DQoS

Deploying Cisco QoS for Enterprise Networks

Version 1.0

Student Guide

Cisco Systems has more than 200 offices in the following countries and regions. Addresses, phone numbers, and fax numbers are listed on the Cisco Web site at www.cisco.com/go/offices.

Argentina • Australia • Austria • Belgium • Brazil • Bulgaria • Canada • Chile • China PRC • Colombia • Costa Rica • Croatia • Czech Republic • Denmark • Dubai, UAE • Finland • France • Germany • Greece • Hong Kong SAR • Hungary India • Indonesia • Ireland • Israel • Italy • Japan • Korea • Luxembourg • Malaysia • Mexico • The Netherlands • New Zealand • Norway • Peru • Philippines • Poland • Portugal • Puerto Rico • Romania • Russia • Saudi Arabia • Scotland • Singapore • Slovakia • Slovenia • South Africa • Spain • Sweden • Switzerland • Taiwan • Thailand • Turkey Ukraine • United Kingdom • United States • Venezuela • Vietnam • Zimbabwe

1

Course Introduction

Overview

"Deploying Cisco QoS for Enterprise Networks" (DQoS) v1.0 is an instructor-led course presented by Cisco training partners to their end-user customers. This five-day course focuses on using IOS QoS tools, up to and including Cisco IOS Software Release 12.1(5)T. The class provides hands-on experience of using the QoS tools in converged networks.

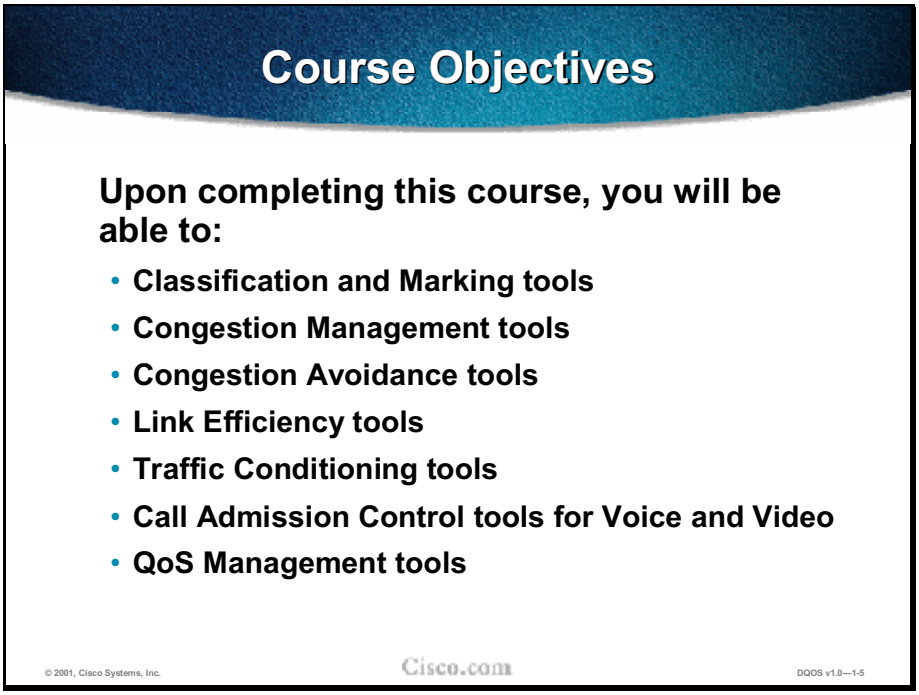
Upon completion of this training course, you will be able to design, implement, and troubleshoot Cisco QoS in Enterprise networks.

This chapter outlines the course prerequisites and course highlights, as well as some administrative issues. It includes the following topics:

- Course Objectives
- Prerequisites
- Participant Role
- General Administration
- Sources of Information
- Course Syllabus

