioning

Traffic downstream from a Broadband Router Access Server (BRAS) requires different levels of QoS provisioning (for example, traffic shaping) depending on the network architecture between the BRAS and the subscriber. The figure below illustrates an Ethernet DSL access network. The sample network includes multiple entities where QoS provisioning is required for different reasons.

Figure 3 Ethernet DSL Access Network



The following entities may require different traffic shaping:

- A VLAN that is shaped to a certain aggregate traffic rate to limit the traffic to a group of subscribers (different 802.1Q interfaces in the figure above).
- Individual sessions that is shaped with certain QoS services for different classes of traffic (individual PCs in the figure above).

Integrated Queueing Hierarchy

Different traffic shaping requirements result in QoS provisioning at multiple levels at the same time. The QoS-Hierarchical Queueing for Ethernet DSLAMs feature provides the ability to form one integrated queueing hierarchy that provides QoS provisioning at multiple levels with support for features such as bandwidth distribution at any of these levels.

The integrated queueing hierarchy is formed on the physical interface. When a service policy is instantiated on a session, the Subscriber Service Switch (SSS) infrastructure invokes the MQC and a common queueing control plane sets up and enables the queueing features.

Session-to-interface associations are resolved to determine the physical interface on which to form the integrated queueing hierarchy for all levels of QoS provisioning. As subinterface session-based policies are added, the respective queues are created and integrated into the queueing hierarchy.

When a subinterface is provisioned followed by session-based policy provisioning, the integrated queueing hierarchy is formed on top of the physical interface as a result of queueing policies provisioned at two different levels. When a session is provisioned before subinterface-based policy provisioning, the queueing hierarchy has a placeholder logical level between the physical queue and the session queue. The placeholder queue becomes the default queue at that level, and all other sessions are parented to that queue.

Configuration Guidelines for Hierarchical Queueing on Ethernet DSLAMs

When configuring the QoS Hierarchical Queueing for Ethernet DSLAMs feature, note the following guidelines:

- An individual subscriber is always identified by a PPP or IP session. A group of subscribers is identified by a particular VLAN by means of the outer tag ISP, E-DSLAM, or user-facing provider edge (U-PE).
- When a subinterface is used to aggregate a number of sessions with queueing policies, a queueing policy at a subinterface level must be a one-level policy map that is configured as class-default with only the shape and bandwidth remaining ratio feature enabled.
- Both subinterfaces and sessions can be oversubscribed and controlled by shaper and bandwidth remaining ratio.

How to Configure QoS Hierarchical Queueing for Ethernet DSLAMs

Configuring and Applying QoS Hierarchical Queueing Policy Maps to Sessions

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** class-map-name
5. bandwidth {*bandwidth-kbps* | **percent***percentage*| **remainingpercent***percentage*}
6. **precedence** *precedence min-threshold max-threshold mark-probability-denominator*
7. **set cos** *cos-value*
8. **exit**
9. **exit**
10. **policy-map** *policy-map-name*
11. **class class-default**
12. **shape** average *cir*
13. **bandwidth remaining ratio** *ratio*
14. **service-polic** y*policy-map-name*
15. **exit**
16. **exit**
17. **interface virtual-template** *number*
18. **service-policy output** *policy-map-name*
19. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**policy-map** *policy-map-name* **Example:** `Router(config)# policy-map session_a_child`	Creates a child policy and enters policy-map configuration mode. • Enter the policy-map name.
Step 4	**class** class-map-name **Example:** `Router(config-pmap)# class voip`	Configures the traffic class that you specify and enters policy-map class configuration mode. • Enter the name of a previously configured class map .
Step 5	bandwidth {*bandwidth-kbps* | **percent***percentage*| **remainingpercent***percentage*} **Example:** `Router(config-pmap-c)# bandwidth 10000` **Example:**	(Optional) Enables class-based weighted fair queueing based on the keywords and arguments specified, as described below. • bandwidth-kbps--Specifies the minimum bandwidth allocated for a class belonging to a policy map. Valid values are from 8 to 2,488,320, which represents from 1 to 99 percent of the link bandwidth. • percent percentage--Specifies the minimum percentage of the link bandwidth allocated for a class belonging to a policy map. Valid values are from 1 to 99. • remaining percent percentage--Specifies the minimum percentage of unused link bandwidth allocated for a class belonging to a policy map. Valid values are from 1 to 99.
Step 6	**precedence** *precedence min-threshold max-threshold mark-probability-denominator* **Example:** `Router(config-pmap-c)# precedence 0 32 256 100`	(Optional) Configures a precedence level for the traffic class based on the arguments specified, as described below. • precedence--Specifies the IP precedence number. Valid values are from 0 to 7. • min-threshold--Specifies the minimum threshold in number of packets. Valid values are from 1 to 4096. • max-threshold--Specifies the maximum threshold in number of packets. Valid values are from the minimum threshold to 4096. • mark-probability-denominator--Specifies the denominator for the fraction of packets dropped when the average queue depth is equal to the maximum threshold. For example, if the denominator is 512, 1 out of every 512 packets is dropped when the average queue is at the maximum threshold. Valid values are from 1 to 65536. The default value is 10 (1 out of every 10 packets is dropped at the maximum threshold).

