CHAPTER 2

Planning for Quality of Service

Effective use of Quality of Service (QoS) capabilities requires careful planning. Before you deploy QoS to your network, consider the types of applications used in the network and which QoS techniques might improve the performance of those applications. Then, use CiscoWorks QoS Policy Manager (QPM) to create and deploy your QoS policies to the network, and analyze QoS performance.

The following topics introduce you to QoS concepts and QoS capabilities supported by QPM:

Planning for QoS Deployment

The process of planning and implementing QoS is cyclical, and requires the following steps:

Step 1 Identify the application traffic in your network:

a. Identify the applications in the network, and the traffic distribution of these applications.

For example, you might identify the following applications—SAP (10% of traffic), HTTP (30%), FTP (20%), Voice over IP (VoIP) (30%), and other applications (10%).

b. Identify the business critical applications. In our example these might be SAP, Intranet HTTP, and VoIP.

c. Evaluate the resources requirements for each application, for example, whether the application requires bulk traffic transfers, or streaming, and so on. For VoIP, calculate the bandwidth requirements.

Based on these calculations, decide what level of service each application requires.

Use QPM monitoring to baseline profile your traffic. See Performing Baseline QoS Analysis, page 10-4.

Step 2 Analyze your network:

a. Verify the capacity of your network devices (CPU, software, and so on).

b. Verify the capacity of your network links (link speeds, overhead, and so on).

c. Decide whether domain boundaries are trusted or untrusted. You can then decide where to classify traffic.

d. Analyze the network topology and traffic flow.

e. Analyze the network links in each layer of your network, and the possible QoS mechanisms that can be implemented on those links.

   In converged networks, QPM's IP telephony wizard guides you through the definition of your network topology, and configures QoS on the relevant network points.

Step 3 Use QPM to configure QoS policies throughout the network:

a. Define policies to mark traffic at the edge of the network.

b. Define policies based on markings for each network site.

c. Define policies for aggregate traffic at the WAN edge.

Step 4 Use QPM to deploy your QoS policies:

a. Test deployment in the laboratory, or in a small section of the network.

b. In the network, deploy policies incrementally—start deployment at the edge, and continue towards the core.

Step 5 Use QPM to monitor QoS, and adjust QoS policies:

a. Check whether you achieved the desired QoS by measuring transmitted traffic and dropped traffic for different traffic classes.

b. Monitor application performance.

c. Adjust QoS policies where necessary, and redeploy.

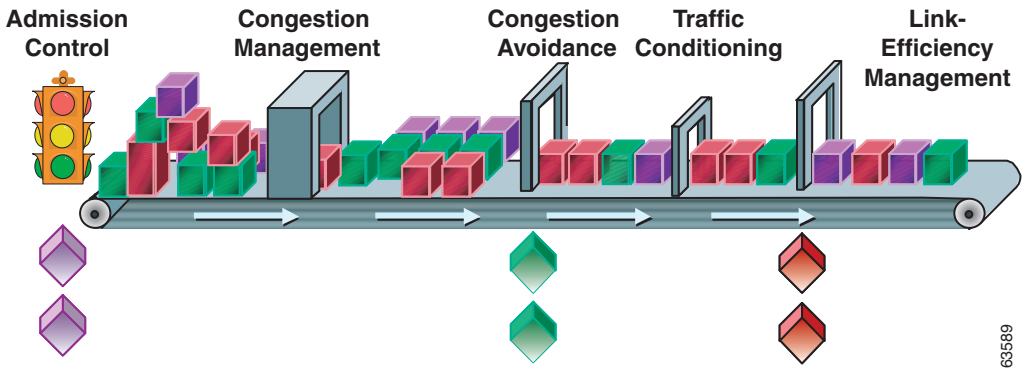
What Types of Quality of Service Does QPM Handle?

QPM detects the QoS capabilities that are available on each of your devices, as defined by the device model, interface type, and the software version running on the device.

You can choose different QoS techniques for different interfaces, as appropriate, to implement your overall networking policies.

Figure 2-1 shows the sequence a packet follows, and the QoS capabilities that might be activated when the packet reaches an interface.

Figure 2-1 QoS Capabilities for a Packet at an Interface



QPM policies let you define the following:

- Classification

  Packet classification identifies traffic flows according to IP precedence or Diff-Serv Code Point (DSCP) values.

  - You can classify traffic for conditional policies by identifying traffic flows according to their classification, source, destination, or application.
  - You can classify traffic by defining marking policies. Marking is generally applied to inbound traffic on its first interface.

- Traffic Conditioning
  - Policing

    The rate of traffic allowed to enter or exit an interface. On routers, you can police aggregate flows. On Catalyst switches, you can police single flows or aggregate flows. Out-of-profile traffic is discarded or its precedence value is marked down.

  - Shaping

    How to smooth the rate of traffic. Shaping can only be defined on outbound interfaces.

- Congestion Management and Congestion Avoidance

  Scheduling and drop preferences to be applied to packets for congestion management and avoidance.

  Some queuing methods queue packets according to their ToS values; for other queuing methods you might need to specify queuing priorities.

- Link-Efficiency Management

  Traffic control mechanisms for management of voice traffic, such as (Compressed Real-time Protocol) CRTP and Link Fragmentation and Interleaving (LFI), and Frame Relay Fragmentation (FRF). They are used in the WAN for voice traffic.

The QoS features supported by any specific device depends on the device type and OS software version. For information about the devices and software versions that are supported by QPM, see the device support tables at:

http://www.cisco.com/en/US/products/sw/cscowork/ps2064/products_device_support_tables_list.html

The following topics describe the types of QoS capabilities you can implement with QPM:

- Packet Marking, page 2-4
- Traffic Policing for Limiting Bandwidth and Marking Traffic, page 2-5
- Traffic Shaping for Controlling Bandwidth, page 2-6
- Queuing Techniques for Congestion Management for Outbound Traffic, page 2-7
- Queuing Techniques for Congestion Avoidance on Outbound Traffic, page 2-14
- Management of Voice and Other Real-Time Traffic, page 2-16
- Managing Traffic Through Access Control, page 2-18
- Signaling Techniques, page 2-18

Related Topics

- More Information About Quality of Service, page 2-20

Packet Marking

Packet marking (also known as classification, or coloring) is used to partition network traffic into multiple priority levels, or classes of service.