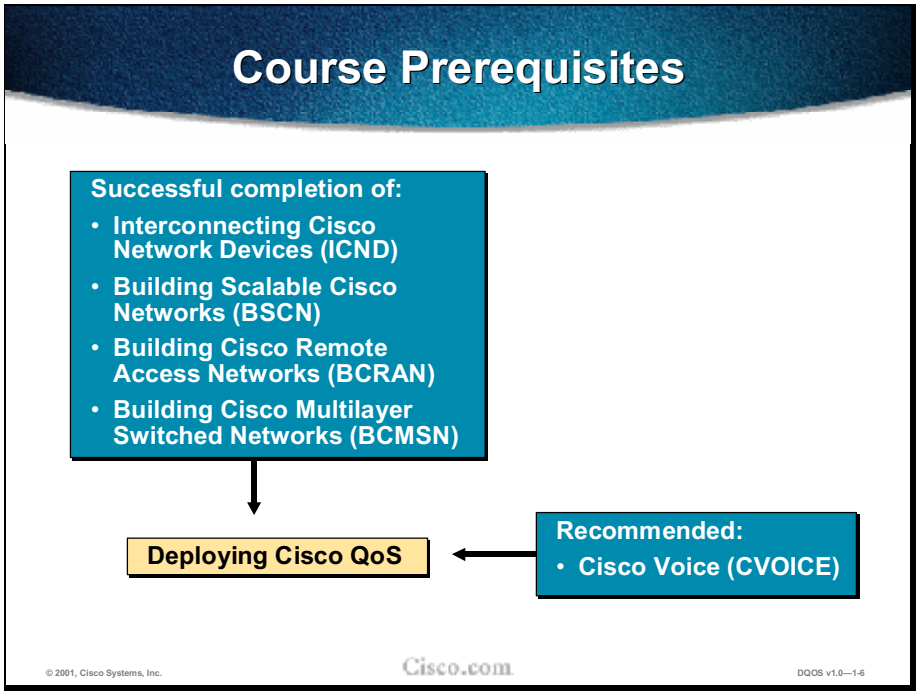
Course Objectives

This section lists the course objectives.



Prerequisites

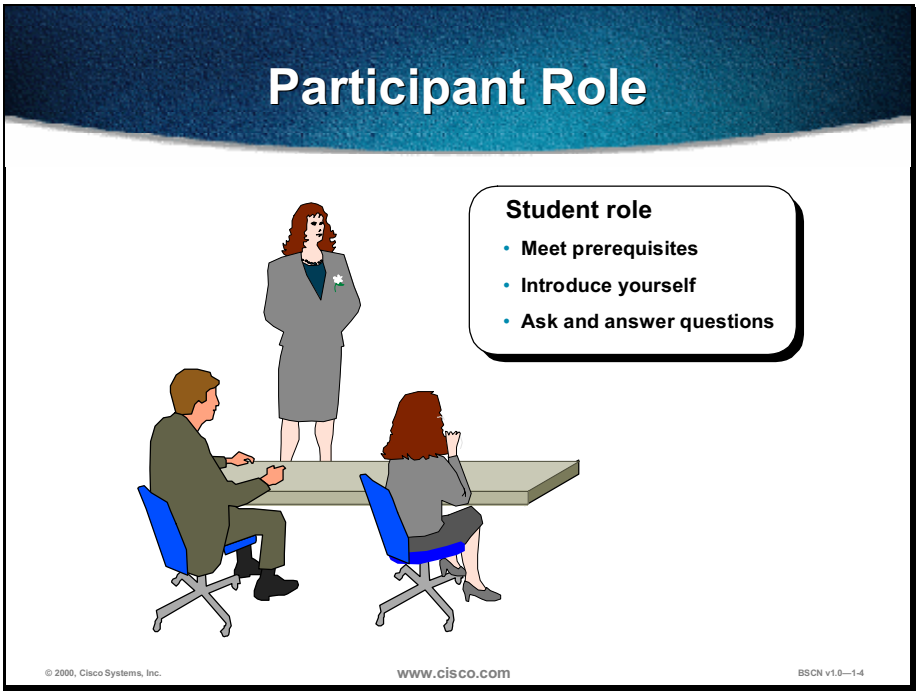
This section lists the course prerequisites.



You will be able to gain more practical experience from the course if you already have work experience and router configuration skills. These skills are best demonstrated through Cisco career certifications Cisco Certified Networking Professional (CCNP) or Cisco Certified Internetworking Expert (CCIE).

Participant Role

This section discusses your responsibilities as a student.

