CCNA 3 v7.0 Curriculum: Module 9 – QoS Concepts

itexamanswers.net/ccna-3-v7-0-curriculum-module-9-qos-concepts.html

April 13, 2020

Contents

9.0. Introduction

9.0.1. Why should I take this module?

Welcome to QoS Concepts!

Imagine driving on a heavily congested road and you are in a rush to meet a friend for dinner. You hear the siren and see the lights of an ambulance behind you. You need to move off the road to let the ambulance through. The ambulance getting to the hospital takes priority over you getting to the restaurant on time.

Much like the ambulance taking priority in the traffic on the highway, some forms of network traffic need priority over others. Why? Get started with this module to find out!

9.0.2. What will I learn to do in this module?

Module Title: QoS Concepts

Module Objective: Explain how networking devices implement QoS.

Topic Title	Topic Objective
Network Transmission Quality	Explain how network transmission characteristics impact quality.
Traffic Characteristics	Describe minimum network requirements for voice, video, and data traffic.
Queuing Algorithms	Describe the queuing algorithms used by networking devices.
QoS Models	Describe the different QoS models.
QoS Implementation Techniques	Explain how QoS uses mechanisms to ensure transmission quality.

9.1. Network Transmission Quality

9.1.1 Video Tutorial – The Purpose of QoS

Click Play for a brief explanation of the purpose of QoS.

9.1.2. Prioritizing Traffic

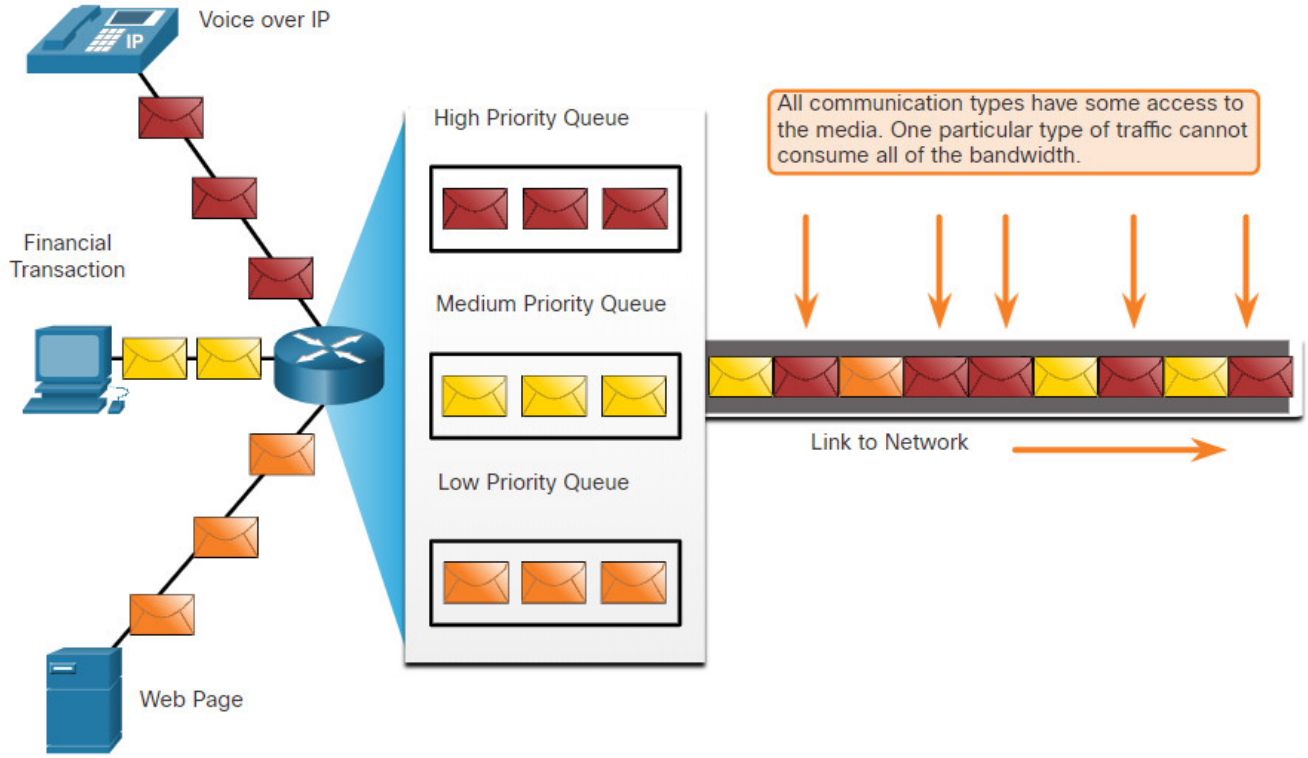
In the previous video, you learned about the purpose of Quality of Service (QoS). QoS is an ever-increasing requirement of networks today. New applications, such as voice and live video transmissions, create higher expectations for quality delivery among users.

Congestion occurs when multiple communication lines aggregate onto a single device such as a router, and then much of that data is placed on just a few outbound interfaces, or onto a slower interface. Congestion can also occur when large data packets prevent smaller packets from being transmitted in a timely manner.

When the volume of traffic is greater than what can be transported across the network, devices queue (hold) the packets in memory until resources become available to transmit them. Queuing packets causes delay because new packets cannot be transmitted until previous packets have been processed. If the number of packets to be queued continues to increase, the memory within the device fills up and packets are dropped. One QoS technique that can help with this problem is to classify data into multiple queues, as shown in the figure.

Note: A device implements QoS only when it is experiencing some type of congestion.

Using Queues to Priorities Communications

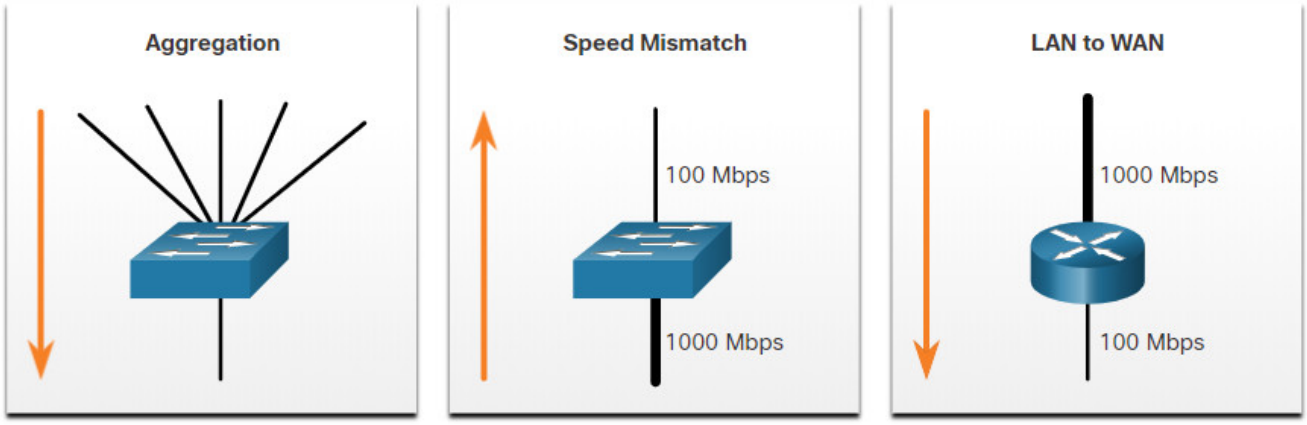


9.1.3. Bandwidth, Congestion, Delay, and Jitter

Network bandwidth is measured in the number of bits that can be transmitted in a single second, or bits per second (bps). For example, a network device may be described as having the capability to perform at 10 gigabits per second (Gbps).

Network congestion causes delay. An interface experiences congestion when it is presented with more traffic than it can handle. Network congestion points are ideal candidates for QoS mechanisms. The figure shows three examples of typical congestion points.

Examples of Congestion Points



Delay or latency refers to the time it takes for a packet to travel from the source to the destination. Two types of delays are fixed and variable. A fixed delay is a specific amount of time a specific process takes, such as how long it takes to place a bit on the transmission media. A variable delay takes an unspecified amount of time and is affected by factors such as how much traffic is being processed.

The sources of delay are summarized in the table.

Sources of Delay

Delay	Description
Code delay	The fixed amount of time it takes to compress data at the source before transmitting to the first internetworking device, usually a switch.
Packetization delay	The fixed time it takes to encapsulate a packet with all the necessary header information.
Queuing delay	The variable amount of time a frame or packet waits to be transmitted on the link.
Serialization delay	The fixed amount of time it takes to transmit a frame onto the wire.

Delay	Description
Propagation delay	The variable amount of time it takes for the frame to travel between the source and destination.
De-jitter delay	The fixed amount of time it takes to buffer a flow of packets and then send them out in evenly spaced intervals.

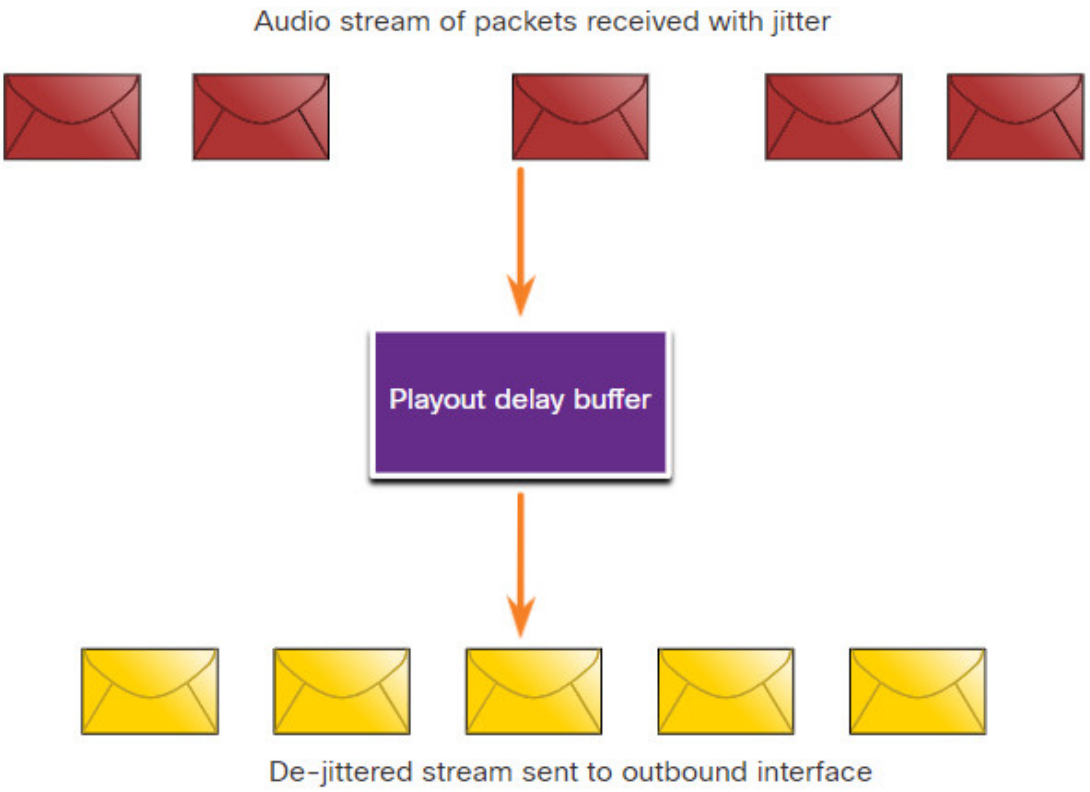
Jitter is the variation in the delay of received packets. At the sending side, packets are sent in a continuous stream with the packets spaced evenly apart. Due to network congestion, improper queuing, or configuration errors, the delay between each packet can vary instead of remaining constant. Both delay and jitter need to be controlled and minimized to support real-time and interactive traffic.