Since the classification value is embedded in the packet, changing it can affect the way the packet is handled on its entire path through the network.

In QPM, you can define marking policies to mark packets, using the following techniques:

- IP precedence value, or DiffServ Code Point (DSCP) value

  Sets the IP precedence bits, or the IP differentiated services code point (DSCP) in the IP type of service (ToS) byte.

- Class of Service (CoS) value

  Sets the layer 2 CoS value. This value can be used by layer 2 devices to determine the packet classification.

- MultiProtocol Label Switching (MPLS) Experimental value

  Specifies the CoS for an MPLS packet; the IP packet's CoS is not changed as the packet travels through the MPLS network.

- Frame Relay Discard Eligibility (DE) Bit value

  Determines the priority of a frame in a congested frame relay network. Frames with the DE bit set to 1 are dropped before frames with the DE bit set to 0.

- Trust state

  The trust state of a port determines how it marks received traffic from connected devices. All frames received through untrusted ports are marked with the port CoS value (the default is zero).

  Frames received through trusted ports retain their CoS or ToS values. Trust extension values let you extend the trust boundary beyond the connected device.

QPM implements marking policies using policy-based routing (PBR), or modular QoS CLI (MQC) marking, depending on the device type and OS software version.

Packet marking can also be defined as part of a policing policy. In these cases, QPM uses Committed Access Rate (CAR) or MQC policing. See Traffic Policing for Limiting Bandwidth and Marking Traffic, page 2-5 for more information.

After you mark packets, you can define queuing, shaping, and policing policies that classify by marking value to create differentiated services.

Note Some scheduling methods automatically prioritize traffic according to packet marking, and you do not need to deploy specific queuing policies for interfaces using those queuing methods.

Related Topics

- Traffic Policing for Limiting Bandwidth and Marking Traffic, page 2-5

Traffic Policing for Limiting Bandwidth and Marking Traffic

Traffic policing allows you to control the rate of traffic sent or received on an interface. Traffic policing is often configured on interfaces at the edge of a network to limit traffic into or out of the network.

You can specify one of the following actions for traffic that conforms to or exceeds the specified rate (depending on the type of policing):

- Transmit—The packet is sent.
- Drop—The packet is discarded.
- Mark and transmit—The ToS bits in the packet header are rewritten. The packet is then sent.
- Markdown—This reduces the packets' IP precedence or DSCP values according to a predefined markdown mapping table.

If you define a policing policy for inbound traffic, you can throttle misbehaving traffic before it gets into your network. Since you control the traffic's rate at the inbound interface, the traffic should be well-behaved while it is in your network.

Since rate limiting does not smooth or shape traffic, it does not buffer packets, and therefore, unlike shaping, it does not add any delay to transmission of packets that conform to the rate limits.

The traffic policing feature works with a token bucket mechanism. For a description of a single token bucket algorithm, see:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/qos_c/qcpart4/qcpolts.htm

In QPM, you can define the following types of policing policies:

- Microflow policing—QoS applies the specified bandwidth limit separately to each flow in matched traffic.
- Policing for aggregate flows on the same interface—QoS applies the specified bandwidth limit to all matched traffic on the interface.
- Policing for cross-interface aggregate flows—QoS applies the specified bandwidth limit to matched traffic from all the interfaces in the device group.

QPM supports the following policing techniques:

- CAR Policing—Uses a single token bucket algorithm based on the following three parameters:
  - Average rate—The average rate determines the long-term average transmission rate. Traffic that falls under this rate will always conform.
  - Normal burst size—The normal burst size determines how large traffic bursts can be before some traffic exceeds the rate limit.
  - Excess burst size—The excess burst (Be) size determines how large traffic bursts can be before all traffic exceeds the rate limit. CAR provides managed discard between the excess burst and extended excess burst parameters. Traffic that falls between the normal burst size and the excess burst size exceeds the rate limit with a probability that increases as the burst size increases.
- MQC Policing—Uses a single token bucket algorithm and a two token bucket algorithm. A single token bucket system is used when the violate action option is not specified, and a two token bucket system is used when the violate action option is specified.

- Two-rate policing on Catalyst switches—Uses a two token bucket algorithm with a normal rate and an excess rate:
  - Normal rate—Packets exceeding this rate are marked down.
  - Excess rate—Packets exceeding this rate are either marked down or dropped as specified by the violate action.

Note You can create policies to mark packets without the traffic policing feature, using packet marking policies. See Packet Marking, page 2-4 for more information.

Related Topics

- Packet Marking, page 2-4
- Traffic Shaping for Controlling Bandwidth, page 2-6

Traffic Shaping for Controlling Bandwidth

Traffic shaping controls how much of the interface's bandwidth should be allocated to traffic flows.

Traffic shaping attempts to smooth the traffic flow to meet your rate requirements by buffering the packets. This puts a cap on the bandwidth available to that traffic, ensuring that the remainder of the interface's bandwidth is available to other kinds of traffic.

Traffic shaping affects flows even during times of little congestion, and because it buffers the packets, adds some delay to transmission time.

You set a target average transmission rate for traffic. You can also define a burst size and an exceed burst size to further model the flow. These values define how much data is sent from the buffer per time interval. When the buffer is full, packets are dropped.

Some types of shaping support two types of shaping commands: average and peak.

- When shape average is configured, the interface sends no more than the committed burst (Bc) in each interval.
- When shape peak is configured, the interface sends the committed burst (Bc) plus the excess burst (Be) bits in each interval.

In a link layer network such as Frame Relay, the network sends messages with the forward explicit congestion notification (FECN) or backwards explicit congestion notification (BECN), if there is congestion.

With some shaping methods, the traffic shaping adaptive mode takes advantage of these signals and adjusts the traffic descriptors. This approximates the rate to the available bandwidth along the path.

In QPM, you can configure the following types of shaping:

- Generic traffic shaping (GTS)

  GTS shapes traffic by reducing outbound traffic flow to avoid congestion by constraining traffic to a particular bit rate using the token bucket mechanism.