	Command or Action	Purpose
Step 7	**set cos** *cos-value* **Example:** `Router(config-pmap-c)# set cos 1`	(Optional) Sets the Layer 2 class of service (CoS) value of an outgoing packet. • Enter the IEEE 802.1Q CoS value from 0 to 7. **Note** Use the set cos command only in service policies that are attached in the output direction of an interface; packets that enter an interface cannot be set with a CoS value. You can configure a CoS value on an Ethernet interface that is configured for 802.1Q or on a virtual access interface that is using an 802.1Q interface.
Step 8	**exit** **Example:** `Router(config-pmap-c)# exit`	Exits policy-map class configuration mode.
Step 9	**exit** **Example:** `Router(config-pmap)# exit`	Exits policy-map configuration mode.
Step 10	**policy-map** *policy-map-name* **Example:** `Router(config)# policy-map session_a_parent`	Creates a parent policy and enters policy-map configuration mode. • Enter the policy-map name.
Step 11	**class class-default** **Example:** `Router(config-pmap)# class class-default`	Configures the traffic class as class-default and enters policy-map class configuration mode. **Note** Do not configure any other traffic class.
Step 12	**shape** average *cir* **Example:** `Router(config-pmap-c)# shape average 10000000`	Specifies average-rate traffic shaping for all traffic that does not match any other traffic class. • Enter the **average** keyword followed by the committed information rate (CIR), in bits per second (bps).

	Command or Action	Purpose
Step 13	**bandwidth remaining ratio** *ratio* **Example:** `Router(config-pmap-c)# bandwidth remaining ratio 10`	Specifies the weight (ratio) for the subinterface. • Enter the relative weight of this subinterface (or class queue). This number (ratio) indicates the proportional relationship between the other subinterfaces or class queues.
Step 14	**service-polic** y*policy-map-name* **Example:** `Router(config-pmap-c)# service-policy session_a_child`	Applies the child policy map to the parent class-default class. • Enter the name of a previously configured child policy map.
Step 15	**exit** **Example:** `Router(config-pmap-c)# exit`	Exits policy-map class configuration mode.
Step 16	**exit** **Example:** `Router(config-pmap)# exit`	Exits policy-map configuration mode.
Step 17	**interface virtual-template** *number* **Example:** `Router(config)# interface virtual-template 1`	Creates a virtual template and enters interface configuration mode. • Enter the virtual template number. Valid range is from 1 to 4095.
Step 18	**service-policy output** *policy-map-name* **Example:** `Router(config-if)# service-policy output session_a_parent`	Applies the service policy to the virtual interface. • Enter the name of the previously configured parent policy map. **Note** You must specify the output keyword to apply the service policy to outbound traffic on the interface.
Step 19	**end** **Example:** `Router(config-if)# end`	(Optional) Returns to privileged EXEC mode.

Examples

The following is an example of how to configure and apply a QoS hierarchical queueing policy map to PPP/IP sessions by using a virtual template:

```
Router> enable
Router# configure terminal
Router(config)# policy-map session_a_child
Router(config-pmap)# class voip
Router(config-pmap-c)# police 1000000
Router(config-pmap-c)# priority level 1
Router(config-pmap-c)# exit
Router(config-pmap)# class video
Router(config-pmap-c)# police 100000
Router(config-pmap-c)# priority level 2
Router(config-pmap-c)# exit
Router(config-pmap)# class precedence_0
Router(config-pmap-c)# bandwidth remaining ratio 10
Router(config-pmap-c)# exit
Router(config-pmap)# class precedence_1
Router(config-pmap-c)# bandwidth remaining ratio 20
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# policy-map session_a_parent
Router(config-pmap-c)# exit
Router(config-pmap)# class class-default
Router(config-pmap-c)# shape average 10000000
Router(config-pmap-c)# bandwidth remaining ratio 10
Router(config-pmap-c)# service-policy session_a_child
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface virtual-template 20
Router(config-if)# service-policy output session_a_parent
Router(config-if)# end
```

Configuring and Applying QoS Hierarchical Queueing Policy Maps to Subinterfaces

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class class-default**
5. **shape average** *cir*
6. **exit**
7. **exit**
8. interface type slot/subslot/port**.**subinterface
9. **encapsulation dot1q** *outer-vlan-id* [**second-dot1q***inner-vlan-id*]
10. **service-policy output** *policy-map-name*
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map-name* **Example:** `Router(config)# policy-map subint_1`	Creates a policy map and enters policy-map configuration mode. • policy-map-name--The name of the policy map.
Step 4	**class class-default** **Example:** `Router(config-pmap)# class class-default`	Configures the traffic class as class-default and enters policy-map class configuration mode. Do not configure any other traffic class. **Note** When a subinterface aggregates a number of sessions with queueing policies, a queueing policy at a subinterface level must be a one-level policy map configured as class-default.
Step 5	**shape average** *cir* **Example:** `Router(config-pmap-c)# shape average 10000000`	Specifies average-rate traffic shaping for all traffic that does not match any other traffic class. • Enter the **average** keyword followed by the CIR, in bps. **Note** When a subinterface aggregates a number of sessions with queueing policies, a queueing policy at a subinterface level must be a one-level policy map with only the shape feature enabled.
Step 6	**exit** **Example:** `Router(config-pmap-c)# exit`	Exits policy-map class configuration mode.
Step 7	**exit** **Example:** `Router(config-pmap)# exit`	Exits policy-map configuration mode.