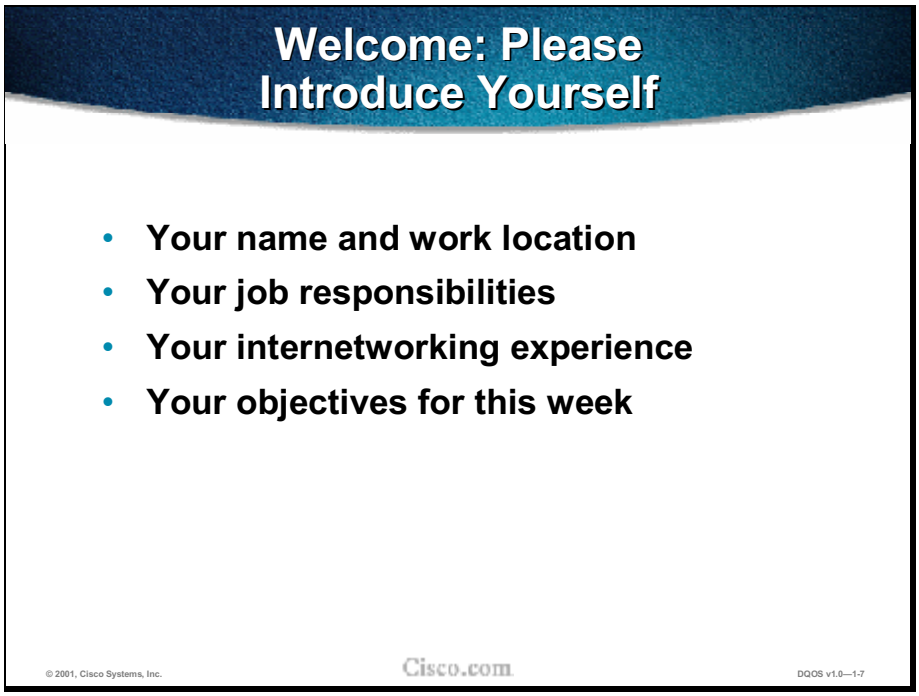


To take full advantage of the information presented in this course, you should meet the prerequisites for this class.

Introduce yourself to the instructor and other students who will be working with you during the five days of this course.

You are encouraged to ask any questions relevant to the course materials.

If you have pertinent questions concerning other Cisco features and products not covered in this course, please bring these topics up during breaks or after class, and the instructor will try to answer the questions or direct you to an appropriate information source.



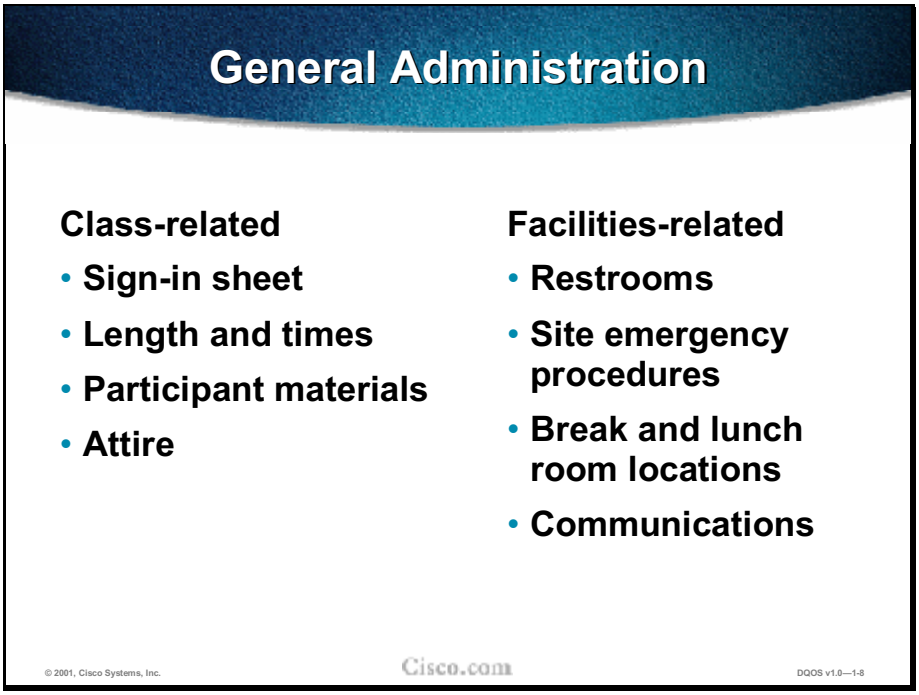
Introduce yourself, stating your name and the job function you perform at your work location.

Briefly describe what experience you have with installing and configuring Cisco routers and attending Cisco classes, and how your work experience helped you meet the prerequisites highlighted earlier.

You should also state what you expect to learn from this course.

General Administration

This section highlights miscellaneous administrative tasks that must be addressed.