- Frame relay traffic shaping (FRTS)

  Lets you specify an average bandwidth size for Frame Relay virtual circuits (VC), defining an average rate commitment for the VC.

  FRTS uses a buffer to hold packets while it transmits the flow at the specified committed information rate (CIR). You can also define a burst size and an exceed burst size to further model the flow.

These values define how much data FRTS can send from the buffer per time interval. After the buffer is full, packets are dropped.

FRTS supports adaptive shaping. When congestion occurs, the default minimum CIR (minCIR) is used, which is half of the CIR. QPM allows you to override this default by specifying a minimum rate to be used when there is congestion.

- Distributed traffic shaping (DTS)

  DTS supports all functionality provided by both GTS and FRTS. DTS uses queues to buffer traffic surges that can congest a network. Data is buffered and then sent into the network at a regulated rate.

  This ensures that traffic will behave according to the configured descriptor, as defined by the Committed Information Rate (CIR), Committed Burst (Bc), and Excess Burst (Be). DTS provides two types of shape commands—average and peak. DTS supports adaptive shaping.

- Modular shaping

  Operates on all traffic flows on an interface. This type of shaping uses DTS on versatile interface processor (VIP) interfaces, or GTS on other types of interfaces. Modular traffic shaping can be used on VIP interfaces on devices that do not support FRTS.

Related Topics

- Traffic Policing for Limiting Bandwidth and Marking Traffic, page 2-5

Queuing Techniques for Congestion Management for Outbound Traffic

You can set a queuing technique on a device's interface to manage how packets are queued to be sent through the interface. Some queuing techniques use the packet marking, while others ignore them.

Queuing techniques are primarily used for managing traffic congestion on an interface. That is, they determine the priority in which to send packets when there is more data than can be sent immediately:

Class-Based QoS Queuing: Multiple-Action, Class-Based Policies

On devices with IOS software versions that support modular QoS CLI (MQC), you can create multiple-action, class-based QoS policies, including a class-based QoS queuing policy. If class-based QoS is available for an interface, Cisco recommends that you use it instead of other scheduling methods.

Class-based QoS queuing uses WFQ processing to give higher weight to high priority traffic, but derives that weight from classes that you create.

These classes are similar to custom queues. They are policy-based, identify traffic based on the traffic's characteristics (protocol, source, destination, and so forth), and allocate a percentage of the interface's bandwidth to the traffic flow.

With class-based QoS queuing, you can create up to 64 classes for an interface. (Unlike WFQ, queues are not automatically based on the packet's ToS value.) Class-based QoS queuing also lets you control the drop mechanism used when congestion occurs on the interface.

You can use WRED for the drop mechanism, and configure the WRED queues, to ensure that high-priority packets within a class are given the appropriate weight. If you use tail drop, all packets within a class are treated equally, even if the ToS values are not equal.

An effective use of class-based QoS queuing would be to guarantee bandwidth to a few critical applications to ensure reliable application performance.

The queues you define constitute a minimum bandwidth allocation for the specified flow. If more bandwidth is available on the interface due to a light load, a queue can use the extra bandwidth. This is handled dynamically by the device.

Unclassified packets that do not match any traffic classifiers defined for class-based policies are processed according to the settings in the default class. The default behavior for unclassified traffic is weighted fair queuing.

- If you use WRED as the drop mechanism for a class, WRED automatically considers the packet's ToS value when determining which packet to drop. Tail drop does not consider a packet's ToS value.
- If you use WFQ on the default class policy, WFQ automatically considers the packet's ToS value when queuing, dropping, and sending packets in the default queues.

Class-based QoS can be used with additional QoS capabilities to enable efficient management of voice and other real-time traffic.

- Traffic Shaping for Controlling Bandwidth, page 2-6
- Low Latency Queuing (LLQ): Strict Priority Queuing, page 2-16
- IP RTP Priority: Providing Strict Priority to Voice Traffic, page 2-17
- Link Fragmentation and Interleaving (LFI): Reducing Delay and Jitter on Lower Speed Links, page 2-17
- Compressed Real-Time Protocol (CRTP): RTP Header Compression to Reduce Delay, page 2-17
- Frame Relay Fragmentation (FRF): Preventing Delay on Frame Relay Links, page 2-18

Related Topics

- Management of Voice and Other Real-Time Traffic, page 2-16
- Distributed Weighted Fair Queuing (DWFQ): High Speed WFQ for VIP Interfaces, page 2-9
- Fair Queuing (FQ): Flow-Based Queuing, page 2-9
- First In, First Out (FIFO) Queuing: Basic Store and Forward on Routers, page 2-11
- Priority Queuing (PQ): Basic Traffic Prioritization on Routers, page 2-10
- Custom Queuing (CQ): Advanced Traffic Prioritization on Routers, page 2-10
- Weighted Fair Queuing (WFQ): Intelligent Traffic Prioritization on Routers, page 2-11

Distributed Weighted Fair Queuing (DWFQ): High Speed WFQ for VIP Interfaces

On devices with IOS software versions that do not support class-based QoS, class-based queuing features are implemented through distributed WFQ (DWFQ).

With DWFQ, packets are assigned to different queues based on their QoS group or the IP precedence in the ToS field. QoS groups allow you to customize your QoS policy.

A QoS group is an internal classification of packets used by the router to determine how packets are treated by certain QoS features, such as DWFQ and committed access rate (CAR).

Like class-based QoS queuing, DWFQ uses WFQ processing to give higher weight to high priority traffic, but derives that weight from classes that you create.

These classes are similar to custom queues. They are policy-based, identify traffic based on the traffic's characteristics (protocol, source, destination, and so forth), and allocate a percentage of the interface's bandwidth to the traffic flow.

An effective use of DWFQ would be to guarantee bandwidth to a few critical applications to ensure reliable application performance.

QPM does not support ToS-based DWFQ.