	Command or Action	Purpose
Step 8	interface type slot/subslot/port**.**subinterface **Example:** `Router(config)# interface GigabitEthernet3/1/1.1`	Specifies the subinterface on which you are attaching the policy map and enters subinterface configuration mode. • Enter the interface type and slot number, subslot number, port number, and subinterface number.
Step 9	**encapsulation dot1q** *outer-vlan-id* [**second-dot1q***inner-vlan-id*] **Example:** `Router(config-subif)# encapsulation dot1q 100`	Enables IEEE 802.1Q encapsulation of traffic on the subinterface. The **second-dot1q**keywordsupports the IEEE 802.1 QinQ VLAN Tag Termination feature to configure an inner VLAN ID. • outer-vlan-id--The outer VLAN identifier. The range is from 1 to 4095. • inner-vlan-id--The inner VLAN identifier. The range is from 1 to 4095.
Step 10	**service-policy output** *policy-map-name* **Example:** `Router(config-subif)# service-policy output subint_1`	Attaches the service policy to the subinterface. • policy-map-name--The name of the previously configured policy map. **Note** You must specify the output keyword to apply the service policy to outbound traffic on the subinterface.
Step 11	**end** **Example:** `Router(config-subif)# end`	(Optional) Returns to privileged EXEC mode.

Examples

The following is an example of how to configure and apply a QoS hierarchical queueing policy map to a subinterface (and provide aggregate shaping for a large number of subscribers):

```
Router> enable
Router# configure terminal
Router(config)# policy-map subint_1
Router(config-pmap)# class class-default
Router(config-pmap-c)# shape average 10000000
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface GigabitEthernet3/1/1.1
Router(config-subif)# encapsulation dot1q 100
Router(config-subif)# service-policy output subint_1
Router(config-subif)# end
```

Displaying Policy-Map Information for Hierarchical Queueing

SUMMARY STEPS

1. **enable**
2. **show policy-map**
3. **show policy-map interface** *type number*
4. **show policy-map session**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show policy-map** **Example:** `Router# show policy-map`	(Optional) Displays all information for all class maps.

Step 3	**show policy-map interface** *type number*
	
	Example:
	`Router#`
	`show policy-map interface GigabitEthernet4/0/0.1`
		(Optional) Displays the packet statistics of all classes that are configured for all service policies either on the specified interface or subinterface or on a specific PVC on the interface.
		• Enter the interface type and number.

Step 4	**show policy-map session**
	
	Example:
	`Router#`
	`show policy-map session`
		(Optional) Displays the QoS policy map in effect for the SSS session.

Step 5	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

Configuration Examples for QoS Hierarchical Queueing for Ethernet DSLAMs

Example Policy Maps on VLANs or QinQ Subinterfaces

The following example shows how to configure and apply QoS hierarchical queueing policy maps on VLANs or QinQ subinterfaces. A child queueing policy is applied to each parent subscriber line level policy. In this example, the policy maps are applied to create subscriber groups on subinterfaces.

```
Router> enable
Router# configure terminal
Router(config)# policy-map service_a_out
Router(config-pmap)# class voip
Router(config-pmap-c)# priority
Router(config-pmap-c)# police cir percent 20 bc 300 ms pir precent 40
Router(config-pmap-c)# set cos 1
Router(config-pmap-c)# exit
Router(config-pmap)# class video
Router(config-pmap-c)# police cir percent 20 bc 300 ms pir prectent 40
Router(config-pmap-c)# set cos 2
Router(config-pmap-c)# exit
Router(config-pmap)# class gaming
Router(config-pmap-c)# bandwidth remaining percent 80
Router(config-pmap-c)# set cos 3
Router(config-pmap-c)# exit
Router(config-pmap)# class class-default
Router(config-pmap-c)# bandwidth remaining percent 20
Router(config-pmap-c)# set cos 4
Router(config-pmap-c)# exit
Router(config-pmap)# exit
!
Router(config)# policy-map service_z_out
Router(config-pmap)# exit
!
Router(config)# policy-map rate_1_service_a_in
Router(config-pmap)# class voip
Router(config-pmap-c)# police cir percent 25 4 ms 1 ms
Router(config-pmap-c)# exit
Router(config-pmap)# class gaming
Router(config-pmap-c)# police cir percent 50 2 ms 1 ms
Router(config-pmap-c)# exit
Router(config-pmap)# class class-default
Router(config-pmap-c)# police percent 20 bc 300 ms pir 40
Router(config-pmap-c)# exit
Router(config-pmap)# exit
!
Router(config)# policy-map rate_x_service_z_in
Router(config-pmap)# exit
!
Router(config)# policy-map rate_1_service_a_out
Router(config-pmap)# class class-default
Router(config-pmap-c)# bandwidth remaining ratio 10
Router(config-pmap-c)# shape average 100000
Router(config-pmap-c)# service policy service_a_out
Router(config-pmap-c)# exit
Router(config-pmap)# exit
!
```

```
Router(config)# policy-map rate_x_service_z_out
Router(config-pmap)# class class-default
Router(config-pmap-c)# bandwidth remaining ratio 10
Router(config-pmap-c)# shape average 100000
Router(config-pmap-c)# service policy service_z_out
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface GigabitEthernet1/0/0.1
Router(config-subif)# encapsulation dot1q 5 second dot1q 20
Router(config-subif)# service-policy output rate_1_service_a_out
Router(config-subif)# service-policy input rate_1_service_a_in
Router(config-subif)# exit
Router(config)# interface GigabitEthernet1/0/0.2
Router(config-subif)# encapsulation dot1q 5 second dot1q 25
Router(config-subif)# service-policy output rate_x_service_z_out
Router(config-subif)# service-policy input rate_x_service_z_in
Router(config-subif)# end
```