9.1.4. Packet Loss

Without any QoS mechanisms in place, packets are processed in the order in which they are received. When congestion occurs, network devices such as routers and switches can drop packets. This means that time-sensitive packets, such as real-time video and voice, will be dropped with the same frequency as data that is not time-sensitive, such as email and web browsing.

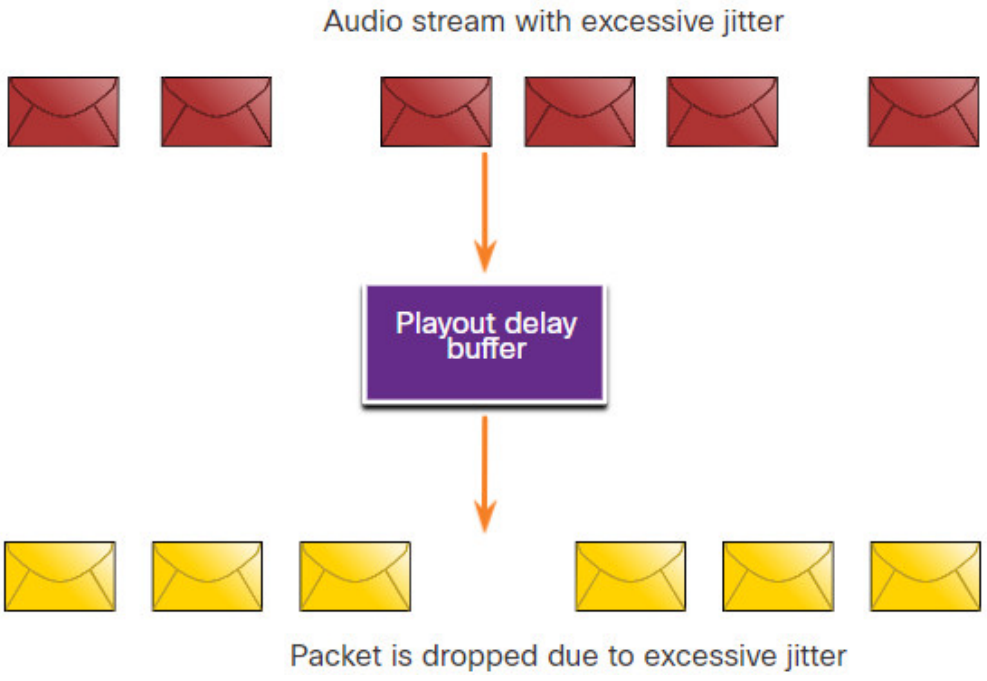
When a router receives a Real-Time Protocol (RTP) digital audio stream for Voice over IP (VoIP), it must compensate for the jitter that is encountered. The mechanism that handles this function is the playout delay buffer. The playout delay buffer must buffer these packets and then play them out in a steady stream, as shown in the figure. The digital packets are later converted back to an analog audio stream.

Playout Delay Buffer Compensates for Jitter



If the jitter is so large that it causes packets to be received out of the range of this buffer, the out-of-range packets are discarded and dropouts are heard in the audio, as shown in the figure.

Packet Dropped Due to Excessive Jitter



For losses as small as one packet, the digital signal processor (DSP) interpolates what it thinks the audio should be and no problem is audible to the user. However, when jitter exceeds what the DSP can do to make up for the missing packets, audio problems are heard.

Packet loss is a very common cause of voice quality problems on an IP network. In a properly designed network, packet loss should be near zero. The voice codecs used by the DSP can tolerate some degree of packet loss without a dramatic effect on voice quality. Network engineers use QoS mechanisms to classify voice packets for zero packet loss. Bandwidth is guaranteed for the voice calls by giving priority to voice traffic over traffic that is not sensitive to delays.

9.2. Traffic Characteristics

9.2.1 Video Tutorial – Traffic Characteristics

Click Play for an overview of how QoS can be used to treat packets differently based on the characteristics of the traffic.

9.2.2. Network Traffic Trends

In a previous topic, you learned about network transmission quality. In this topic you will learn about traffic characteristics (voice, video, and data). In the early 2000s, the predominant types of IP traffic were voice and data. Voice traffic has a predictable bandwidth need and known packet arrival times. Data traffic is not real-time and has unpredictable bandwidth need. Data traffic can temporarily burst, as when a large file is being downloaded. This bursting can consume the entire bandwidth of a link.

More recently, video traffic has become the increasingly important to business communications and operations. According to the Cisco Visual Networking Index (VNI), video traffic represented 70% of all traffic in 2017. By 2022, video will represent 82% of all traffic. In addition, mobile video traffic will reach 60.9 exabytes per month by 2022, up from 6.8 exabytes per month in 2017.The type of demands that voice, video, and data traffic place on the network are very different.

9.2.3. Voice

Voice traffic is predictable and smooth, as shown in the figure. However, voice is very sensitive to delays and dropped packets. It makes no sense to re-transmit voice if packets are lost; therefore, voice packets must receive a higher priority than other types of traffic. For example, Cisco products use the RTP port range 16384 to 32767 to prioritize voice traffic. Voice can tolerate a certain amount of latency, jitter, and loss without any noticeable effects.

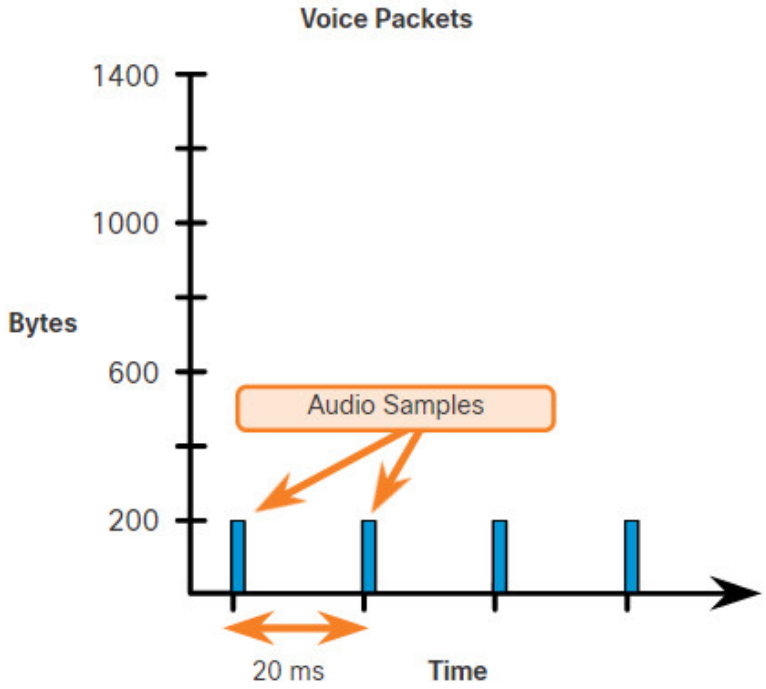
Latency should be no more than 150 milliseconds (ms). Jitter should be no more than 30 ms, and voice packet loss should be no more than 1%. Voice traffic requires at least 30 Kbps of bandwidth. The table gives a summary of voice traffic characteristics and requirements.

Voice Traffic Characteristics	One-Way Requirements
• Smooth • Benign • Drop sensitive • Delay sensitive • UDP priority	• Latency ≤ 150ms • Jitter ≤ 30ms • Loss ≤ 1% Bandwidth (30 – 128 Kbps)

9.2.4. Video

Without QoS and a significant amount of extra bandwidth capacity, video quality typically degrades. The picture appears blurry, jagged, or in slow motion. The audio portion of the feed may become unsynchronized with the video.

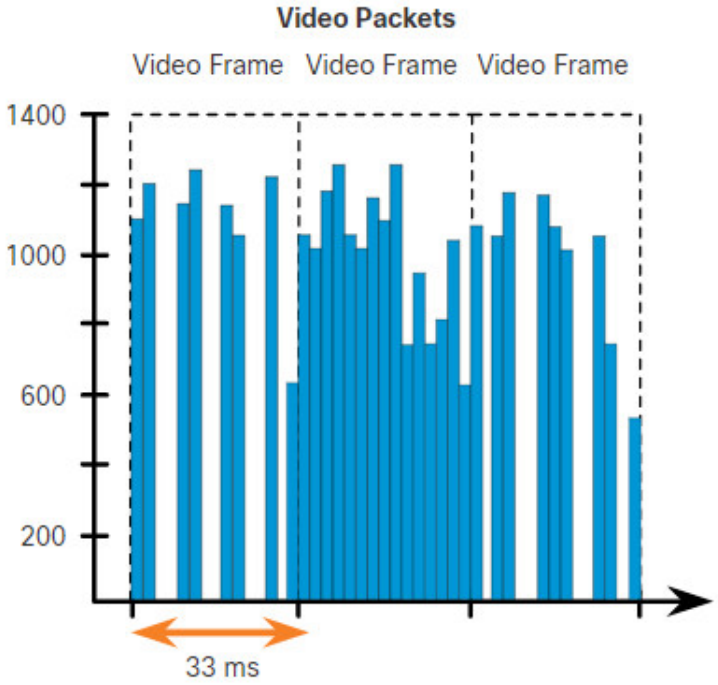
Video traffic tends to be unpredictable, inconsistent, and bursty compared to voice traffic. Compared to voice, video is less resilient to loss and has a higher volume of data per packet. Notice in the figure how voice packets arrive every 20 ms and are a predictable 200 bytes each.



In contrast, the number and size of video packets varies every 33 ms based on the content of the video, as shown in the figure. For example, if the video stream consists of content that is not changing much from frame to frame, then the video packets will be small, and fewer are required to maintain acceptable user experience. However, if the video steam consists of

content that is rapidly changing, such as an action sequence in a movie, then the video packets will be larger. More are required per the 33 ms time slot to maintain an acceptable user experience.

UDP ports such as 554, are used for the Real-Time Streaming Protocol (RSTP) and should be given priority over other, less delay-sensitive, network traffic. Similar to voice, video can tolerate a certain amount of latency, jitter, and loss without any noticeable effects. Latency should be no more than 400 milliseconds (ms). Jitter should be no more than 50 ms, and video packet loss should be no more than 1%. Video traffic requires at least 384 Kbps of bandwidth. The table gives a summary of video traffic characteristics and requirements.



Video Traffic Characteristics	One-Way Requirements
• Bursty • Greedy • Drop sensitive • Delay sensitive • UDP priority	• Latency ≤ 200-400 ms • Jitter ≤ 30-50 ms • Loss ≤ 0.1–1% • Bandwidth (384 Kbps – > 20 Mbps

9.2.5. Data

Most applications use either TCP or UDP. Unlike UDP, TCP performs error recovery. Data applications that have no tolerance for data loss, such as email and web pages, use TCP to ensure that, if packets are lost in transit, they will be resent. Data traffic can be smooth or bursty. Network control traffic is usually smooth and predictable. When there is a topology change, the network control traffic may burst for a few seconds. But the capacity of today's networks can easily handle the increase in network control traffic as the network converges.