Related Topics

- Class-Based QoS Queuing: Multiple-Action, Class-Based Policies, page 2-8
- Fair Queuing (FQ): Flow-Based Queuing, page 2-9
- First In, First Out (FIFO) Queuing: Basic Store and Forward on Routers, page 2-11
- Priority Queuing (PQ): Basic Traffic Prioritization on Routers, page 2-10
- Custom Queuing (CQ): Advanced Traffic Prioritization on Routers, page 2-10
- Weighted Fair Queuing (WFQ): Intelligent Traffic Prioritization on Routers, page 2-11

Fair Queuing (FQ): Flow-Based Queuing

Fair queuing gives all packets an equal weight, and all queues are allocated equal bandwidth.

With FQ, packets are classified by flow. Packets with the same source IP address, destination IP address, source TCP or User Datagram Protocol (UDP) port, destination TCP or UDP port, protocol, and ToS field belong to the same flow. (All non-IP packets are treated as flow 0.)

Each flow corresponds to a separate output queue. When a packet is assigned to a flow, it is placed in the queue for that flow.

Related Topics

- Class-Based QoS Queuing: Multiple-Action, Class-Based Policies, page 2-8
- Distributed Weighted Fair Queuing (DWFQ): High Speed WFQ for VIP Interfaces, page 2-9
- First In, First Out (FIFO) Queuing: Basic Store and Forward on Routers, page 2-11
- Priority Queuing (PQ): Basic Traffic Prioritization on Routers, page 2-10

- Custom Queuing (CQ): Advanced Traffic Prioritization on Routers, page 2-10
- Weighted Fair Queuing (WFQ): Intelligent Traffic Prioritization on Routers, page 2-11

Priority Queuing (PQ): Basic Traffic Prioritization on Routers

Priority queuing (PQ) is a rigid traffic prioritization scheme. If packet A has a higher priority than packet B, packet A always goes through the interface before packet B.

An effective use of priority queuing would be for placing time-critical but low-bandwidth traffic in the high queue. This ensures that this traffic is transmitted immediately, but because of the low-bandwidth requirement, lower queues are unlikely to be starved.

The disadvantage of priority queuing is that the higher queue is given absolute precedence over lower queues. For example, packets in the low queue are only sent when the high, medium, and normal queues are completely empty.

If a queue is always full, the lower-priority queues are never serviced. They fill up and packets are lost. Thus, one particular kind of network traffic can come to dominate a priority queuing interface. Packets that do not match any traffic classifier are placed in the normal queue.

Related Topics

- Class-Based QoS Queuing: Multiple-Action, Class-Based Policies, page 2-8
- Distributed Weighted Fair Queuing (DWFQ): High Speed WFQ for VIP Interfaces, page 2-9
- Fair Queuing (FQ): Flow-Based Queuing, page 2-9
- First In, First Out (FIFO) Queuing: Basic Store and Forward on Routers, page 2-11
- Custom Queuing (CQ): Advanced Traffic Prioritization on Routers, page 2-10
- Weighted Fair Queuing (WFQ): Intelligent Traffic Prioritization on Routers, page 2-11

Custom Queuing (CQ): Advanced Traffic Prioritization on Routers

Custom queuing (CQ) is a flexible traffic prioritization scheme that allocates a minimum bandwidth to specified types of traffic. You can create up to 16 of these custom queues.

For custom queue interfaces, the device services the queues in a round robin fashion, sending out packets from a queue until the byte count on the queue is met, then moving on to the next queue. This ensures that no queue gets starved, in comparison to priority queuing.

An effective use of custom queuing would be to guarantee bandwidth to a few critical applications to ensure reliable application performance.

The custom queues you define constitute a minimum bandwidth allocation for the specified flow. If more bandwidth is available on the interface because of a light load, a queue can use the extra bandwidth. This is handled dynamically by the device.

If you do not create queuing policies for a Custom Queuing interface, all traffic is placed in a single queue (the default queue), and is processed first in, first out, in the same manner as a FIFO queuing interface.

Related Topics

- Class-Based QoS Queuing: Multiple-Action, Class-Based Policies, page 2-8
- Distributed Weighted Fair Queuing (DWFQ): High Speed WFQ for VIP Interfaces, page 2-9
- Fair Queuing (FQ): Flow-Based Queuing, page 2-9

- First In, First Out (FIFO) Queuing: Basic Store and Forward on Routers, page 2-11
- Priority Queuing (PQ): Basic Traffic Prioritization on Routers, page 2-10
- Weighted Fair Queuing (WFQ): Intelligent Traffic Prioritization on Routers, page 2-11

Weighted Fair Queuing (WFQ): Intelligent Traffic Prioritization on Routers

Weighted fair queuing (WFQ) acknowledges and uses a packet's priority without starving low-priority packets for bandwidth.

Weighted fair queuing divides packets into two classes:

- Interactive traffic is placed at the front of the queue to reduce response time
- Noninteractive traffic shares the remaining bandwidth proportionately

Since interactive traffic is typically low-bandwidth, its higher priority does not starve the remaining traffic. A complex algorithm is used to determine the amount of bandwidth assigned to each traffic flow. Packet marking is considered when making this determination.

Weighted fair queuing is very efficient and requires little configuration. To implement weighted fair queuing, you define Weighted Fair Queuing for the interface.

You do not need to define queuing policies because WFQ automatically prioritizes the packets according to their IP precedence or DSCP value.

When you apply WFQ automatically, consider marking all traffic that enters the device (or mark the traffic at the point where it enters your network, to ensure that packets receive the service level you intend. Otherwise, the originator of the traffic, or another network device along the traffic's path, determines the service level for the traffic.

Related Topics

- Class-Based QoS Queuing: Multiple-Action, Class-Based Policies, page 2-8
- Distributed Weighted Fair Queuing (DWFQ): High Speed WFQ for VIP Interfaces, page 2-9
- Fair Queuing (FQ): Flow-Based Queuing, page 2-9
- First In, First Out (FIFO) Queuing: Basic Store and Forward on Routers, page 2-11
- Priority Queuing (PQ): Basic Traffic Prioritization on Routers, page 2-10
- Custom Queuing (CQ): Advanced Traffic Prioritization on Routers, page 2-10

First In, First Out (FIFO) Queuing: Basic Store and Forward on Routers

First In, First Out (FIFO) queuing is the basic queuing technique. In FIFO queuing, packets are queued on a first come, first served basis: if packet A arrives at the interface before packet B, packet A leaves the interface before packet B. This is true even if packet B has a higher IP precedence than packet A since FIFO queuing ignores packet characteristics.