Example Policy Maps on VLANs with Arbitrary QinQ

The following example shows how to configure and apply QoS hierarchical queueing policy maps on VLANs with subscriber lines grouped by arbitrary QinQ. A child queueing policy is applied to each parent subscriber line level policy. This example includes the configuration of multiple class maps.

```
Router> enable
Router# configure terminal
Router(config)# class-map match-all user_1
Router(config-cmap)# match vlan 10
Router(config-cmap)# exit
Router(config)# class-map match-all user_2
Router(config-cmap)# match vlan 11
Router(config-cmap)# exit
Router(config)# class-map match-all user_3
Router(config-cmap)# match vlan 10
Router(config-cmap)# exit
Router(config)# class-map match-any user_4
Router(config-cmap)# match vlan 11
Router(config-cmap)# exit
Router(config)# class-map match-all user_n
Router(config-cmap)# exit
Router(config)# class-map match-any isp_A
Router(config-cmap)# match class user_1
Router(config-cmap)# match class user_2
Router(config-cmap)# exit
Router(config)# class-map match-any isp_Z
Router(config-cmap)# match class user_3
Router(config-cmap)# match class user_4
Router(config-cmap)# exit
!
Router(config)# policy-map service_a_out
Router(config-pmap)# class voip
Router(config-pmap-c)# priority
Router(config-pmap-c)# police cir percent 20 bc 300 ms pir precent 40
Router(config-pmap-c)# set cos 1
Router(config-pmap-c)# exit
Router(config-pmap)# class video
Router(config-pmap-c)# police cir percent 20 bc 300 ms pir precent 40
Router(config-pmap-c)# set cos 2
Router(config-pmap-c)# exit
Router(config-pmap)# class gaming
Router(config-pmap-c)# bandwidth remaining percent 80
Router(config-pmap-c)# set cos 3
Router(config-pmap-c)# exit
Router(config-pmap)# class class-default
Router(config-pmap-c)# bandwidth remaining percent 20
Router(config-pmap-c)# set cos 4
Router(config-pmap-c)# exit
Router(config-pmap)# exit
!
Router(config)# policy-map service_z_out
```

```
Router(config)# policy-map service_a_in
Router(config-pmap)# class voip
Router(config-pmap-c)# police cir percent 25 4 ms 1 ms
Router(config-pmap-c)# exit
Router(config-pmap)# class gaming
Router(config-pmap-c)# police cir percent 50 2 ms 1 ms
Router(config-pmap-c)# exit
Router(config-pmap)# class class-default
Router(config-pmap-c)# police cir percent 20 bc 300 ms pir precent 40
Router(config-pmap-c)# exit
Router(config-pmap)# exit
!
Router(config)# policy-map service_z_in
Router(config-pmap)# exit
!
Router(config)# policy-map isp_A_out
Router(config-pmap)# class user_1
Router(config-pmap-c)# bandwidth remaining ratio 10
Router(config-pmap-c)# shape average 100000
Router(config-pmap-c)# service policy service_a_out
Router(config-pmap-c)# exit
Router(config-pmap)# class user_n
Router(config-pmap-c)# bandwidth remaining ratio 20
Router(config-pmap-c)# shape average 100000
Router(config-pmap-c)# service policy service_z_out
Router(config-pmap-c)# exit
Router(config-pmap)# exit
!
Router(config)# policy-map isp_Z_out
Router(config-pmap)# exit
!
Router(config)# policy-map isp_A_in
Router(config-pmap)# class user_1
Router(config-pmap-c)# service policy service_a_in
Router(config-pmap-c)# class user_n
Router(config-pmap-c)# service policy service_z_in
Router(config-pmap-c)# exit
Router(config-pmap)# exit
!
Router(config)# policy-map isp_Z_in
Router(config-pmap)# exit
!
Router(config)# policy-map interface_policy_out
Router(config-pmap)# class isp_A
Router(config-pmap-c)# shape average 100000
Router(config-pmap-c)# service policy isp_A_out
Router(config-pmap-c)# exit
Router(config-pmap)# class isp_Z
Router(config-pmap-c)# shape average 100000
Router(config-pmap-c)# service policy isp_Z_out
Router(config-pmap-c)# exit
Router(config-pmap)# exit
!
Router(config)# policy-map interface_policy_in
Router(config-pmap)# class isp_A
Router(config-pmap-c)# service policy isp_A_in
Router(config-pmap-c)# exit
Router(config-pmap)# class isp_Z
Router(config-pmap-c)# service policy isp_Z_in
Router(config-pmap-c)# exit
Router(config-pmap)# exit
!
Router(config)# interface GigabitEthernet1/0/0.1
Router(config-subif)# encapsulation dot1q 5 second dot1q any
Router(config-subif)# service-policy output interface_policy_out
Router(config-subif)# service-policy input interface_policy_in
Router(config-subif)# end
```