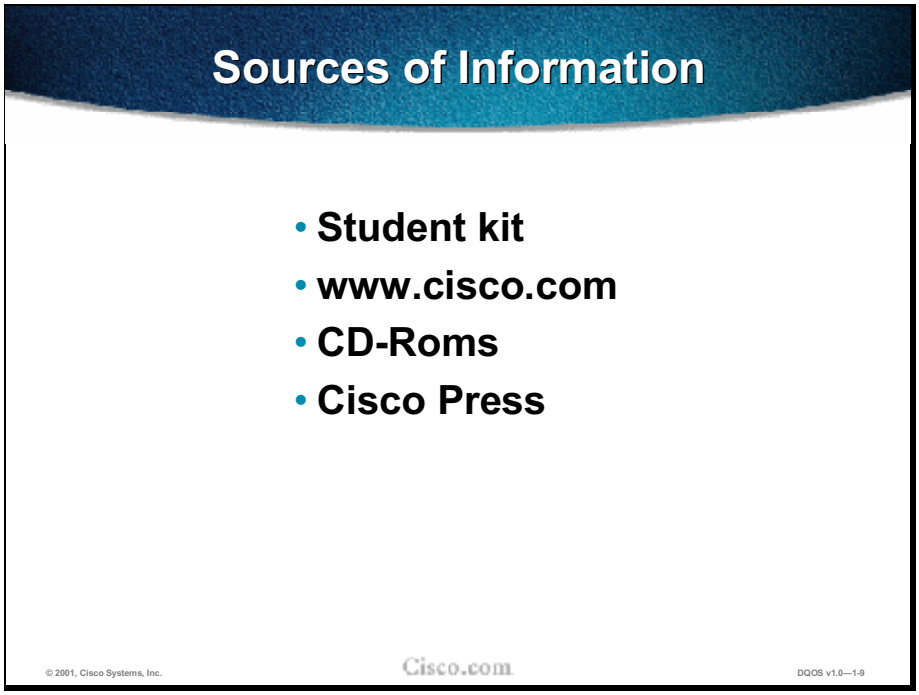


The instructor will discuss the administrative issues in detail so you will know exactly what to expect from both the class and facilities. The following items will be discussed:

- Recording your name on a sign-in sheet
- The starting and anticipated ending time of each class day
- What materials you can expect to receive during the class
- The appropriate attire during class attendance
- Rest-room locations
- What to do in the event of an emergency
- Class breaks and lunch facilities
- How to send and receive telephone, e-mail, and fax messages

Sources of Information

This section identifies additional sources of information.



Most of the information presented in this course can be found on the Cisco Systems website or on CD-ROM. These supporting materials are available in HTML format and as manuals and release notes.

To learn more about the subjects covered in this course, feel free to access the following sources of information:

- Cisco Documentation CD-ROM
- ITM CD-ROM
- *Cisco IOS 12.1 Configuration Guide*
- *Cisco IOS 12.1 Command Reference Guide*

Many of these documents can be found at the following URL:

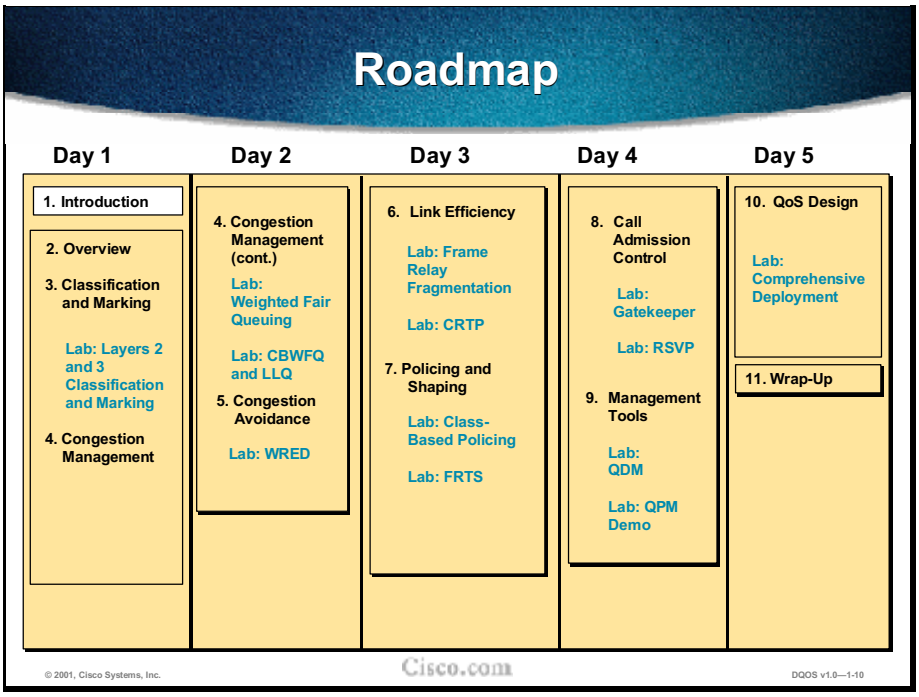
- http://www.cisco.com

Cisco Press books and documents can be found at the following URL:

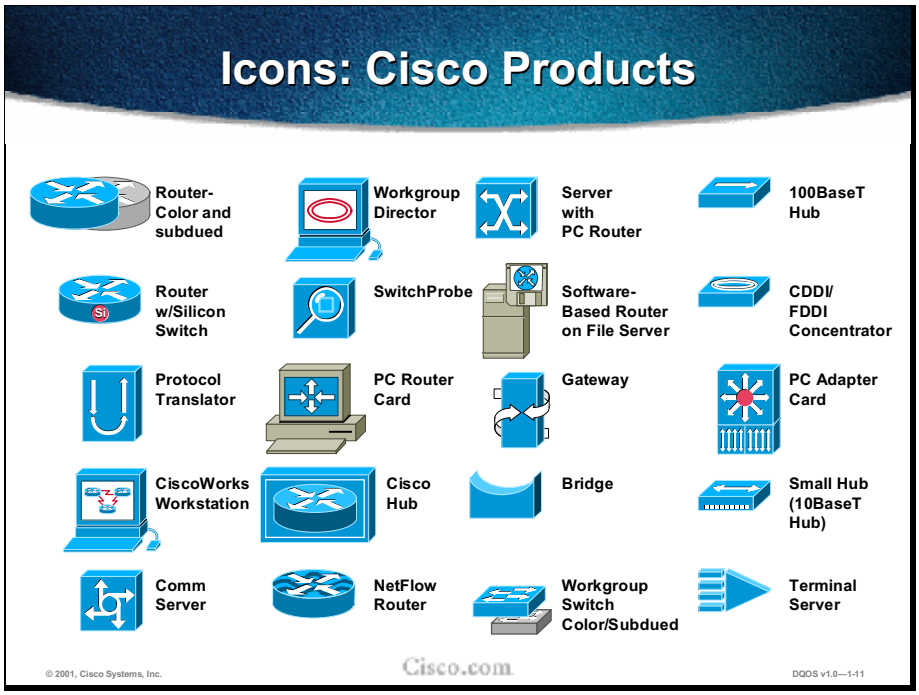
- http://www.ciscopress.com

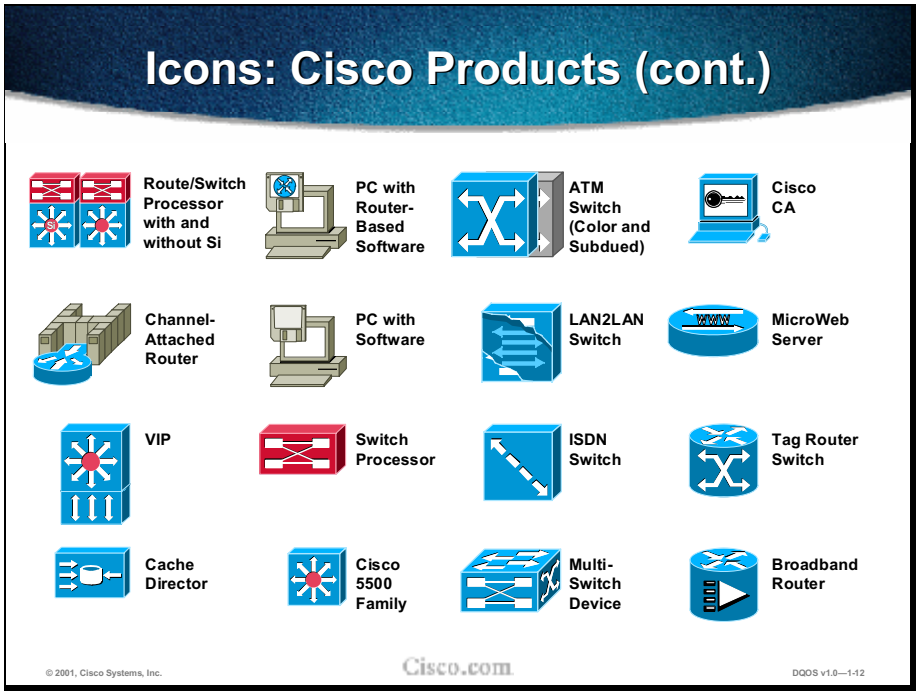
Course Syllabus

The recommended structure for this course is noted here. This structure allows enough time for your instructor to present the course information to you and for you to work through the laboratory exercises. The exact timing of the subject materials and labs depends on the pace of your specific class.