However, some TCP applications can consume a large portion of network capacity. FTP will consume as much bandwidth as it can get when you download a large file, such as a movie or game. The table summarizes data traffic characteristics.

Data Traffic Characteristics

- Smooth/bursty
- Benign/greedy
- Drop insensitive
- Delay insensitive
- TCP retransmits

Although data traffic is relatively insensitive to drops and delays compared to voice and video, a network administrator still needs to consider the quality of the user experience, sometimes referred to as Quality of Experience or QoE. There are two main factors that a network administrator needs to ask about the flow of data traffic:

- Does the data come from an interactive application?
- Is the data mission critical?

The table compares these two factors.

Factors to Consider for Data Delay

Factor	Mission Critical	Not Mission Critical
Interactive	Prioritize for the lowest delay of all data traffic and strive for a 1 to 2 second response time.	Applications could benefit from lower delay.
Not interactive	Delay can vary greatly as long as the necessary minimum bandwidth is supplied.	Gets any leftover bandwidth after all voice, video, and other data application needs are met.

9.3. Queuing Algorithms

9.3.1 Video Tutorial – QoS Algorithms

Click Play for an overview of the different types of QoS queuing algorithms.

9.3.2. Queuing Overview

The previous topic covered traffic characteristics. This topic will explain the queuing algorithms used to implement QoS. The QoS policy implemented by the network administrator becomes active when congestion occurs on the link. Queuing is a congestion

management tool that can buffer, prioritize, and, if required, reorder packets before being transmitted to the destination.

A number of queuing algorithms are available. For the purposes of this course, we will focus on the following:

- First-In, First-Out (FIFO)
- Weighted Fair Queuing (WFQ)
- Class-Based Weighted Fair Queuing (CBWFQ)
- Low Latency Queuing (LLQ)

9.3.3. First In First Out

In its simplest form, First In First Out (FIFO) queuing, also known as first-come, first-served queuing, buffers and forwards packets in the order of their arrival.

FIFO has no concept of priority or classes of traffic and consequently, makes no decision about packet priority. There is only one queue, and all packets are treated equally. Packets are sent out an interface in the order in which they arrive, as shown in the figure. Although some traffic may be more important or time-sensitive based on the priority classification, notice that the traffic is sent out in the order it is received.

When FIFO is used, important or time-sensitive traffic can be dropped when there is congestion on the router or switch interface. When no other queuing strategies are configured, all interfaces, except serial interfaces at E1 (2.048 Mbps) and below, use FIFO by default. (Serial interfaces at E1 and below use WFQ by default.)

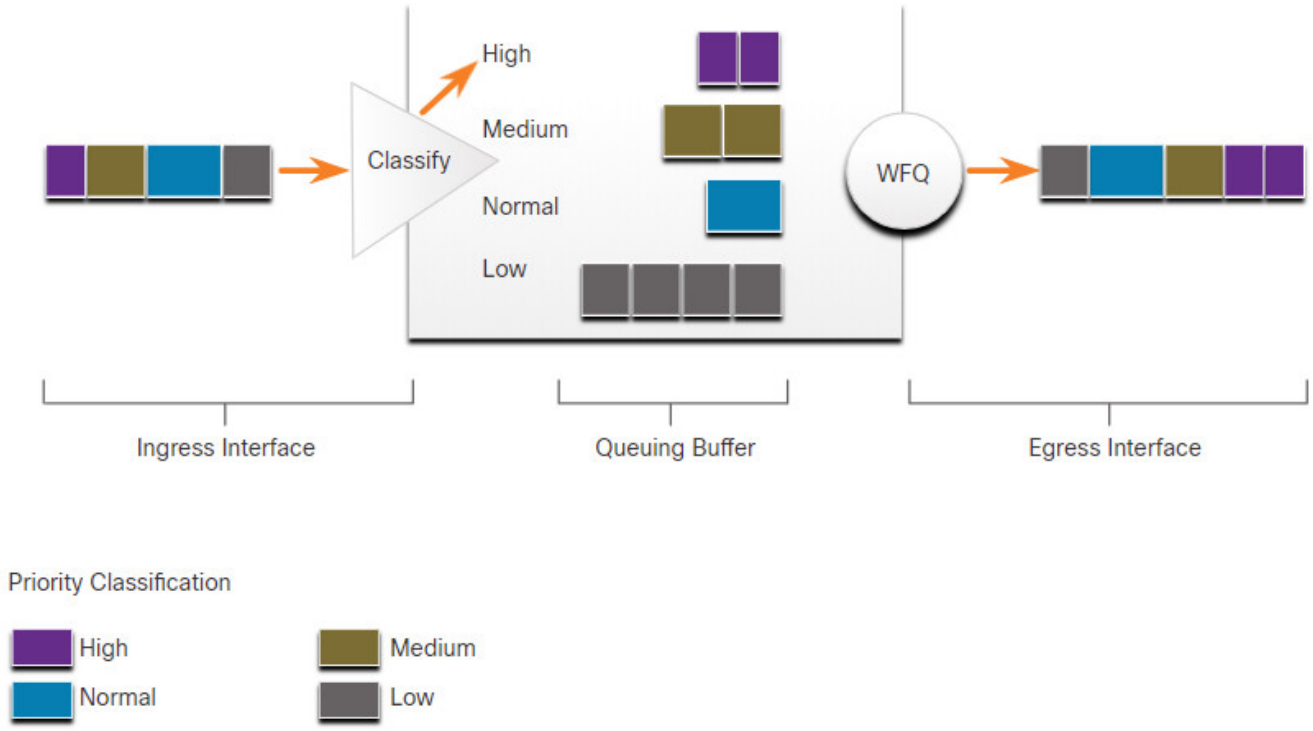
FIFO, which is the fastest method of queuing, is effective for large links that have little delay and minimal congestion. If your link has very little congestion, FIFO queuing may be the only queuing you need to use.

FIFO Queuing Example

9.3.4. Weighted Fair Queuing (WFQ)

Weighted Fair Queuing (WFQ) is an automated scheduling method that provides fair bandwidth allocation to all network traffic. WFQ does not allow classification options to be configured. WFQ applies priority, or weights, to identified traffic and classifies it into conversations or flows, as shown in the figure.

Weighted Fair Queuing Example



WFQ then determines how much bandwidth each flow is allowed relative to other flows. The flow-based algorithm used by WFQ simultaneously schedules interactive traffic to the front of a queue to reduce response time. It then fairly shares the remaining bandwidth among high-bandwidth flows. WFQ allows you to give low-volume, interactive traffic, such as Telnet sessions and voice, priority over high-volume traffic, such as FTP sessions. When multiple file transfers flows are occurring simultaneously, the transfers are given comparable bandwidth.

WFQ classifies traffic into different flows based on packet header addressing, including such characteristics as source and destination IP addresses, MAC addresses, port numbers, protocol, and Type of Service (ToS) value. The ToS value in the IP header can be used to classify traffic.

Low-bandwidth traffic flows, which comprise the majority of traffic, receive preferential service which allows their entire offered loads to be sent in a timely fashion. High-volume traffic flows share the remaining capacity proportionally among themselves.

Limitations

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 precise control over bandwidth allocation that CBWFQ offers.

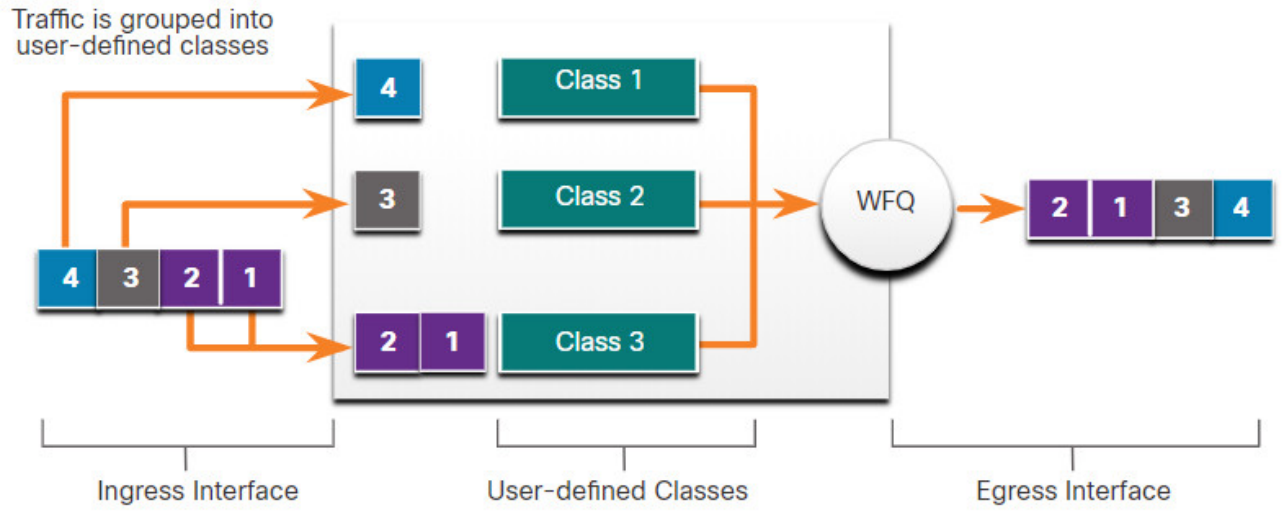
9.3.5. Class-Based Weighted Fair Queuing (CBWFQ)

Class-Based Weighted Fair Queuing (CBWFQ) extends the standard WFQ functionality to provide support for user-defined traffic classes. With 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. A FIFO queue is reserved for each class, and traffic belonging to a class is directed to the queue for that class, as shown in the figure.

When 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.

CBWFQ Example



After a queue has reached its configured queue limit, adding more packets to the class causes tail drop or packet drop to take effect, depending on how class policy is configured. Tail drop means a router simply discards any packet that arrives at the tail end of a queue that has

completely used up its packet-holding resources. This is the default queuing response to congestion. Tail drop treats all traffic equally and does not differentiate between classes of service.