FIFO queuing works well on uncongested high-capacity interfaces that have minimal delay, or when you do not want to differentiate services for packets traveling through the device.

The disadvantage of FIFO queuing is that when a station starts a file transfer, it can consume all the bandwidth of a link to the detriment of interactive sessions. This phenomenon is referred to as a *packet train* because one source sends a "train" of packets to its destination and packets from other stations get caught behind the train.

You do not need to define any queuing parameters for FIFO queuing.

Related Topics

- Class-Based QoS Queuing: Multiple-Action, Class-Based Policies, page 2-8
- Distributed Weighted Fair Queuing (DWFQ): High Speed WFQ for VIP Interfaces, page 2-9
- Fair Queuing (FQ): Flow-Based Queuing, page 2-9
- Priority Queuing (PQ): Basic Traffic Prioritization on Routers, page 2-10
- Custom Queuing (CQ): Advanced Traffic Prioritization on Routers, page 2-10
- Weighted Fair Queuing (WFQ): Intelligent Traffic Prioritization on Routers, page 2-11

Weighted Round Robin (WRR): Managing Layer 3 Switch Congestion

Weighted round robin (WRR) scheduling is used on layer 3 switches. WRR queuing is handled differently on the Catalyst 8500 family and on other layer 3 switches.

Weighted round robin (WRR) scheduling is used automatically on layer 3 switches on egress ports to manage the queuing and sending of packets.

WRR places a packet in one of four queues based on the packet's IP precedence, from which it derives a delay priority.

Table 2-1 shows the queue assignments based on the IP precedence value and derived delay priority of the packet, and the weight of the queue if you do not change it.

Table 2-1 WRR Queue Packet Assignments

IP Precedence	Delay Priority	Queue Assignment	Default Queue Weight (Catalyst 8500)	Default Queue Weight (Other Layer 3 Switches)
0, 1	0	0	1	1
2, 3	1	1	2	2
4, 5	2	2[1]	4	3
6, 7	3	3	8	4

1. Queue 2 is the queue typically used for voice traffic.

With WRR, each queue is given a weight. This weight is used when congestion occurs on the port to give weighted priority to high-priority traffic without starving low priority traffic.

The weights provide the queues with an implied bandwidth for the traffic on the queue. The higher the weight, the greater the implied bandwidth. However, the queues are not assigned specific bandwidth and when the port is not congested, all queues are treated equally.

Devices that use WRR automatically create the four queues with default weights for each interface. You need only define policies if you want to change the queue weights. For the Catalyst 8500, these policies are assigned to the device. For other layer 3 switches, policies are assigned to the destination ports.

Managing Congestion on Switch Ports

Queuing methods on switch ports use a packet's precedence setting to determine how that packet is serviced on the port. The queuing methods use multiple queues of different priority, with one or more thresholds for each queue, to determine the bandwidth allowed for traffic based on each Class of Service (CoS) value.

These queues and thresholds are serviced using weighted round robin (WRR) techniques to ensure a fair chance of transmission to each class of traffic. These queuing methods favor high-priority traffic without starving low-priority traffic.

QPM supports the following queuing methods for switch ports:

- 2 Queues, 2 Thresholds (2Q2T), page 2-13
- 1 Priority Queue, and 2 Queues 2 Thresholds (1P2Q2T), page 2-13
- 2 Queues, 1 Threshold (2Q1T), page 2-13
- 4 Queues, I Threshold (4Q1T), page 2-14
- 4 Queues, 1 Threshold (4Q1T) Shape, page 2-14
- 4 Queues, 2 Thresholds (4Q2T), page 2-14

2 Queues, 2 Thresholds (2Q2T)

2Q2T queuing uses two queues, one high priority, the other low priority, with two thresholds for each queue, to determine the bandwidth allowed for traffic based on each Class of Service (CoS) value. 2Q2T assigns each precedence to a specific queue and threshold on that queue.

For example, packets with CoS value of 0 (the lowest priority) are placed in the low priority queue and use the lower threshold by default. This ensures that the least important traffic gets less service than any other traffic.

2Q2T queuing comes with a default configuration for the queues, thresholds, and traffic assignments based on CoS settings.

You can change this configuration if it does not suit your requirements. You can change the size of the queues, their relative WRR weights, the sizes of their thresholds, and the assignment of precedence values to the appropriate queue and threshold. You do not need to define queuing policies for ports with 2Q2T queuing.

1 Priority Queue, and 2 Queues 2 Thresholds (1P2Q2T)

1P2Q2T queuing uses three queues:

- One strict priority queue, usually used for voice traffic
- One high priority queue with two thresholds
- One low priority queue with two thresholds

1P2Q2T assigns each precedence to a specific queue and threshold on that queue.

You can mark voice traffic so that it will be assigned to the strict priority queue. On 1P2Q2T interfaces, the switch services traffic in the strict priority queue before servicing the standard queues.

When the switch is servicing a standard queue, after transmitting a packet, it checks for traffic in the strict priority queue. If the switch detects traffic in the strict priority queue, it suspends its service of the standard queue and completes service of all traffic in the strict priority queue before returning to the standard queue.

2 Queues, 1 Threshold (2Q1T)

2Q1T queuing uses two queues, with one threshold for each queue. Each pair of CoS values is associated with either queue 1 or queue 2. For each pair of CoS values, you can define the queue to which packets with those CoS values will be directed.

4 Queues, I Threshold (4Q1T)

4Q1T queuing uses four queues, with one threshold for each queue, to determine the bandwidth allowed for traffic based on each Class of Service (CoS) value.

4Q1T queuing comes with a default configuration for the queues, and traffic assignments based on CoS settings. You can change this configuration if it does not suit your requirements.

You can change the relative WRR weights of the queues, and the assignment of precedence values to the appropriate queue and threshold. You do not need to define queuing policies for ports with 4Q1T queuing.

4 Queues, 1 Threshold (4Q1T) Shape

4Q1T Shape queuing uses four queues, with one threshold for each queue, to determine the bandwidth allowed for traffic based on each DSCP value. In addition, you can set a shaping rate for each queue to specify the maximum bandwidth allowed for the queue.