Cisco Icons


Icons: Cisco Products
Router-Color and subdued
Workgroup Director
Server with PC Router
100BaseT Hub
Router w/Silicon Switch
SwitchProbe
Software-Based Router on File Server
CDDI/FDDI Concentrator
Protocol Translator
PC Router Card
Gateway
PC Adapter Card
CiscoWorks Workstation
Cisco Hub
Bridge
Small Hub (10BaseT Hub)
Comm Server
NetFlow Router
Workgroup Switch Color/Subdued
Terminal Server
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—1-11


Icons: Cisco Products (cont.)
Route/Switch Processor with and without Si
PC with Router-Based Software
ATM Switch (Color and Subdued)
Cisco CA
Channel-Attached Router
PC with Software
LAN2LAN Switch
MicroWeb Server
VIP
Switch Processor
ISDN Switch
Tag Router Switch
Cache Director
Cisco 5500 Family
Multi-Switch Device
Broadband Router
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—1-12

Icons: Cisco Products (cont.)


NetRanger


NetSonar


Cisco 1000


ATM Router


NetFlow/ ATM Router


IP Standard and Old (Some Prefer)


Cisco 7505


Cisco 7507


Cisco 7500 (7513)

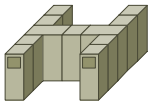

Multilayer Switch, with and without Text and Subdued