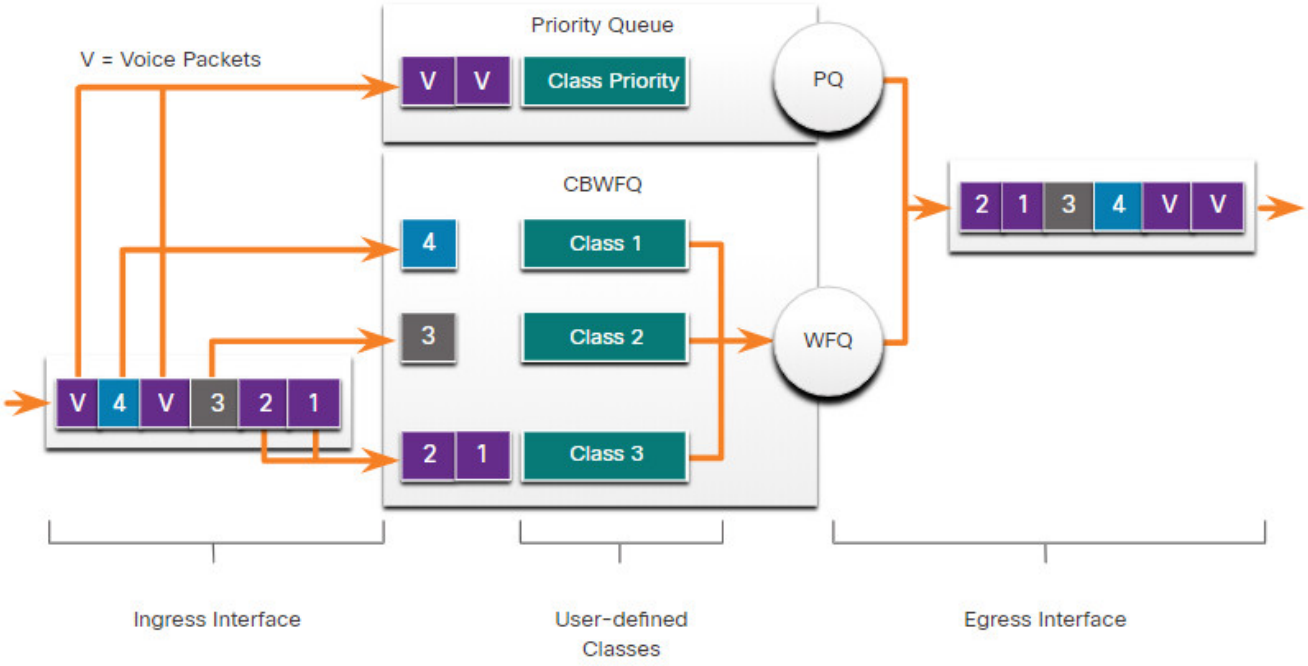
9.3.6. Low Latency Queuing (LLQ)

The Low Latency Queuing (LLQ) feature brings strict priority queuing (PQ) to CBWFQ. Strict PQ allows delay-sensitive packets such as voice to be sent before packets in other queues. LLQ provides strict priority queuing for CBWFQ, reducing jitter in voice conversations, as shown in the figure.

Without LLQ, CBWFQ provides WFQ based on defined classes with no strict priority queue available for real-time traffic. 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 allows delay-sensitive packets such as voice to be sent first (before packets in other queues), giving delay-sensitive packets preferential treatment over other traffic. Although it is possible to classify various types of real-time traffic to the strict priority queue, Cisco recommends that only voice traffic be directed to the priority queue.

LLQ Example



9.4. QoS Models

9.4.1 Video Tutorial – QoS Models

Click Play for a brief explanation of the purpose of QoS.

9.4.2. Selecting an Appropriate QoS Policy Model

How can QoS be implemented in a network? There are three models for implementing QoS:

- Best-effort model
- Integrated services (IntServ)
- Differentiated services (DiffServ)

The table summarizes these three models. QoS is implemented in a network using either IntServ or DiffServ. While IntServ provides the highest guarantee of QoS, it is very resource-intensive, and therefore, not easily scalable. In contrast, DiffServ is less resource-intensive and more scalable. The two are sometimes co-deployed in network QoS implementations.

Models for Implementing QoS

Model	Description
Best-effort model	• This is not really an implementation as QoS is not explicitly configured. • Use this when QoS is not required.
Integrated services (IntServ)	• IntServ provides very high QoS to IP packets with guaranteed delivery. • It defines a signaling process for applications to signal to the network that they require special QoS for a period and that bandwidth should be reserved. • IntServ can severely limit the scalability of a network.
Differentiated services (DiffServ)	• DiffServ provides high scalability and flexibility in implementing QoS. • Network devices recognize traffic classes and provide different levels of QoS to different traffic classes.

9.4.3. Best Effort

The basic design of the internet is best-effort packet delivery and provides no guarantees. This approach is still predominant on the internet today and remains appropriate for most purposes. The best-effort model treats all network packets in the same way, so an emergency

voice message is treated the same way that a digital photograph attached to an email is treated. Without QoS, the network cannot tell the difference between packets and, as a result, cannot treat packets preferentially.

The best-effort model is similar in concept to sending a letter using standard postal mail. Your letter is treated exactly the same as every other letter. With the best-effort model, the letter may never arrive, and, unless you have a separate notification arrangement with the letter recipient, you may never know that the letter did not arrive.

The table lists the benefits and drawbacks of the best effort model.

Benefits and Drawbacks of Best-Effort Model

Benefits	Drawbacks
The model is the most scalable.	There are no guarantees of delivery.
Scalability is only limited by available bandwidth, in which case all traffic is equally affected.	Packets will arrive whenever they can and in any order possible, if they arrive at all.
No special QoS mechanisms are required.	No packets have preferential treatment.
It is the easiest and quickest model to deploy.	Critical data is treated the same as casual email is treated.

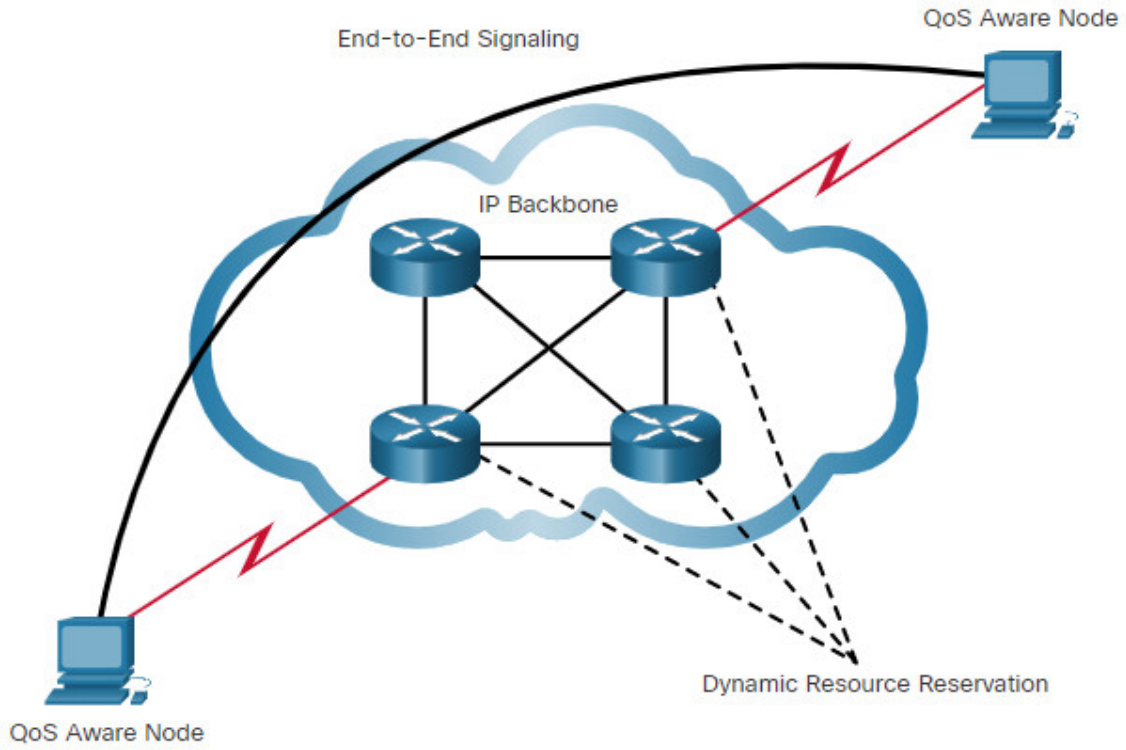
9.4.4. Integrated Services

The IntServ architecture model (RFC 1633, 2211, and 2212) was developed in 1994 to meet the needs of real-time applications, such as remote video, multimedia conferencing, data visualization applications, and virtual reality. IntServ is a multiple-service model that can accommodate many QoS requirements.

IntServ delivers the end-to-end QoS that real-time applications require. IntServ explicitly manages network resources to provide QoS to individual flows or streams, sometimes called microflows. It uses resource reservation and admission-control mechanisms as building blocks to establish and maintain QoS. This is similar to a concept known as “hard QoS.” Hard QoS guarantees traffic characteristics, such as bandwidth, delay, and packet-loss rates, from end to end. Hard QoS ensures both predictable and guaranteed service levels for mission-critical applications.

The figure shows a simple illustration of the IntServ model.

Simple IntServ Example



IntServ uses a connection-oriented approach inherited from telephony network design. Each individual communication must explicitly specify its traffic descriptor and requested resources to the network. The edge router performs admission control to ensure that available resources are sufficient in the network. The IntServ standard assumes that routers along a path set and maintain the state for each individual communication.

In the IntServ model, the application requests a specific kind of service from the network before sending data. The application informs the network of its traffic profile and requests a particular kind of service that can encompass its bandwidth and delay requirements. IntServ uses the Resource Reservation Protocol (RSVP) to signal the QoS needs of an application's traffic along devices in the end-to-end path through the network. If network devices along the path can reserve the necessary bandwidth, the originating application can begin transmitting. If the requested reservation fails along the path, the originating application does not send any data.

The edge router performs admission control based on information from the application and available network resources. The network commits to meeting the QoS requirements of the application as long as the traffic remains within the profile specifications. The network fulfills its commitment by maintaining the per-flow state and then performing packet classification, policing, and intelligent queuing based on that state.

The table lists the benefits and drawbacks of the IntServ model.

Benefits and Drawbacks of IntServ Model

Benefits	Drawbacks
• Explicit end-to-end resource admission control • Per-request policy admission control • Signaling of dynamic port numbers	• Resource intensive due to the stateful architecture requirement for continuous signaling. • Flow-based approach not scalable to large implementations such as the internet.

9.4.5. Differentiated Services

The differentiated services (DiffServ) QoS model specifies a simple and scalable mechanism for classifying and managing network traffic. For example, DiffServ can provide low-latency guaranteed service to critical network traffic such as voice or video, while providing simple best-effort traffic guarantees to non-critical services such as web traffic or file transfers.