Example CPolicy Maps on Sessions

The following example shows how to configure and apply QoS hierarchical queueing policy maps on sessions. A child queueing policy is applied to each parent subscriber line level policy.

```
Router> enable
Router# configure terminal
Router(config)# policy-map service_a_out
Router(config-pmap)# class voip
Router(config-pmap-c)# priority
Router(config-pmap-c)# set cos 1
Router(config-pmap-c)# exit
Router(config-pmap)# class video
Router(config-pmap-c)# set cos 2
Router(config-pmap-c)# exit
Router(config-pmap)# class gaming
Router(config-pmap-c)# bandwidth remaining percent 80
Router(config-pmap-c)# set cos 3
Router(config-pmap-c)# exit
Router(config-pmap)# class class-default
Router(config-pmap-c)# bandwidth remaining percent 20
Router(config-pmap-c)# set cos 4
Router(config-pmap-c)# exit
Router(config-pmap)# exit
!
Router(config)# policy-map service_z_out
Router(config-pmap)# exit
!
Router(config)# policy-map rate_1_service_a_out
Router(config-pmap)# class class-default
Router(config-pmap-c)# bandwidth remaining ratio 10
Router(config-pmap-c)# shape average 100000
Router(config-pmap-c)# service-policy service_a_out
Router(config-pmap-c)# exit
Router(config-pmap)# exit
!
Router(config)# policy-map rate_x_service_z_out
Router(config-pmap)# class class-default
Router(config-pmap-c)# bandwidth remaining ratio 10
Router(config-pmap-c)# shape average 100000
Router(config-pmap-c)# service-policy service_z_out
Router(config-pmap-c)# exit
Router(config-pmap)# exit
!
Router(config)# policy-map rate_1_service_a_in
Router(config-pmap)# class voip
Router(config-pmap-c)# police cir percent 25 4 ms 1 ms
Router(config-pmap-c)# exit
Router(config-pmap)# class gaming
Router(config-pmap-c)# police cir percent 50 2 ms 1 ms
Router(config-pmap-c)# exit
Router(config-pmap)# class class-default
Router(config-pmap-c)# police cir percent 20 bc 300 ms pir precent 40
Router(config-pmap-c)# exit
Router(config-pmap)# exit
!
Router(config)# policy-map rate_x_service_z_in
Router(config-pmap)# exit
!
Router(config)# policy-map isp_A_out
Router(config-pmap)# class class-default
Router(config-pmap-c)# shape average 100000
Router(config-pmap-c)# bandwidth remaining ratio 10
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# exit
Router(config)# policy-map isp_Z_out
Router(config-pmap-c)# exit
Router(config-pmap)# class class-default
Router(config-pmap-c)# shape average 200000
```

```
Router(config-pmap-c)# bandwidth remaining ratio 30
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface GigabitEthernet1/0/0.1
Router(config-subif)# encapsulation dot1q 1
Router(config-subif)# service-policy output isp_A_out
Router(config-subif)# exit
Router(config)# interface GigabitEthernet2/0/0.2
Router(config-subif)# encapsulation dot1q 2
Router(config-subif)# service-policy output isp_Z_out
Router(config-subif)# end
```

Example Policy Maps on Sessions with Aggregate Shaping

The following example shows how to configure and apply QoS hierarchical queueing policy maps on sessions with multiple PPP/IP sessions per subscriber line. In this example, the same policies are applied to all sessions using the same virtual interface.

```
Router> enable
Router# configure terminal
Router(config)# policy-map service_a_out
Router(config-pmap)# class voip
Router(config-pmap-c) priority
Router(config-pmap-c)# police cir percent 25 4 ms 1 ms
Router(config-pmap-c)# set cos 1
Router(config-pmap-c)# exit
Router(config-pmap)# class video
Router(config-pmap-c)# police cir percent 30 5 ms 1 ms
Router(config-pmap-c)# set cos 2
Router(config-pmap-c)# exit
Router(config-pmap)# class class-default
Router(config-pmap-c)# bandwidth remaining percent 20
Router(config-pmap-c)# set cos 3
Router(config-pmap-c)# exit
Router(config-pmap)# exit
!
Router(config)# policy-map service_z_out
Router(config-pmap)# exit
!
Router(config)# policy-map rate_1_service_a_in
Router(config-pmap)# class voip
Router(config-pmap-c)# police cir percent 25 4 ms 1 ms
Router(config-pmap-c)# exit
Router(config-pmap)# class video
Router(config-pmap-c)# police cir percent 30 2 ms 1 ms
Router(config-pmap-c)# exit
Router(config-pmap)# class class-default
Router(config-pmap-c)# police cir percent 40 2 ms 1 ms
Router(config-pmap-c)# exit
Router(config-pmap)# exit
!
Router(config)# policy-map rate_x_service_z_in
Router(config-pmap)# exit
!
Router(config)# policy-map rate_1_service_a_out
Router(config-pmap)# class class-default
Router(config-pmap-c)# bandwidth remaining ratio 10
Router(config-pmap-c)# shape average 100000
Router(config-pmap-c)# service policy service_a_out
Router(config-pmap-c)# exit
Router(config-pmap)# exit
!
Router(config)# policy-map rate_x_service_z_out
Router(config-pmap)# class class-default
Router(config-pmap-c)# bandwidth remaining ratio 10
Router(config-pmap-c)# shape average 100000
Router(config-pmap-c)# service policy service_z_out
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface GigabitEthernet1/0/0
```

```
Router(config-if)# encapsulation dot1q 1
Router(config-if)# service-policy output isp_A_out
Router(config-if)# exit
Router(config)# interface GigabitEthernet2/0/0
Router(config-if)# encapsulation dot1q 2
Router(config-if)# service-policy output isp_Z_out
Router(config-if)# end
```

Additional References

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
Traffic shaping	"Regulating Traffic Flow Using Traffic Shaping" module
MQC	"Applying QoS Features Using the MQC" module

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE Software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for QoS Hierarchical Queueing for Ethernet DSLAMs

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 5 Feature Information for QoS Hierarchical Queueing for Ethernet DSLAMs

Feature Name	Releases	Feature Information
QoS Hierarchical Queueing for Ethernet DSLAMs	Cisco IOS XE Release 2.4	This feature module describes how to configure QoS hierarchical queueing policy maps on sessions and subinterfaces in Ethernet Digital Subscriber Line Access Multiplexer (E-DSLAM) applications. This feature was implemented on Cisco ASR 1000 Series Routers.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

QoS Hierarchical Queueing for ATM DSLAMs

This feature module describes how to configure quality of service (QoS) hierarchical queueing policy maps on sessions and ATM VCs in ATM Digital Subscriber Line Access Multiplexer (A-DSLAM) applications on a Cisco ASR 1000 Series Aggregation Services Router.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for QoS Hierarchical Queueing for ATM DSLAMs

You must configure traffic classes using the class-map command.