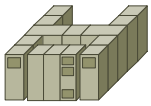

System Controller

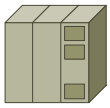

ATM Tag Switch Router


Cisco Directory Server

Icons: IBM


IBM Mainframe

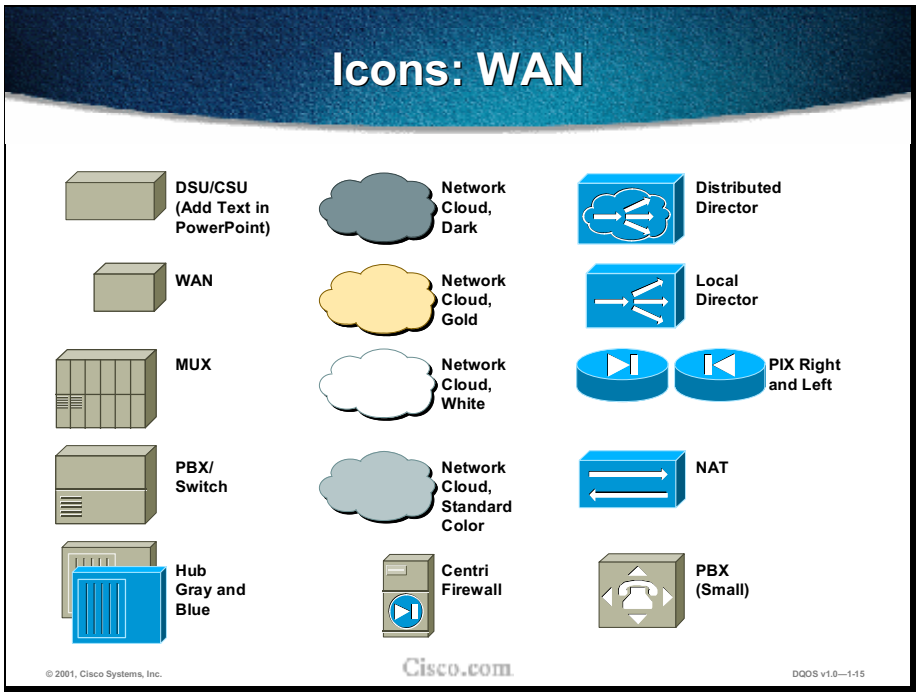

IBM Mainframe with FEP

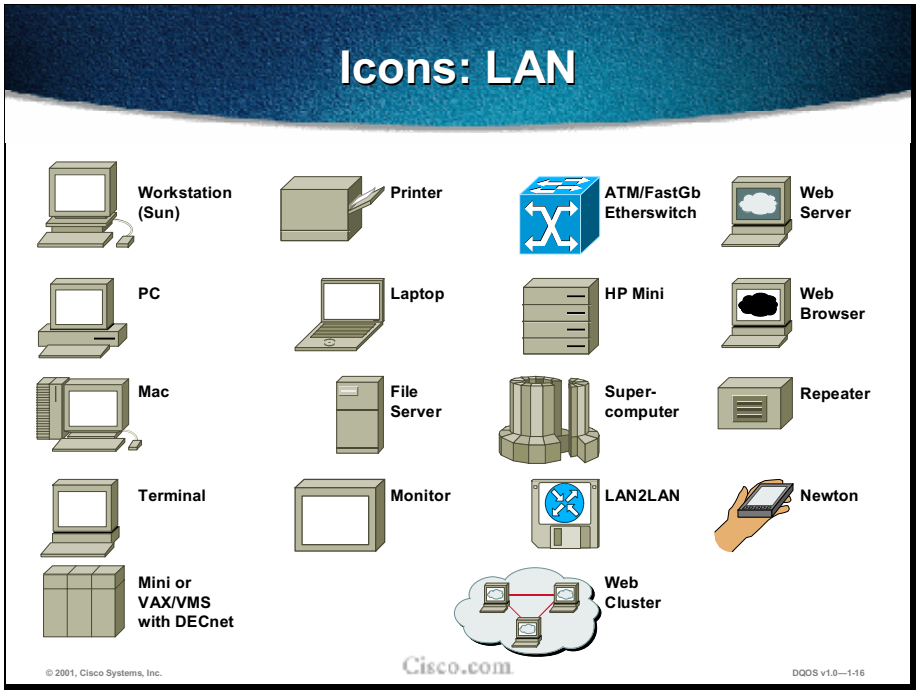

FEP (Front End Processor)


Cluster Controller/3174 (Desktop Model, Not Used Much)

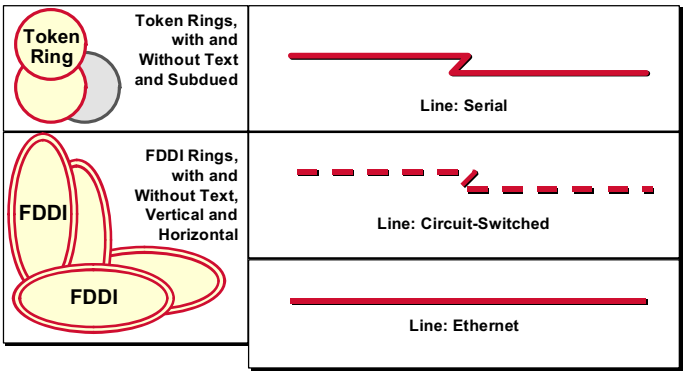

IBM Mini (AS400)


Cluster Controller/ 3274 or 3174 (Most Common Type)

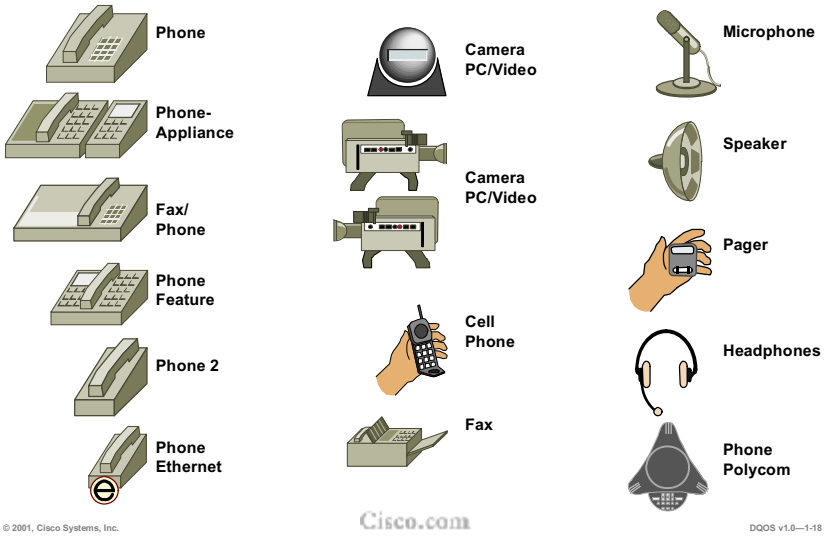

Icons: WAN
DSU/CSU (Add Text in PowerPoint)
WAN
MUX
PBX/ Switch
Hub Gray and Blue
Network Cloud, Dark
Network Cloud, Gold
Network Cloud, White
Network Cloud, Standard Color
Centri Firewall
Distributed Director
Local Director
PIX Right and Left
NAT
PBX (Small)
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—1-15


Icons: LAN
Workstation (Sun)
PC
Mac
Terminal
Mini or VAX/VMS with DECnet
Printer
Laptop
File Server
Monitor
ATM/FastGb Etherswitch
HP Mini
Super-computer
LAN2LAN
Web Cluster
Web Server
Web Browser
Repeater
Newton
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—1-16