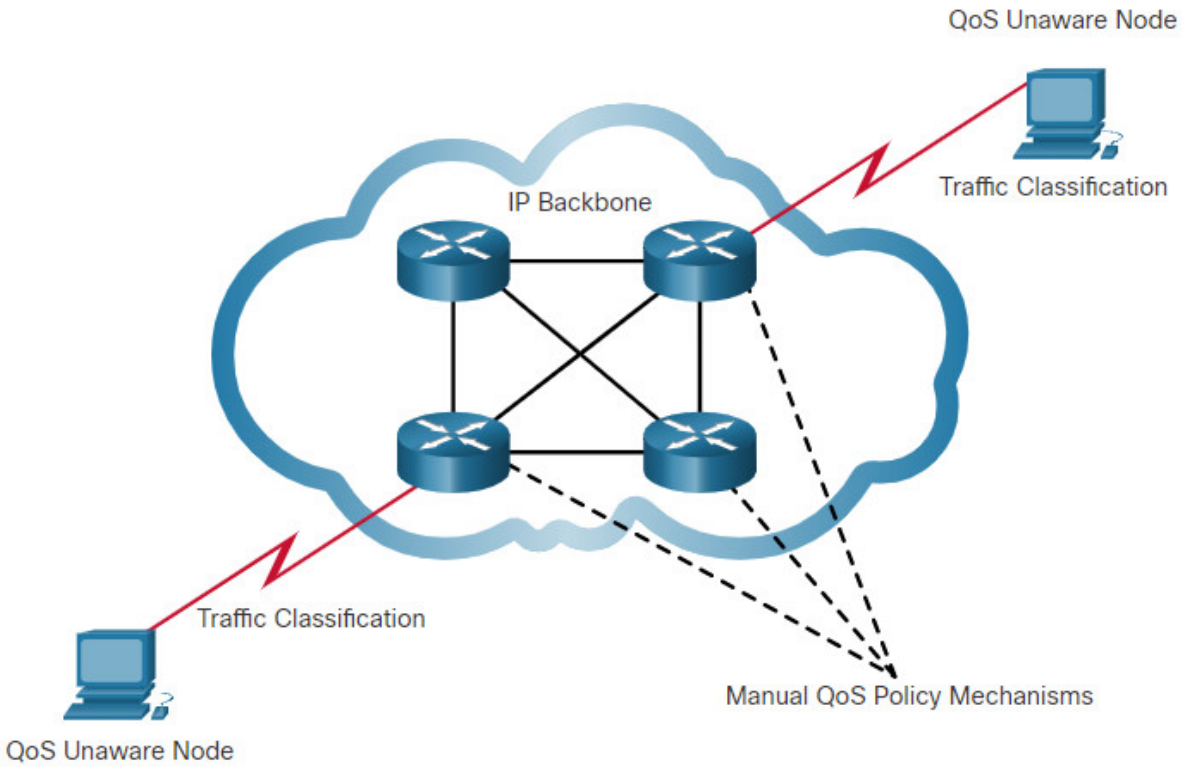
The DiffServ design overcomes the limitations of both the best-effort and IntServ models. The DiffServ model is described in RFCs 2474, 2597, 2598, 3246, 4594. DiffServ can provide an “almost guaranteed” QoS while still being cost-effective and scalable.

The DiffServ model is similar in concept to sending a package using a delivery service. You request (and pay for) a level of service when you send a package. Throughout the package network, the level of service you paid for is recognized and your package is given either preferential or normal service, depending on what you requested.

DiffServ is not an end-to-end QoS strategy because it cannot enforce end-to-end guarantees. However, DiffServ QoS is a more scalable approach to implementing QoS. Unlike IntServ and hard QoS, in which the end-hosts signal their QoS needs to the network, DiffServ does not use signaling. Instead, DiffServ uses a “soft QoS” approach. It works on the provisioned-QoS model, where network elements are set up to service multiple classes of traffic each with varying QoS requirements.

The figure shows a simple illustration of the DiffServ model.

Simple DiffServ Example



As a host forwards traffic to a router, the router classifies the flows into aggregates (classes) and provides the appropriate QoS policy for the classes. DiffServ enforces and applies QoS mechanisms on a hop-by-hop basis, uniformly applying global meaning to each traffic class to provide both flexibility and scalability. For example, DiffServ could be configured to group all TCP flows as a single class, and allocate bandwidth for that class, rather than for the individual flows as IntServ would do. In addition to classifying traffic, DiffServ minimizes signaling and state maintenance requirements on each network node.

Specifically, DiffServ divides network traffic into classes based on business requirements. Each of the classes can then be assigned a different level of service. As the packets traverse a network, each of the network devices identifies the packet class and services the packet according to that class. It is possible to choose many levels of service with DiffServ. For example, voice traffic from IP phones is usually given preferential treatment over all other application traffic, email is generally given best-effort service, and nonbusiness traffic can either be given very poor service or blocked entirely.

The figure lists the benefits and drawbacks of the DiffServ model.

Note: Modern networks primarily use the DiffServ model. However, due to the increasing volumes of delay- and jitter-sensitive traffic, IntServ and RSVP are sometimes co-deployed.

Benefits and Drawbacks of DiffServ Model

Benefits	Drawbacks

Benefits	Drawbacks
• Highly scalable • Provides many different levels of quality	• No absolute guarantee of service quality • Requires a set of complex mechanisms to work in concert throughout the network

9.5. QoS Implementation Techniques

9.5.1 Video Tutorial – QoS Implementation Techniques

Click Play for an overview of classification, marking, trust boundaries, congestion avoidance, shaping and policing.

9.5.2. Avoiding Packet Loss

Now that you have learned about traffic characteristics, queuing algorithms, and QoS models, it is time to learn about QoS implementation techniques.

Let's start with packet loss. Packet loss is usually the result of congestion on an interface. Most applications that use TCP experience slowdown because TCP automatically adjusts to network congestion. Dropped TCP segments cause TCP sessions to reduce their window sizes. Some applications do not use TCP and cannot handle drops (fragile flows).

The following approaches can prevent drops in sensitive applications:

- Increase link capacity to ease or prevent congestion.
- Guarantee enough bandwidth and increase buffer space to accommodate bursts of traffic from fragile flows. WFQ, CBWFQ, and LLQ can guarantee bandwidth and provide prioritized forwarding to drop-sensitive applications.
- Drop lower-priority packets before congestion occurs. Cisco IOS QoS provides queuing mechanisms, such as weighted random early detection (WRED), that start dropping lower-priority packets before congestion occurs.

9.5.3. QoS Tools

There are three categories of QoS tools, as described in the table:

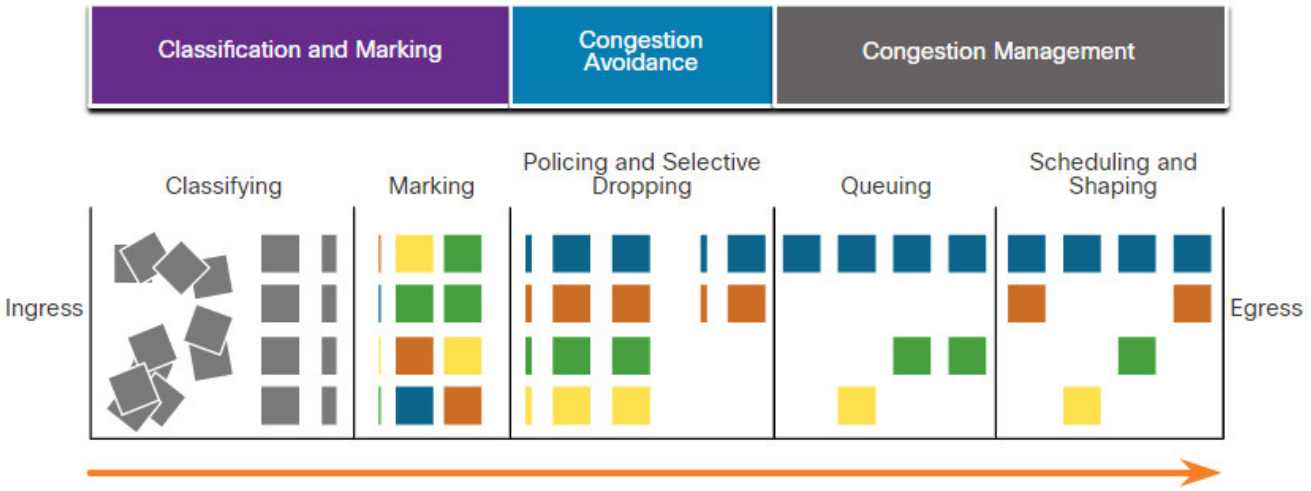
- Classification and marking tools
- Congestion avoidance tools
- Congestion management tools

Tools for Implementing QoS

QoS Tools	Description
Classification and marking tools	• Sessions, or flows, are analyzed to determine what traffic class they belong to. • When the traffic class is determined, the packets are marked.
Congestion avoidance tools	• Traffic classes are allotted portions of network resources, as defined by the QoS policy. • The QoS policy also identifies how some traffic may be selectively dropped, delayed, or re-marked to avoid congestion. • The primary congestion avoidance tool is WRED and is used to regulate TCP data traffic in a bandwidth-efficient manner before tail drops caused by queue overflows occur.
Congestion management tools	• When traffic exceeds available network resources, traffic is queued to await availability of resources. • Common Cisco IOS-based congestion management tools include CBWFQ and LLQ algorithms.

Refer to the figure to help understand the sequence of how these tools are used when QoS is applied to packet flows.

QoS Sequence



As shown in the figure, ingress packets (gray squares) are classified and their respective IP header is marked (colored squares). To avoid congestion, packets are then allocated resources based on defined policies. Packets are then queued and forwarded out the egress interface based on their defined QoS shaping and policing policy.

Note: Classification and marking can be done on ingress or egress, whereas other QoS actions such queuing and shaping are usually done on egress.

9.5.4. Classification and Marking

Before a packet can have a QoS policy applied to it, the packet has to be classified. Classification and marking allows us to identify or "mark" types of packets. Classification determines the class of traffic to which packets or frames belong. Only after traffic is marked can policies be applied to it.

How a packet is classified depends on the QoS implementation. Methods of classifying traffic flows at Layer 2 and 3 include using interfaces, ACLs, and class maps. Traffic can also be classified at Layers 4 to 7 using Network Based Application Recognition (NBAR).

Note: NBAR is a classification and protocol discovery feature of Cisco IOS software that works with QoS features. NBAR is out of scope for this course.

Marking means that we are adding a value to the packet header. Devices receiving the packet look at this field to see if it matches a defined policy. Marking should be done as close to the source device as possible. This establishes the trust boundary.

How traffic is marked usually depends on the technology. The table in the figure describes some the marking fields used in various technologies. The decision of whether to mark traffic at Layers 2 or 3 (or both) is not trivial and should be made after consideration of the following points:

- Layer 2 marking of frames can be performed for non-IP traffic.
- Layer 2 marking of frames is the only QoS option available for switches that are not "IP aware".
- Layer 3 marking will carry the QoS information end-to-end.