4Q1T Shape queuing comes with a default configuration for the queues, and traffic assignments based on DSCP settings. By default, each queue gets 25% of the interface's bandwidth, and the queues are serviced round-robin based on the bandwidth and shaping rates.

You can change this configuration if it does not suit your requirements. You can:

- Change the mapping of DSCP values to queues.
- Change the bandwidth for each queue (the queue's minimum rate).
- Configure a shaping rate for each queue (the queue's maximum rate).
- Configure queue 3 as a priority queue, so that it is serviced preferentially over the other queues if its minimum bandwidth is not met. However, if you define a shaping rate for queue 3, its rate is limited by the shaping rate, unless there is excess bandwidth available.

4 Queues, 2 Thresholds (4Q2T)

4Q2T queuing uses four queues, with two thresholds for each queue, to determine the bandwidth allowed for traffic based on each Class of Service (CoS) value. 4Q2T assigns each precedence to a specific queue and threshold on that queue.

4Q2T queuing comes with a default configuration for the queues, thresholds, and traffic assignments based on CoS settings. You can change this configuration if it does not suit your requirements. You can change the size of the queues, their relative WRR weights, the sizes of their thresholds, and the assignment of precedence values to the appropriate queue and threshold.

You can also choose the drop method for each queue. You do not need to define queuing policies for ports with 4Q2T queuing.

You can define Queue 4 as a strict priority queue, which transmits traffic whenever it is detected.

Queuing Techniques for Congestion Avoidance on Outbound Traffic

Weighted random early detection (WRED) is a queuing technique for congestion avoidance, meaning, it manages how packets are handled when an interface starts to be congested.

With WRED, when traffic begins to exceed the interface's traffic thresholds, but before congestion occurs, the interface starts dropping packets from selected flows.

If the dropped packets are TCP, the TCP source recognizes that packets are getting dropped, and lowers its transmission rate. The lowered transmission rate then reduces the traffic to the interface, thus avoiding congestion. Since TCP retransmits dropped packets, no actual data loss occurs.

WRED drops packets according to the following criteria:

- RSVP flows are given precedence over non-RSVP flows, to ensure that time-critical packets are transmitted as required.
- The IP precedence or DSCP value of the packets. Packets with higher priority are less likely to be dropped.

  You can control how WRED determines when and how often to drop packets based on IP precedence or DSCP value if you are not satisfied with the default settings. (DSCP-based WRED is not available on all devices and IOS software versions that support precedence-based WRED.)
- The amount of bandwidth used by the traffic flow. Flows that use the most bandwidth are more likely to have packets dropped.
- The weight factor you have defined for the interface determines how frequently packets are dropped.

WRED chooses the packets to drop after considering these factors in combination, the net result being that the highest priority and lowest bandwidth traffic is preserved.

WRED differs from standard random early detection (RED) in that RED ignores IP precedence, and instead drops packets from all traffic flows, not selecting low precedence or high bandwidth flows.

By selectively dropping packets before congestion occurs, WRED prevents an interface from getting flooded, necessitating a large number of dropped packets. This increases the overall bandwidth usage for the interface.

On devices with a versatile interface processor (VIP), when you configure an interface to use WRED, it automatically uses distributed WRED. Distributed WRED takes advantage of the VIP.

An effective use of weighted random early detection would be to avoid congestion on a predominantly TCP/IP network, one that has minimal UDP traffic and no significant traffic from other networking protocols.

It is especially effective on core devices rather than edge devices, because the traffic marking you perform on edge devices can then affect the WRED interfaces throughout the network.

The disadvantage of WRED is that only predominantly TCP/IP networks can benefit. Other protocols, such as UDP or NetWare (IPX), do not respond to dropped packets by lowering their transmission rates, instead retransmitting the packets at the same rate.

WRED treats all non-TCP/IP packets as having precedence 0. If you have a mixed network, WRED might not be the best choice for queuing traffic.

Weighted random early detection interfaces, favor high priority, low bandwidth traffic flows. No specific policies are needed. However, because WRED automatically uses the IP precedence or DSCP settings in packets, consider marking all traffic that enters the device (or mark the traffic at the point where it enters your network).

By marking all traffic, you can ensure that packets receive the service level you intend. Otherwise, the originator of the traffic, or another network device along the traffic's path, determines the service level for the traffic.

You can also create class-based QoS policies that use WRED as the drop mechanism for the class-based queues.

Related Topics

- Class-Based QoS Queuing: Multiple-Action, Class-Based Policies, page 2-8

Management of Voice and Other Real-Time Traffic

Real-time-based applications, such as voice applications, have different characteristics and requirements from those of other data applications. Voice applications tolerate minimal variation in the amount of delay affecting delivery of their voice packets.

Voice traffic is also intolerant of packet loss and jitter, both of which degrade the quality of the voice transmission delivered to the recipient end user.

To effectively transport voice traffic over IP, mechanisms are required that ensure reliable delivery of packets with low latency.

To simplify the process of defining end-to-end QoS for voice traffic, QPM provides you with a voice application. This includes a wizard, which guides you through the process of defining your IP network topology, and then automatically creates the required QoS configuration at each relevant network point.

QPM also includes predefined IP telephony templates, which contain the QoS configurations and policies required at each relevant point in the network. All you must do is add your devices to the device inventory, assign their interfaces to the relevant policy groups, and deploy.

For detailed information, see Chapter 7, “Provisioning: Configuring QoS for IP Telephony.”

The following features can be used to manage voice traffic:

- Low Latency Queuing (LLQ): Strict Priority Queuing, page 2-16
- IP RTP Priority: Providing Strict Priority to Voice Traffic, page 2-17
- Link Fragmentation and Interleaving (LFI): Reducing Delay and Jitter on Lower Speed Links, page 2-17
- Compressed Real-Time Protocol (CRTP): RTP Header Compression to Reduce Delay, page 2-17
- Frame Relay Fragmentation (FRF): Preventing Delay on Frame Relay Links, page 2-18

On Catalyst switches, the following is available for management of voice traffic:

- 1 Priority Queue, and 2 Queues 2 Thresholds (1P2Q2T), page 2-13
- 4 Queues, 2 Thresholds (4Q2T), page 2-14

Low Latency Queuing (LLQ): Strict Priority Queuing

Low latency queuing (LLQ) is used with class-based QoS for strict priority queuing. Strict priority queuing allows delay-sensitive data such as voice to be dequeued and sent first (before packets in other queues are dequeued), giving delay-sensitive data preferential treatment over other traffic. LLQ is not limited to UDP port numbers, as is IP RTP priority.