Restrictions for QoS Hierarchical Queueing for ATM DSLAMs

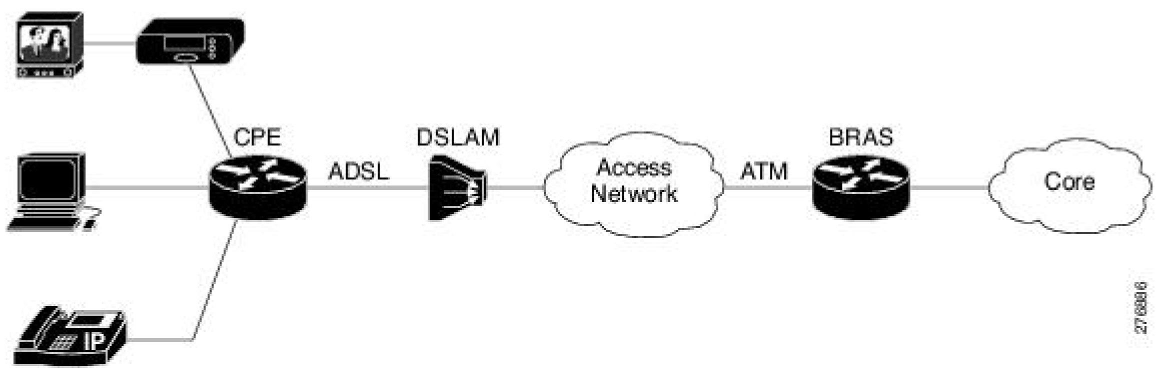
The QoS Hierarchical Queueing for ATM DSLAMs feature is not supported in combination with load balancing when a session service policy is routed to a Layer 2 Tunnel Protocol (L2TP) tunnel. This feature is supported only with shaped ATM VCs, which means ATM VCs that are defined as constant bit rate (CBR), Variable bit rate (VBR) or shaped unspecified bit rate (UBR), (that is, UBR with a peak cell rate).

Information About QoS Hierarchical Queueing for ATM DSLAMs

Different Levels of QoS Provisioning

Traffic downstream from a Broadband Router Access Server (BRAS) requires different levels of QoS provisioning (for example, traffic shaping) depending on the network architecture between the BRAS and the subscriber. The figure below illustrates an ATM DSL access network. The sample network includes multiple entities where QoS provisioning is required for different reasons.

Figure 4 ATM DSL Access Network



Integrated Queueing Hierarchy

Different traffic shaping requirements result in QoS provisioning at multiple levels at the same time. The QoS-Hierarchical Queueing for ATM DSLAMs feature provides the ability to form one integrated queueing hierarchy that provides QoS provisioning at multiple levels with support for features such as bandwidth distribution at any of these levels.

The integrated queueing hierarchy is formed on the physical interface. When a service policy is instantiated on a session, the Subscriber Service Switch (SSS) infrastructure invokes the Modular QoS CLI (MQC) and a common queueing control plane sets up and enables the queueing features.

Session-to-ATM associations are resolved to determine the ATM VC on which the session QoS queues are built. QoS policies consisting of a shaper may also be applied simultaneously at the VC level.

Configuration Guidelines for Hierarchical Queueing on ATM DSLAMs

When configuring the QoS Hierarchical Queueing for ATM DSLAMs feature, note the following guidelines:

- When an ATM VC is used to aggregate a number of sessions with queueing policies, a queueing policy at an ATM VC level must be a one-level policy map that is configured as class-default with only the shape feature enabled.
- Both ATM VCs and sessions can be oversubscribed and controlled by shapers.

How to Configure QoS Hierarchical Queueing for ATM DSLAMs

Configuring and Applying QoS Hierarchical Queueing Policy Maps to Sessions

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** class-map-name
5. bandwidth {*bandwidth-kbps* | **percent***percentage*| **remainingpercent***percentage*}
6. **exit**
7. **exit**
8. **policy-map** *policy-map-name*
9. **class class-default**
10. **shape** average {*cir*| **percent***percentage*}
11. **bandwidth remaining ratio** *ratio*
12. **service-polic** y*policy-map-name*
13. **exit**
14. **exit**
15. **interface virtual-template** *number*
16. **service-policy output** *policy-map-name*
17. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map-name* **Example:** `Router(config)# policy-map session-a-child`	Creates a child policy and enters policy-map configuration mode. • Enter the policy-map name.
Step 4	**class** class-map-name **Example:** `Router(config-pmap)# class voip`	Configures the traffic class that you specify and enters policy-map class configuration mode. • Enter the name of a previously configured class map .
Step 5	bandwidth {*bandwidth-kbps* | **percent***percentage*| **remainingpercent***percentage*} **Example:** `Router(config-pmap-c)# bandwidth 10000` **Example:**	(Optional) Enables class-based weighted fair queueing based on the keywords and arguments specified. • bandwidth-kbps--Specifies the minimum bandwidth allocated for a class belonging to a policy map. Valid values are from 1 to 2,000,000. • percent percentage--Specifies the minimum percentage of the link bandwidth allocated for a class belonging to a policy map. Valid values are from 1 to 100. • remaining percent percentage--Specifies the minimum percentage of unused link bandwidth allocated for a class belonging to a policy map. Valid values are from 1 to 99.
Step 6	**exit** **Example:** `Router(config-pmap-c)# exit`	Exits policy-map class configuration mode.