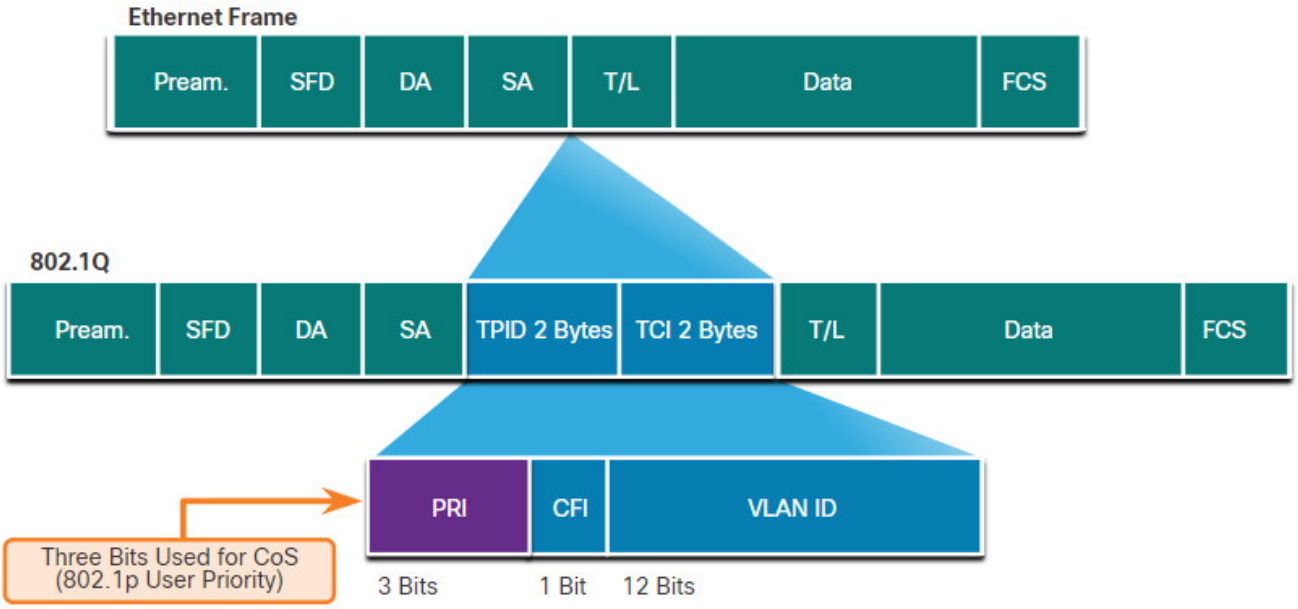
Traffic Marking for QoS

QoS Tools	Layer	Marking Field	Width in Bits
Ethernet (802.1Q, 802.1p)	2	Class of Service (CoS)	3
802.11 (Wi-Fi)	2	Wi-Fi Traffic Identifier (TID)	3
MPLS	2	Experimental (EXP)	3
IPv4 and IPv6	3	IP Precedence (IPP)	3
IPv4 and IPv6	3	Differentiated Services Code Point (DSCP)	6

9.5.5. Marking at Layer 2

802.1Q is the IEEE standard that supports VLAN tagging at Layer 2 on Ethernet networks. When 802.1Q is implemented, two fields are added to the Ethernet Frame. As shown in the figure, these two fields are inserted into the Ethernet frame following the source MAC address field.

Ethernet Class of Service (CoS) Values



The 802.1Q standard also includes the QoS prioritization scheme known as IEEE 802.1p. The 802.1p standard uses the first three bits in the Tag Control Information (TCI) field. Known as the Priority (PRI) field, this 3-bit field identifies the Class of Service (CoS) markings. Three bits means that a Layer 2 Ethernet frame can be marked with one of eight levels of priority (values 0-7) as displayed in the figure.

Ethernet Class of Service (CoS) Values

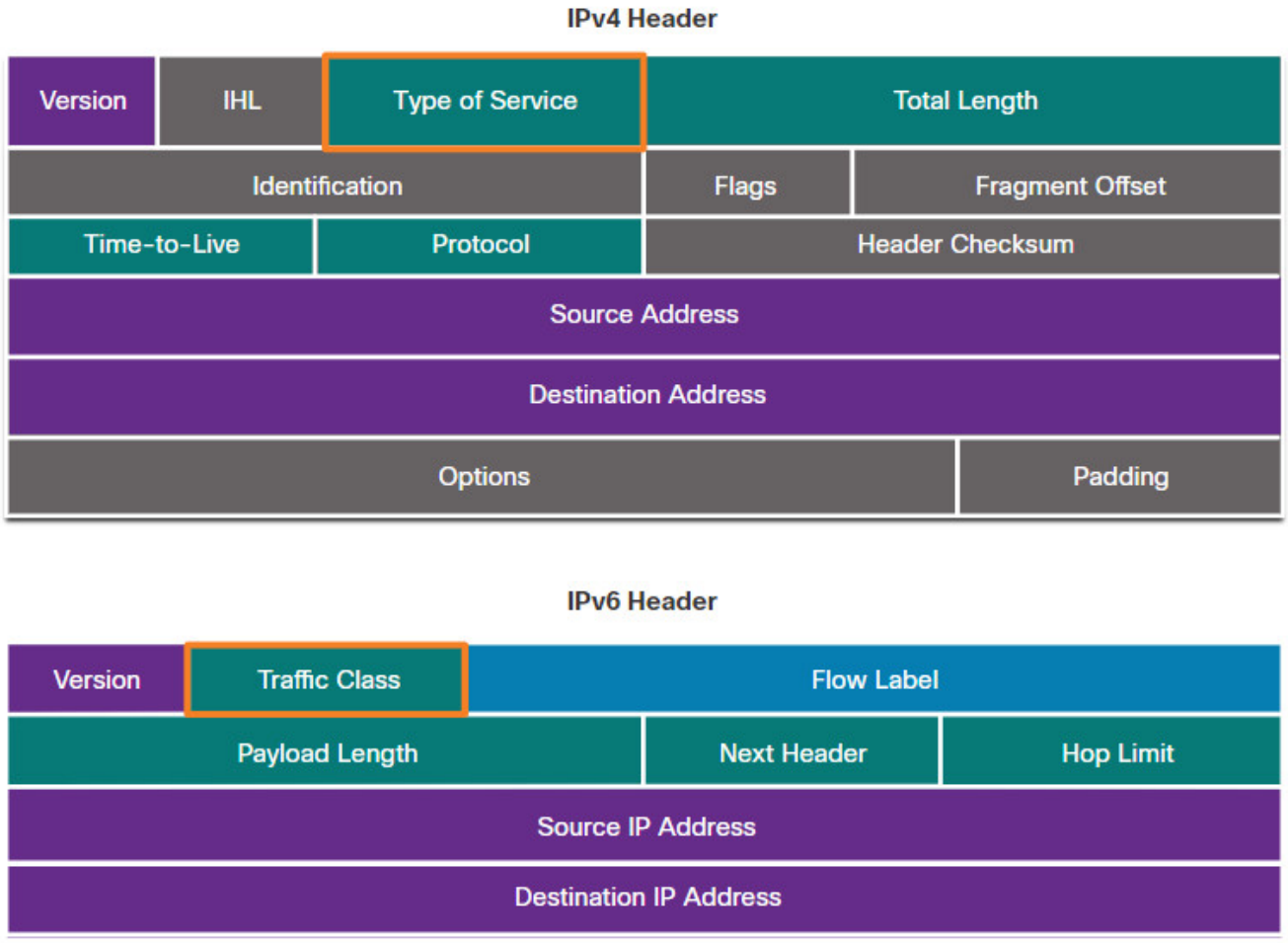
CoS Value	CoS Binary Value	Description
0	000	Best-Effort Data
1	001	Medium-Priority Data
2	010	High-Priority Data
3	011	Call Signaling
4	100	Videoconferencing
5	101	Voice bearer (voice traffic)
6	110	Reserved

CoS Value	CoS Binary Value	Description
7	111	Reserved

9.5.6. Marking at Layer 3

IPv4 and IPv6 specify an 8-bit field in their packet headers to mark packets. As shown in the figure, both IPv4 and IPv6 support an 8-bit field for marking: the Type of Service (ToS) field for IPv4 and the Traffic Class field for IPv6.

IPv4 and IPv6 Packet Headers

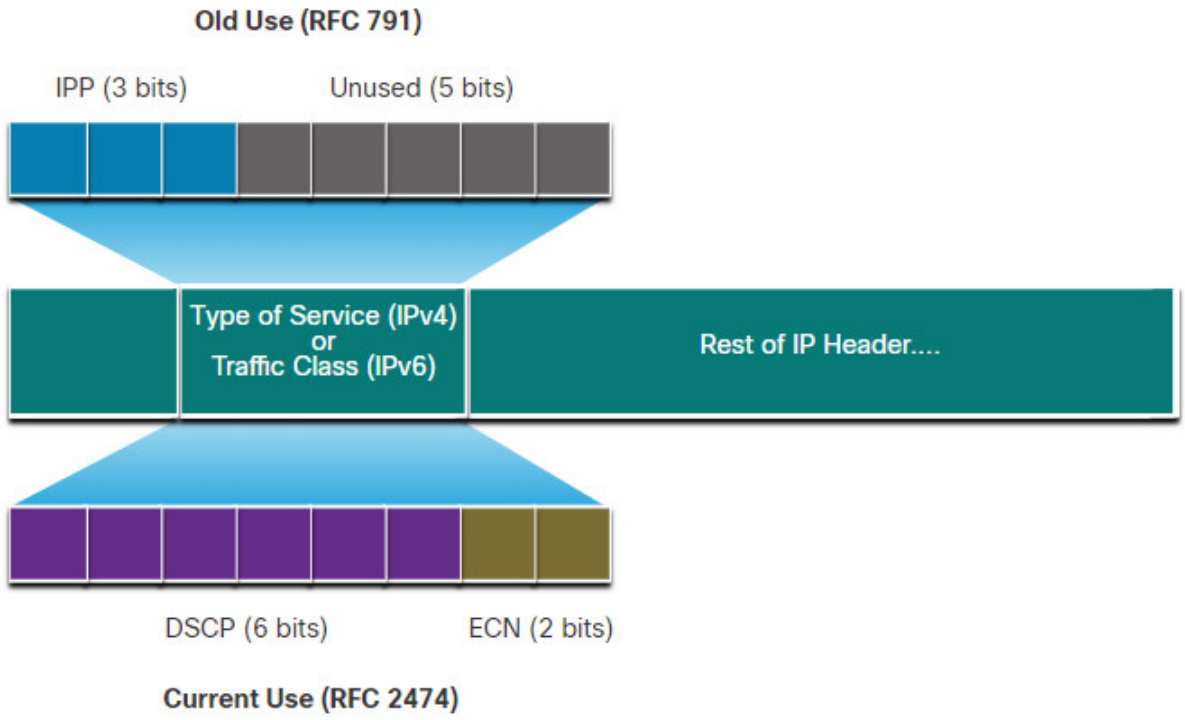


9.5.7. Type of Service and Traffic Class Field

The Type of Service (IPv4) and Traffic Class (IPv6) carry the packet marking as assigned by the QoS classification tools. The field is then referred to by receiving devices which forward the packets based on the appropriate assigned QoS policy.

The figure displays the contents of the 8-bit field. In RFC 791, the original IP standard specified the IP Precedence (IPP) field to be used for QoS markings. However, in practice, these three bits did not provide enough granularity to implement QoS.

RFC 2474 supersedes RFC 791 and redefines the ToS field by renaming and extending the IPP field. The new field, as shown in the figure, has 6-bits allocated for QoS. Called the Differentiated Services Code Point (DSCP) field, these six bits offer a maximum of 64 possible classes of service. The remaining two IP Extended Congestion Notification (ECN) bits can be used by ECN-aware routers to mark packets instead of dropping them. The ECN marking informs downstream routers that there is congestion in the packet flow.