Using LLQ reduces delay and jitter in voice conversations. LLQ is enabled when you configure the priority status in a class-based QoS queuing policy. When several types of traffic on an interface are configured as priority classes, all these types of traffic are enqueued to the same, single, strict priority queue.

Related Topics

- Class-Based QoS Queuing: Multiple-Action, Class-Based Policies, page 2-8

IP RTP Priority: Providing Strict Priority to Voice Traffic

IP RTP Priority creates a strict priority queue for real-time transport protocol (RTP) traffic. The IP RTP Priority queue is emptied before other queues are serviced. This is typically used to provide absolute priority to voice traffic, which uses RTP ports.

Since voice traffic is delay-sensitive and low bandwidth, you can typically give it absolute priority without starving other data traffic. This ensures that voice quality is adequate.

IP RTP Priority is especially useful on slow-speed WAN links, including Frame Relay, Multilink PPP (MLP), and T1 ATM links. It works with WFQ and class-based QoS.

For class-based QoS interfaces, you can configure custom class-based queues for other types of traffic. The bandwidth allocated to the IP RTP priority queue counts as part of the total allocated class-based QoS queue bandwidth.

IP RTP priority cannot be configured on the interface when FRTS is enabled. IP RTP priority is not available on VIP cards.

IP RTP Priority ignores compression, treating a compressed 12 kbps flow as a 24 kbps flow.

Related Topics

- Weighted Fair Queuing (WFQ): Intelligent Traffic Prioritization on Routers, page 2-11
- Class-Based QoS Queuing: Multiple-Action, Class-Based Policies, page 2-8

Link Fragmentation and Interleaving (LFI): Reducing Delay and Jitter on Lower Speed Links

Voice over IP is susceptible to increased latency and jitter when the network processes large packets, such as LAN-to-LAN FTP Telnet transfers traversing a WAN link. This susceptibility increases as the traffic is queued on slower links. LFI was designed especially for lower-speed links in which serialization delay is significant.

LFI reduces delay and jitter on slower speed links by breaking up large data packets so that they are small enough to satisfy the delay requirements of real-time traffic. The low-delay traffic packets, such as voice packets, are interleaved with the fragmented packets. LFI also provides a special transmit queue for the smaller, delay-sensitive packets, enabling them to be sent earlier than other flows.

QPM cannot detect or implement MLP and will assume that the multilink PPP command is enabled on the interface. QPM will configure only the interleave and fragmentation commands. When LFI is defined on an interface group, it is only deployed to the interfaces that support it.

Related Topics

- Compressed Real-Time Protocol (CRTP): RTP Header Compression to Reduce Delay, page 2-17

Compressed Real-Time Protocol (CRTP): RTP Header Compression to Reduce Delay

Real-Time Protocol (RTP) is a host-to-host protocol used for carrying multimedia application traffic, including packetized audio and video, over an IP network. RTP provides end-to-end network transport functions intended for applications sending real-time requirements, such as audio and video.

To avoid the unnecessary consumption of available bandwidth, CRTP, the RTP header compression feature, is used on a link-by-link basis. CRTP compresses the IP/UDP/RTP header in an RTP data packet from 40 bytes to approximately 2 to 5 bytes resulting in decreased consumption of available bandwidth for voice traffic. A corresponding reduction in delay is realized.

Related Topics

- Link Fragmentation and Interleaving (LFI): Reducing Delay and Jitter on Lower Speed Links, page 2-17

Frame Relay Fragmentation (FRF): Preventing Delay on Frame Relay Links

Frame Relay fragmentation (FRF) ensures predictability for voice traffic, by aiming to provide better throughput on low-speed Frame Relay links.

FRF allows long data frames on one virtual circuit (VC) to be fragmented into smaller pieces and interleaved with delay-sensitive voice traffic on another VC utilizing the same interface.

In this way, real-time voice and non-real-time data frames can be carried together on lower-speed links without causing excessive delay to the real-time traffic.

VoIP packets should not be fragmented. However, VoIP packets can be interleaved with fragmented packets. If some PVCs are carrying voice traffic, you can enable fragmentation on all PVCs. The fragmentation header is only included for frames that are greater than the fragment size configured.

Managing Traffic Through Access Control

You can control traffic access by permitting or denying transport of packets into or out of interfaces.

You can define access control policies, which will deny or permit traffic that matches the traffic classifier definition in the specified direction. You can also define a traffic classifier condition to deny specific types of traffic as part of a QoS policy definition.

The access control feature can be used as a security feature, and can be enabled or disabled globally for all databases in your system. You can overwrite the global configuration on a per-domain or per-device basis.

You cannot create Access Control policies for the Cisco 8500 family of devices or for Catalyst switches.

Signaling Techniques

To implement end-to-end quality of service, a traffic flow must contain or use some type of signal to identify the requirements of the traffic. With QPM, you can control these types of signaling techniques:

- IP Precedence and DSCP Values: Differentiated Services, page 2-18
- Resource Reservation Protocol (RSVP): Guaranteed Services, page 2-19

IP Precedence and DSCP Values: Differentiated Services

The simplest form of signal is the IP precedence or DSCP setting in data packets: the packet's color or classification.

This signal is carried with the packet, and can affect the packet's handling at each node in the network. Queuing techniques such as WFQ and WRED automatically use this signal to provide differentiated services to high-priority traffic.

To use the IP precedence or DSCP setting effectively, ensure that you mark traffic at the edges of your network so that the marking affects the packet's handling throughout the network. See Packet Marking, page 2-4, for information on how to change a packet's IP precedence or DSCP setting.