	Command or Action	Purpose
Step 7	**exit** **Example:** `Router(config-pmap)# exit`	Exits policy-map configuration mode.
Step 8	**policy-map** *policy-map-name* **Example:** `Router(config)# policy-map session_a_parent`	Creates a parent policy and enters policy-map configuration mode. • Enter the policy-map name.
Step 9	**class class-default** **Example:** `Router(config-pmap)# class class-default`	Configures the traffic class as class-default and enters policy-map class configuration mode. **Note** Do not configure any other traffic class.
Step 10	**shape** average {*cir*| **percent***percentage*} **Example:** `Router(config-pmap-c)# shape average 10000000`	Specifies average-rate traffic shaping for all traffic that does not match any other traffic class. • Enter the **average** keyword followed by the committed information rate (CIR), in bits per second (bps), or enter the **average** keyword followed by **percentage** keyword to specify a percentage of the interface bandwidth for the CIR. Valid values are from 1 to 100.
Step 11	**bandwidth remaining ratio** *ratio* **Example:** `Router(config-pmap-c)# bandwidth remaining ratio 10`	Specifies the weight (ratio) for the ATM VC. • Enter the relative weight of this ATM VC (or class queue). This number (ratio) indicates the proportional relationship between the other ATM VCs or class queues.
Step 12	**service-polic** y*policy-map-name* **Example:** `Router(config-pmap-c)# service-policy session-a-child`	Applies the child policy map to the parent class-default class. • Enter the name of a previously configured child policy map.
Step 13	**exit** **Example:** `Router(config-pmap-c)# exit`	Exits policy-map class configuration mode.

	Command or Action	Purpose
Step 14	**exit** **Example:** `Router(config-pmap)# exit`	Exits policy-map configuration mode.
Step 15	**interface virtual-template** *number* **Example:** `Router(config)# interface virtual-template 1`	Creates a virtual template and enters interface configuration mode. • Enter the virtual template number. Valid range is from 1 to 4095.
Step 16	**service-policy output** *policy-map-name* **Example:** `Router(config-if)# service-policy output session_a_parent`	Applies the service policy to the virtual interface. • Enter the name of the previously configured parent policy map. **Note** You must specify the output keyword to apply the service policy to outbound traffic on the interface.
Step 17	**end** **Example:** `Router(config-if)# end`	(Optional) Returns to privileged EXEC mode.

Examples

The following is an example of how to configure and apply a QoS hierarchical queueing policy map to PPP/IP sessions by using a virtual template:

```
Router> enable
Router# configure terminal
Router(config)# policy-map session-a-child
Router(config-pmap)# class voip
Router(config-pmap-c)# police 1000000
Router(config-pmap-c)# priority level 1
Router(config-pmap-c)# exit
Router(config-pmap)# class video
Router(config-pmap-c)# police 100000
Router(config-pmap-c)# priority level 2
Router(config-pmap-c)# exit
Router(config-pmap)# class precedence_0
Router(config-pmap-c)# bandwidth remaining ratio 10
Router(config-pmap-c)# exit
Router(config-pmap)# class precedence_1
Router(config-pmap-c)# bandwidth remaining ratio 20
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# policy-map session_a_parent
Router(config-pmap-c)# exit
Router(config-pmap)# class class-default
Router(config-pmap-c)# shape average 10000000
Router(config-pmap-c)# bandwidth remaining ratio 10
Router(config-pmap-c)# service-policy session-a-child
Router(config-pmap-c)# exit
```

```
Router(config-pmap)# exit
Router(config)# interface virtual-template 20
Router(config-if)# service-policy output session_a_parent
Router(config-if)# end
```

Configuring and Applying QoS Hierarchical Queueing Policy Maps to ATM VCs

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class class-default**
5. **shape average** {*cir*| **percent***percentage*}
6. **exit**
7. **exit**
8. interface type slot/subslot/port**.**subinterface
9. pvc [*name*] *vpi*/*vci* [**ces** | **ilmi** | **qsaal** | **smds**| **l2transport**]
10. **vbr-nrt** *peak-cell-rate average-cell-rate*
11. **service-policy output** *policy-map-name*
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map-name* **Example:** `Router(config)# policy-map subint-1`	Creates a policy map and enters policy-map configuration mode. • policy-map-name--The name of the policy map.

	Command or Action	Purpose
Step 4	**class class-default** **Example:** `Router(config-pmap)# class class-default`	Configures the traffic class as class-default and enters policy-map class configuration mode. • Do not configure any other traffic class. **Note** When an ATM VC aggregates a number of sessions with queueing policies, a queueing policy at an ATM VC level must be a one-level policy map that is configured as class-default.
Step 5	**shape average** {*cir*| **percent***percentage*} **Example:** `Router(config-pmap-c)# shape average 10000000`	Specifies average-rate traffic shaping for all traffic that does not match any other traffic class. • Enter the **average** keyword followed by the CIR, in bps or enter the **average** keyword followed by **percentage** keyword to specify a percentage of the interface bandwidth for the CIR. Valid values are from 1 to 100. **Note** When an ATM VC aggregates a number of sessions with queueing policies, a queueing policy at an ATM VC level must be a one-level policy map with only the shape feature enabled.
Step 6	**exit** **Example:** `Router(config-pmap-c)# exit`	Exits policy-map class configuration mode.
Step 7	**exit** **Example:** `Router(config-pmap)# exit`	Exits policy-map configuration mode.
Step 8	interface type slot/subslot/port**.**subinterface **Example:** `Router(config)# interface ATM 3/1/1.1`	Specifies the ATM VC on which you are attaching the policy map and enters ATM VC configuration mode. • Enter the interface type and slot number, subslot number, port number, and ATM VC number.
Step 9	pvc [*name*] *vpi*/*vci* [**ces** | **ilmi** | **qsaal** | **smds**| **l2transport**] **Example:** `Router(config-if-atm-vc)# pvc 2/100`	Selects the ATM VC to which the service policy is to be applied.