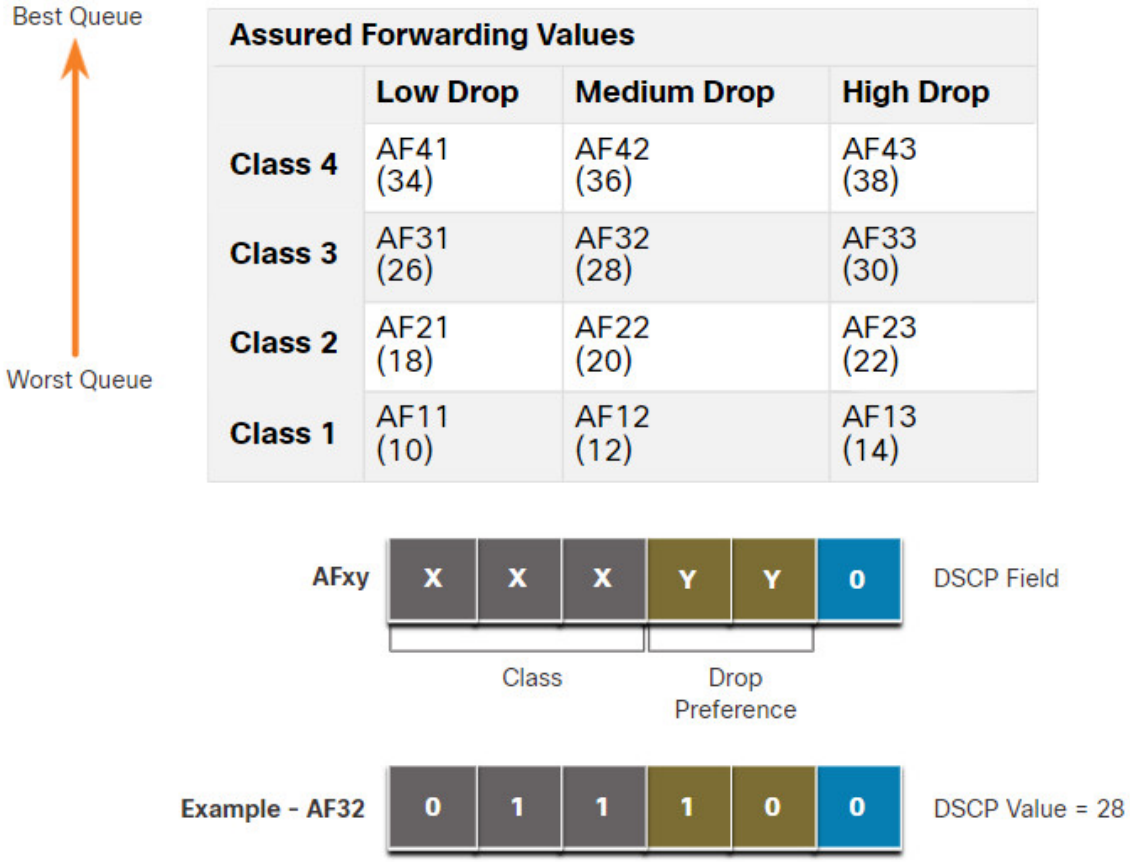


9.5.8. DSCP Values

The 64 DSCP values are organized into three categories:

- **Best-Effort (BE)** – This is the default for all IP packets. The DSCP value is 0. The per-hop behavior is normal routing. When a router experiences congestion, these packets will be dropped. No QoS plan is implemented.
- **Expedited Forwarding (EF)** – RFC 3246 defines EF as the DSCP decimal value 46 (binary 101110). The first 3 bits (101) map directly to the Layer 2 CoS value 5 used for voice traffic. At Layer 3, Cisco recommends that EF only be used to mark voice packets.
- **Assured Forwarding (AF)** – RFC 2597 defines AF to use the 5 most significant DSCP bits to indicate queues and drop preference. The definition of AF is illustrated in the figure.

Assured Forwarding Values



The AFxy formula is specified as follows:

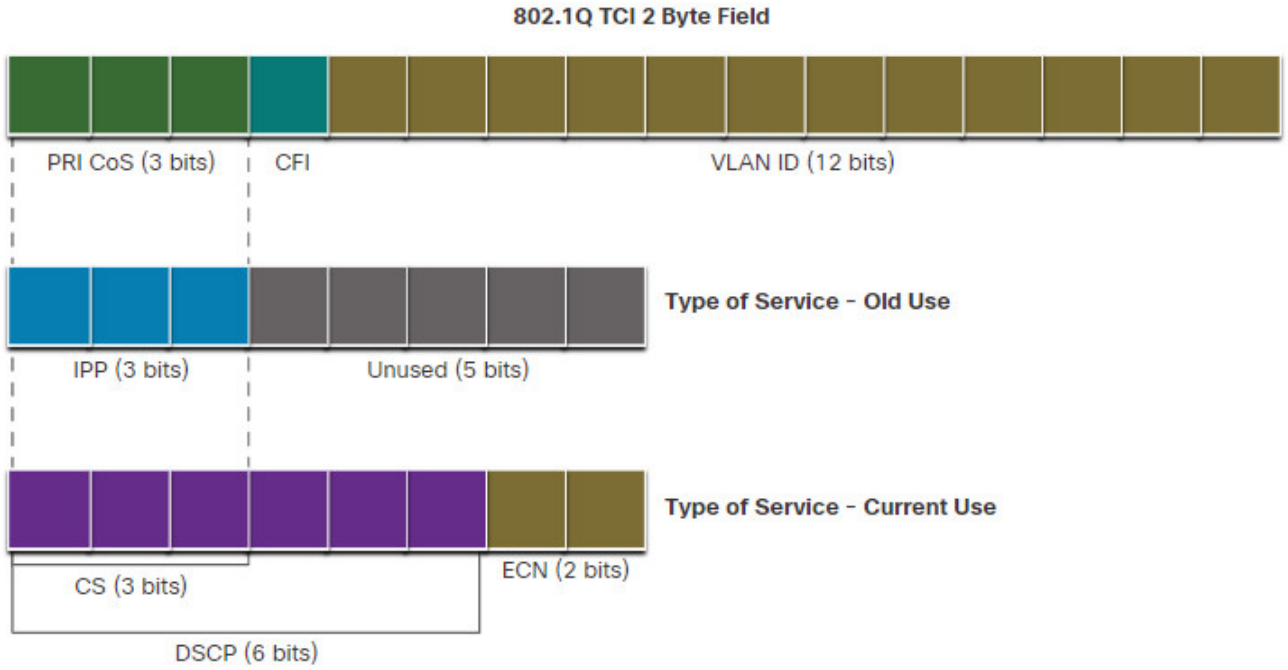
- The first 3 most significant bits are used to designate the class. Class 4 is the best queue and Class 1 is the worst queue.
- The 4th and 5th most significant bits are used to designate the drop preference.
- The 6th most significant bit is set to zero.

For example, AF32 belongs to class 3 (binary 011) and has a medium drop preference (binary 10). The full DSCP value is 28 because you include the 6th 0 bit (binary 011100).

9.5.9. Class Selector Bits

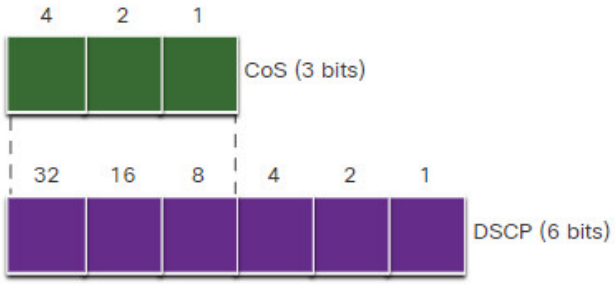
Because the first 3 most significant bits of the DSCP field indicate the class, these bits are also called the Class Selector (CS) bits. These 3 bits map directly to the 3 bits of the CoS field and the IPP field to maintain compatibility with 802.1p and RFC 791, as shown in the figure.

Layer 2 CoS and Layer 3 ToS



The table in the figure shows how the CoS values map to the Class Selectors and the corresponding DSCP 6-bit value. This same table can be used to map IPP values to the Class Selectors.

Mapping CoS to Class Selectors in DSCP



CoS values, Class Selectors, and corresponding DSCP 6-bit value

CoS Value	CoS Binary Value	Class Selector (CS)	CS Binary	DSCP Decimal Value
0	000	CS0*/DF	000 000	0
1	001	CS1	001 000	8
2	010	CS2	010 000	16
3	011	CS3	011 000	24
4	100	CS4	100 000	32
5	101	CS5	101 000	40
6	110	CS6	110 000	48
7	111	CS7	111 000	56

CoS values, Class Selectors, and corresponding DSCP 6-bit value

CoS Value	CoS Binary Value	Class Selector (CS)	CS Binary	DSCP Decimal Value
0	000	CS0*/DF	000 000	0
1	001	CS1	001 000	8
2	010	CS2	010 000	16
3	011	CS3	011 000	24
4	100	CS4	100 000	32
5	101	CS5	101 000	40
6	110	CS6	110 000	48
7	111	CS7	111 000	56

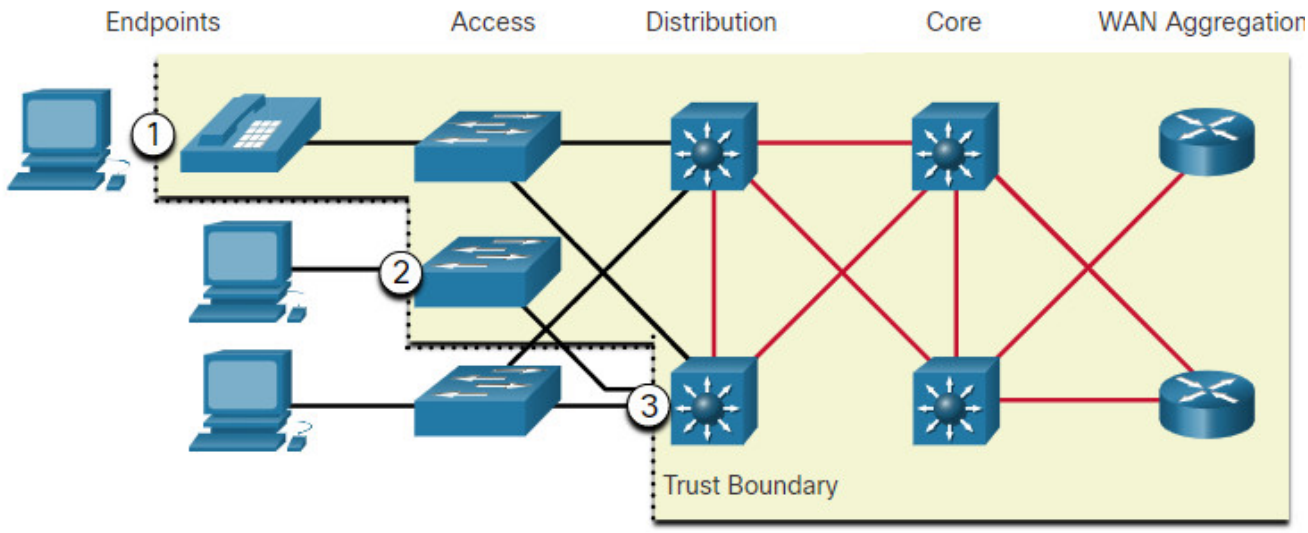
9.5.10. Trust Boundaries

Where should markings occur? Traffic should be classified and marked as close to its source as technically and administratively feasible. This defines the trust boundary, as shown in the figure.

1. Trusted endpoints have the capabilities and intelligence to mark application traffic to the appropriate Layer 2 CoS and/or Layer 3 DSCP values. Examples of trusted endpoints include IP phones, wireless access points, videoconferencing gateways and systems, IP conferencing stations, and more.
2. Secure endpoints can have traffic marked at the Layer 2 switch.
3. Traffic can also be marked at Layer 3 switches / routers.

Re-marking traffic, for example, re-marking CoS values to IP Precedent or DSCP values, is typically necessary.