IP precedence and DSCP can only provide differentiated services on interfaces that use a queuing technique that is sensitive to the precedence setting in the packet. For example, WFQ, WRED, WRR, 1P2Q2T, and 2Q2T automatically consider the precedence settings.

Related Topics

- Packet Marking, page 2-4
- Queuing Techniques for Congestion Management for Outbound Traffic, page 2-7
- Queuing Techniques for Congestion Avoidance on Outbound Traffic, page 2-14

Resource Reservation Protocol (RSVP): Guaranteed Services

A more sophisticated form of signaling than IP precedence is the resource reservation protocol (RSVP). RSVP is used by applications to dynamically request specific bandwidth resources from each device along the traffic flow's route to its destinations.

After the reservations are made, the application can start the traffic flow with the assurance that the required resources are available.

RSVP is mainly used by applications that produce real-time traffic, such as voice, video, and audio. Unlike standard data traffic, such as HTTP, FTP, or Telnet, real-time applications are delay sensitive, and can become unusable if too many packets are dropped from a traffic flow. RSVP helps the application ensure there is sufficient bandwidth so that jitter, delay, and packet drop can be avoided.

RSVP is typically used by multicast applications. With multicasting, an application sends a stream of traffic to several destinations. For example, the Cisco IP/TV application can provide several audio-video programs to users. If a user accesses one of the provided programs, IP/TV sends a stream of video and audio to the user's computer.

Network devices consolidate multicast traffic to reduce bandwidth usage. Thus, if there are ten users for a traffic flow behind a router, the router sees one traffic flow, not ten. In unicast traffic, the router sees ten traffic flows. Although RSVP can work with unicast traffic (one sender, one destination), RSVP unicast flows can quickly use up RSVP resources on the network devices if a lot of users access unicast applications. In other words, unicast traffic scales poorly.

To configure RSVP on network devices, you must determine the bandwidth requirements of the RSVP-enabled applications on your network.

If you do not configure the devices to allow RSVP to reserve enough bandwidth, the applications will perform poorly. See the documentation for the applications to determine their bandwidth requirements.

When an RSVP request is made, RSVP calculates the bandwidth request by considering the mean data rate, the amount of data the interface can hold in the queue, and the minimum QoS requirement for the traffic flow. The interface determines if it can meet the request, and replies to the requesting application.

When the traffic flow begins, RSVP can dynamically respond to changes in routes, switching reservations to new devices and releasing reservations for devices no longer on the path. After the flow is complete, all reservations are removed and the bandwidth on the interfaces released.

RSVP with WFQ or class-based QoS provides guaranteed rate service, providing an absolute rate even during congestion events. This is good for delay-sensitive real-time applications like voice over IP.

RSVP with WRED provides controlled load service, providing low delay and high throughput during congestion events. This is good for adaptive real-time applications such as the playback of a recorded conference call. With WRED advanced properties, you can control the WRED thresholds for RSVP traffic.

Related Topics

- Weighted Fair Queuing (WFQ): Intelligent Traffic Prioritization on Routers, page 2-11
- Class-Based QoS Queuing: Multiple-Action, Class-Based Policies, page 2-8
- Queuing Techniques for Congestion Avoidance on Outbound Traffic, page 2-14

More Information About Quality of Service

This publication cannot cover everything you might want to know about quality of service. This section provides pointers to more information available on the web.

For pages that require a Cisco.com login, you can register at the Cisco.com web site at:

https://tools.cisco.com/RPF/register/register.do

The references are broken down into the following categories:

General QoS Information

- Quality of Service (Internetworking Technology Overview)—Detailed overview of QoS capabilities:

  http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/qos.htm

Voice over IP Information

- Cisco AVVID Network Infrastructure Enterprise Quality of Service Design Guide—Provides a blueprint for implementing the end-to-end Quality of Service (QoS) that is required for successful deployment of Cisco AVVID solutions in today's enterprise environment:

  www.cisco.com/warp/customer/771/srnd/qos_srnd.pdf

- Quality of Service for Voice over IP—Information on QoS methods for voice over IP:

  http://www.cisco.com/univercd/cc/td/doc/cisintwk/intsolns/qossol/qosvoip.htm

IOS Software Release 12.x Documentation

- QoS Solutions Configuration Guide—Includes information on QoS mechanisms supported by IOS 12.2:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fqos_c/index.htm

- QoS Solutions Configuration Reference—Commands for configuring QoS:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fqos_r/index.htm

- Wide-Area Networking Configuration Guide—Includes information on FRTS:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fwan_c/index.htm

- New Features in IOS 12.2—Includes information about the latest QoS methods supported by IOS 12.2:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/cgft14.htm

IOS Software Release 11.1cc Documentation

- Committed Access Rate (CAR)—For release 11.1cc:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios111/cc111/car.htm

- Distributed WRED—For release 11.1cc:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios111/cc111/wred.htm

- Distributed WFQ—For release 11.1cc:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios111/cc111/dwfq.htm

Catalyst Documentation

- Catalyst 8500 Quality of Service Feature Summary—Information on WRR:

  http://www.cisco.com/en/US/products/hw/switches/ps718/products_feature_guide_chapter09186a00800ef93c.html

- Catalyst 4908 Quality of Service Feature Summary—Information on WRR:

  http://www.cisco.com/univercd/cc/td/doc/product/l3sw/4908g_l3/ios_12/7w515d/config/qos_cnfg.htm

- Configuring Quality of Service (Software Configuration Guide)—For the Catalyst 5000 family, software release 6.3. Information on Catalyst classification:

  http://www.cisco.com/en/US/docs/switches/lan/catalyst5000/catos/6.x/configuration/guide/qos.html

- Configuring Quality of Service (Software Configuration Guide)—For the Catalyst 6000 family, software release 6.3. Information on Catalyst classification, policing, and queuing methods:

  http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_6_3/confg_gd/qos.htm

- Configuring Quality of Service—For the Catalyst 2950:

  http://www.cisco.com/univercd/cc/td/doc/product/lan/cat2950/1216ea2b/scg/swgqos.htm

- Configuring Quality of Service—For the Catalyst 3550:

  http://www.cisco.com/univercd/cc/td/doc/product/lan/c3550/1219ea1/3550scg/swqos.htm