	Command or Action	Purpose
Step 10	**vbr-nrt** *peak-cell-rate average-cell-rate* **Example:** `Router(config-if-atm-vc)# vbr-nrt 800000 800000`	Sets the VC type to VBR with a peak and average cell rate.
Step 11	**service-policy output** *policy-map-name* **Example:** `Router(config-subif)# service-policy output subint-1`	Attaches the service policy to the ATM VC. • policy-map-name--The name of the previously configured policy map. **Note** You must specify the output keyword to apply the service policy to outbound traffic on the ATM VC.
Step 12	**end** **Example:** `Router(config-subif)# end`	(Optional) Returns to privileged EXEC mode.

Examples

The following is an example of how to configure and apply a QoS hierarchical queueing policy map to an ATM VC (and provide aggregate shaping for a large number of subscribers):

```
Router> enable
Router# configure terminal
Router(config)# policy-map subint-1
Router(config-pmap)# class class-default
Router(config-pmap-c)# shape average 10000000
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface ATM 3/1/1.1
Router(config-if-atm-vc)# pvc 2/100
Router (config-if-atm-vc)# vbr-nrt 800000 800000
Router(config-subif)# service-policy output subint-1
Router(config-subif)# end
```

Displaying Policy-Map Information for Hierarchical Queueing

SUMMARY STEPS

1. **enable**
2. **show policy-map**
3. **show policy-map interface** *type number*
4. **show policy-map session**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show policy-map** **Example:** `Router# show policy-map`	(Optional) Displays all information for all class maps.

Step 3	**show policy-map interface** *type number*
	
	Example:
	`Router#`
	`show policy-map interface ATM 4/0/0.1`
		(Optional) Displays the packet statistics of all classes that are configured for all service policies either on the specified interface or ATM VC or on a specific PVC on the interface.
		• Enter the interface type and number.

Step 4	**show policy-map session**
	
	Example:
	`Router#`
	`show policy-map session`
		(Optional) Displays the QoS policy map in effect for the SSS session.

Step 5	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

Configuration Examples for QoS Hierarchical Queueing for ATM DSLAMs

- Example Policy Maps on Sessions, page 83
- Example Policy Maps on Sessions with Aggregate Shaping, page 83

Example Policy Maps on Sessions

The following example shows how to configure and apply QoS hierarchical queueing policy maps on sessions. A child queueing policy is applied to each parent subscriber line level policy.

```
Router> enable
Router# configure terminal
Router(config)# policy-map service-a-out
Router(config-pmap)# class voip
Router(config-pmap-c)# priority
Router(config-pmap-c)# set cos 1
Router(config-pmap-c)# exit
Router(config-pmap)# class video
Router(config-pmap-c)# set cos 2
Router(config-pmap-c)# exit
Router(config-pmap)# class gaming
Router(config-pmap-c)# bandwidth remaining percent 80
Router(config-pmap-c)# set cos 3
Router(config-pmap-c)# exit
Router(config-pmap)# class class-default
Router(config-pmap-c)# bandwidth remaining percent 20
Router(config-pmap-c)# set cos 4
Router(config-pmap-c)# exit
Router(config-pmap)# exit
!
Router(config)# policy-map rate-1-service-a-out
Router(config-pmap)# class class-default
Router(config-pmap-c)# bandwidth remaining ratio 10
Router(config-pmap-c)# shape average 100000
Router(config-pmap-c)# service-policy service-a-out
Router(config-pmap-c)# exit
Router(config-pmap)# exit
!
Router(config)# policy-map rate-1-service-a-in
Router(config-pmap)# class voip
Router(config-pmap-c)# police percent 25
Router(config-pmap-c)# exit
Router(config-pmap)# class gaming
Router(config-pmap-c)# police percent 50
Router(config-pmap-c)# exit
Router(config-pmap)# class class-default
Router(config-pmap-c)# police percent 20
Router(config-pmap-c)# exit
Router(config-pmap)# exit
!
Router(config)# interface virtual-template 20
Router(config-if)# service-policy output rate-1-service-a-out
Router(config-if)# service-policy input rate-1-service-a-in
Router(config-if)# end
```

Example Policy Maps on Sessions with Aggregate Shaping

The following example shows how to configure and apply QoS hierarchical queueing policy maps on sessions with multiple PPP/IP sessions per subscriber line. In this example, queueing is configured as in previous example. The VC is configured as follows:

```
Router(config)# policy-map isp_A_out
Router(config-pmap)# class class-default
Router(config-pmap-c)# shape average 500000
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface ATM 1/0/0.1
Router(config-subif)# pvc 10/100
Router(config-if-atm-vc)# vbr-nrt 800000 800000
Router(config-if-atm-vc)# service-policy output isp-A-out
Router(config-if-atm-vc)# exit
Router(config-subif)# exit
```

Additional References

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
Traffic shaping	"Regulating Traffic Flow Using Traffic Shaping" module
MQC	"Applying QoS Features Using the MQC" module

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE Software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for QoS Hierarchical Queueing for ATM DSLAMs

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 6 Feature Information for QoS Hierarchical Queueing for ATM DSLAMs

Feature Name	Releases	Feature Information
QoS Hierarchical Queueing for ATM DSLAMs	Cisco IOS XE Release 2.4 Cisco IOS XE Release 2.5	This feature module describes how to configure QoS hierarchical queueing policy maps on sessions and ATM VCs in ATM Digital Subscriber Line Access Multiplexer (A-DSLAM) applications. This feature was implemented on Cisco ASR 1000 Series Aggregation Services Routers.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.