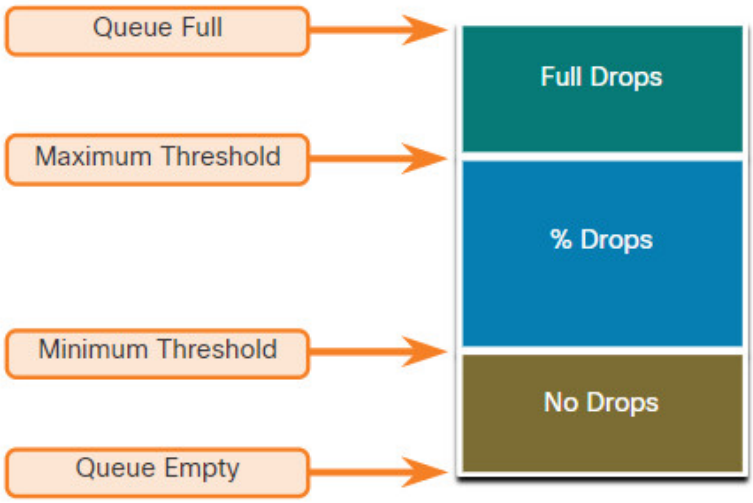
Various Trust Boundaries



9.5.11. Congestion Avoidance

Congestion management includes queuing and scheduling methods where excess traffic is buffered or queued (and sometimes dropped) while it waits to be sent out an egress interface. Congestion avoidance tools are simpler. They monitor network traffic loads in an effort to anticipate and avoid congestion at common network and internetwork bottlenecks before congestion becomes a problem. These tools can monitor the average depth of the queue, as represented in the figure. When the queue is below the minimum threshold, there are no drops. As the queue fills up to the maximum threshold, a small percentage of packets are dropped. When the maximum threshold is passed, all packets are dropped.

Congestion Avoidance Mechanisms



Some congestion avoidance techniques provide preferential treatment for which packets will get dropped. For example, Cisco IOS QoS includes weighted random early detection (WRED) as a possible congestion avoidance solution. The WRED algorithm allows for congestion avoidance on network interfaces by providing buffer management and allowing TCP traffic to decrease, or throttle back, before buffers are exhausted. Using WRED helps avoid tail drops and maximizes network use and TCP-based application performance. There is no congestion avoidance for User Datagram Protocol (UDP)-based traffic, such as voice traffic. In case of UDP-based traffic, methods such as queuing and compression techniques help to reduce and even prevent UDP packet loss.

9.5.12. Shaping and Policing

Traffic shaping and traffic policing are two mechanisms provided by Cisco IOS QoS software to prevent congestion.

Traffic shaping retains excess packets in a queue and then schedules the excess for later transmission over increments of time. The result of traffic shaping is a smoothed packet output rate, as shown in the figure.

Shaping Traffic Example

Shaping implies the existence of a queue and of sufficient memory to buffer delayed packets, while policing does not.

Ensure that you have sufficient memory when enabling shaping. In addition, shaping requires a scheduling function for later transmission of any delayed packets. This scheduling function allows you to organize the shaping queue into different queues. Examples of scheduling functions are CBWFQ and LLQ.

Shaping is an outbound concept; packets going out an interface get queued and can be shaped. In contrast, policing is applied to inbound traffic on an interface. When the traffic rate reaches the configured maximum rate, excess traffic is dropped (or remarked).

Policing is commonly implemented by service providers to enforce a contracted customer information rate (CIR). However, the service provider may also allow bursting over the CIR if the service provider's network is not currently experiencing congestion.

Policing Traffic Example

9.5.13. QoS Policy Guidelines

Your QoS policy must consider the full path from source to destination. If one device in the path is using a different policy than desired, then the entire QoS policy is impacted. For example, the stutter in video playback could be the result of one switch in the path that does not have the CoS value set appropriately.

A few guidelines that help ensure the best experience for end users includes the following:

- Enable queuing at every device in the path between source and destination.
- Classify and mark traffic as close the source as possible.
- Shape and police traffic flows as close to their sources as possible.

9.6. Module Practice and Quiz

9.6.1. What did I learn in this module?

Network Transmission Quality

Voice and live video transmissions create higher expectations for quality delivery among users, and create a need for Quality of Service (QoS). Congestion occurs when multiple communication lines aggregate onto a single device such as a router, and then much of that data is placed on just a few outbound interfaces, or onto a slower interface. Congestion can also occur when large data packets prevent smaller packets from being transmitted in a timely manner. Without any QoS mechanisms in place, packets are processed in the order in which they are received. When congestion occurs, network devices such as routers and switches can drop packets. This means that time-sensitive packets, such as real-time video and voice, will be dropped with the same frequency as data that is not time-sensitive, such as email and web browsing. When the volume of traffic is greater than what can be transported across the network, devices queue (hold) the packets in memory until resources become available to transmit them. Queuing packets causes delay because new packets cannot be transmitted until previous packets have been processed. One QoS technique that can help with this problem is to classify data into multiple queues. Network congestion points are ideal candidates for QoS mechanisms to mitigate delay and latency. Two types of delays are fixed and variable. Sources of delay are code delay, packetization delay, queuing delay, serialization delay, propagation delay, and de-jitter delay. Jitter is the variation in the delay of received packets. Due to network congestion, improper queuing, or configuration errors, the delay between each packet can vary instead of remaining constant. Both delay and jitter need to be controlled and minimized to support real-time and interactive traffic.

Traffic Characteristics

Voice and video traffic are two of the main reasons for QoS. Voice traffic is smooth and benign, but it is sensitive to drops and delays. Voice can tolerate a certain amount of latency, jitter, and loss without any noticeable effects. Latency should be no more than 150

milliseconds (ms). Jitter should be no more than 30 ms, and voice packet loss should be no more than 1%. Voice traffic requires at least 30 Kbps of bandwidth. Video traffic is more demanding than voice traffic because of the size of the packets it sends across the network. Video traffic is bursty, greedy, drop sensitive, and delay sensitive. Without QoS and a significant amount of extra bandwidth, video quality typically degrades. UDP ports such as 554, are used for the Real-Time Streaming Protocol (RSTP) and should be given priority over other, less delay-sensitive, network traffic. Similar to voice, video can tolerate a certain amount of latency, jitter, and loss without any noticeable effects. Latency should be no more than 400 milliseconds (ms). Jitter should be no more than 50 ms, and video packet loss should be no more than 1%. Video traffic requires at least 384 Kbps of bandwidth. Data traffic is not as demanding as voice and video traffic. Data packets often use TCP applications which can retransmit data and, therefore, are not sensitive to drops and delays. Although data traffic is relatively insensitive to drops and delays compared to voice and video, a network administrator still needs to consider the quality of the user experience, sometimes referred to as Quality of Experience (QoE). The two main factors that a network administrator needs to ask about the flow of data traffic are if the data comes from an interactive application and if the data is mission critical.

Queuing Algorithms

The QoS policy implemented by the network administrator becomes active when congestion occurs on the link. Queuing is a congestion management tool that can buffer, prioritize, and, if required, reorder packets before being transmitted to the destination. This course focuses on the following queuing algorithms: First-In, First-Out (FIFO), Weighted Fair Queuing (WFQ), Class-Based Weighted Fair Queuing (CBWFQ), and Low Latency Queuing (LLQ). FIFO queuing buffers and forwards packets in the order of their arrival. FIFO has no concept of priority or classes of traffic and consequently, makes no decision about packet priority. When FIFO is used, important or time-sensitive traffic can be dropped when there is congestion on the router or switch interface. WFQ is an automated scheduling method that provides fair bandwidth allocation to all network traffic. WFQ applies priority, or weights, to identified traffic and classifies it into conversations or flows. WFQ classifies traffic into different flows based on packet header addressing, including such characteristics as source and destination IP addresses, MAC addresses, port numbers, protocol, and Type of Service (ToS) value. The ToS value in the IP header can be used to classify traffic. CBWFQ extends the standard WFQ functionality to provide support for user-defined traffic classes. With CBWFQ, you define traffic classes based on match criteria including protocols, access control lists (ACLs), and input interfaces. LLQ feature brings strict priority queuing (PQ) to CBWFQ. Strict PQ allows delay-sensitive packets, such as voice, to be sent before packets in other queues, reducing jitter in voice conversations.

QoS Models

There are three models for implementing QoS: Best-effort model, Integrated services (IntServ), and Differentiated services (DiffServ). The Best-effort model is the most scalable but does not guarantee delivery and does not give any packet preferential treatment. The IntServ architecture model was developed to meet the needs of real-time applications, such as remote video, multimedia conferencing, data visualization applications, and virtual reality. IntServ is a multiple-service model that can accommodate many QoS requirements. IntServ explicitly manages network resources to provide QoS to individual flows or streams, sometimes called microflows. It uses resource reservation and admission-control mechanisms as building blocks to establish and maintain QoS. The DiffServ QoS model specifies a simple and scalable mechanism for classifying and managing network traffic. The DiffServ design overcomes the limitations of both the best-effort and IntServ models. The DiffServ model can provide an “almost guaranteed” QoS, while still being cost-effective and scalable. DiffServ divides network traffic into classes based on business requirements. Each of the classes can then be assigned a different level of service. As the packets traverse a network, each of the network devices identifies the packet class and services the packet according to that class. It is possible to choose many levels of service with DiffServ.

QoS Implementation Techniques

There are three categories of QoS tools: classification and marking tools, congestion avoidance tools, and congestion management tools. Before a packet can have a QoS policy applied to it, the packet has to be classified. Classification and marking allows us to identify or “mark” types of packets. Classification determines the class of traffic to which packets or frames belong. Methods of classifying traffic flows at Layer 2 and 3 include using interfaces, ACLs, and class maps. Traffic can also be classified at Layers 4 to 7 using Network Based Application Recognition (NBAR). The Type of Service (IPv4) and Traffic Class (IPv6) carry the packet marking as assigned by the QoS classification tools. The field is then referred to by receiving devices which forward the packets based on the appropriate assigned QoS policy. These fields have 6-bits allocated for QoS. Called the Differentiated Services Code Point (DSCP) field, these six bits offer a maximum of 64 possible classes of service. The field is then referred to by receiving devices which forward the packets based on the appropriate assigned QoS policy. The 64 DSCP values are organized into three categories: Best-Effort (BE), Expedited Forwarding (EF), Assured Forwarding (AF). Because the first 3 most significant bits of the DSCP field indicate the class, these bits are also called the Class Selector (CS) bits. Traffic should be classified and marked as close to its source as technically and administratively feasible. This defines the trust boundary. Congestion management includes queuing and scheduling methods where excess traffic is buffered or queued (and sometimes dropped) while it waits to be sent out an egress interface. Congestion avoidance tools help to monitor network traffic loads in an effort to anticipate and avoid congestion at common network and internetwork bottlenecks before congestion becomes a problem. Cisco IOS QoS includes weighted random early detection (WRED) as a possible congestion avoidance solution. The WRED algorithm allows for congestion avoidance on network interfaces by

providing buffer management and allowing TCP traffic to decrease, or throttle back, before buffers are exhausted. Traffic shaping and traffic policing are two mechanisms provided by Cisco IOS QoS software to prevent congestion.

9.6.2 Module Quiz – QoS Concepts

Download Slide Powerpoint (PPT)

CCNA 3 v7.0 Curriculum: Module 9 - QoS Concepts.pptx

1 file(s) 1.62 MB

Download

Tags:ccna 3 v7 modules