Icons: Media



Icons: Multimedia/Voice/Phone



2

QoS Overview

Overview

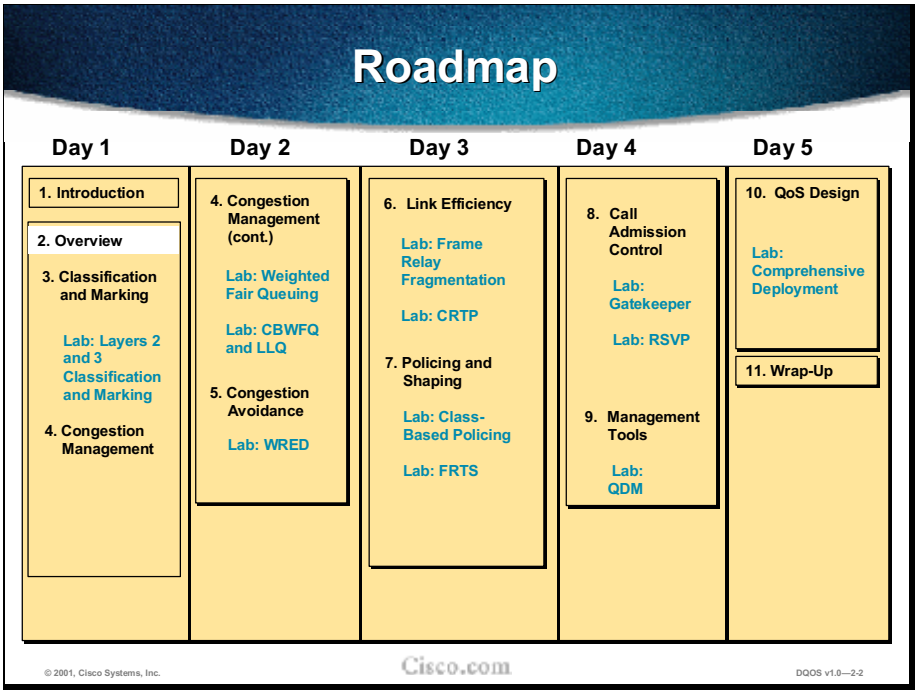
This chapter explains quality of service (QoS) and the range of QoS mechanisms that are defined within the QoS framework. The chapter ends with a brief introduction to the QoS models: Differentiated Services (DiffServ) and Integrated Services (IntServ). There are no labs in this chapter.

Objectives

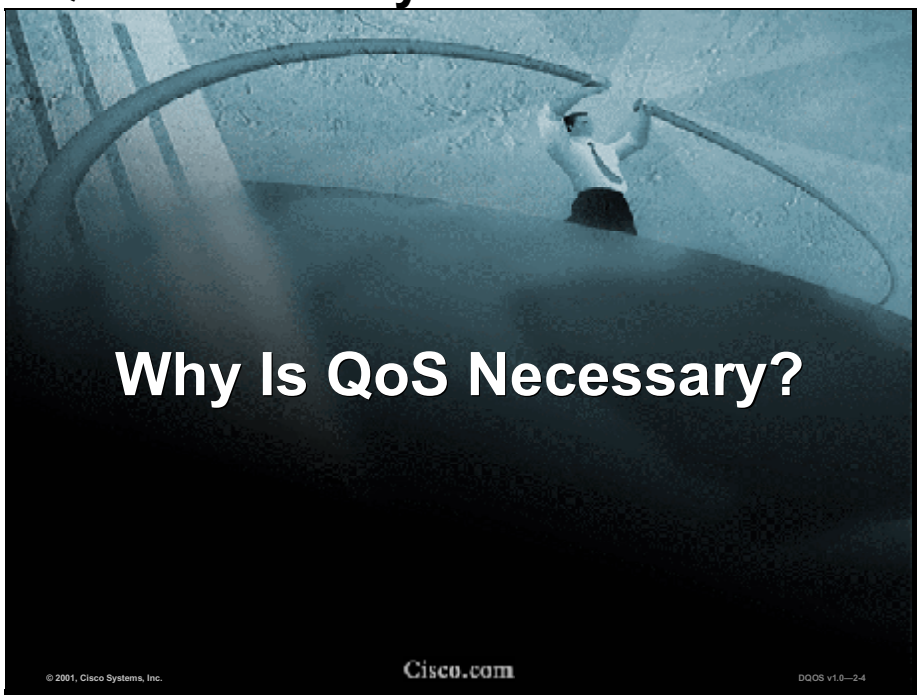
Upon completing this chapter, you will be able to:

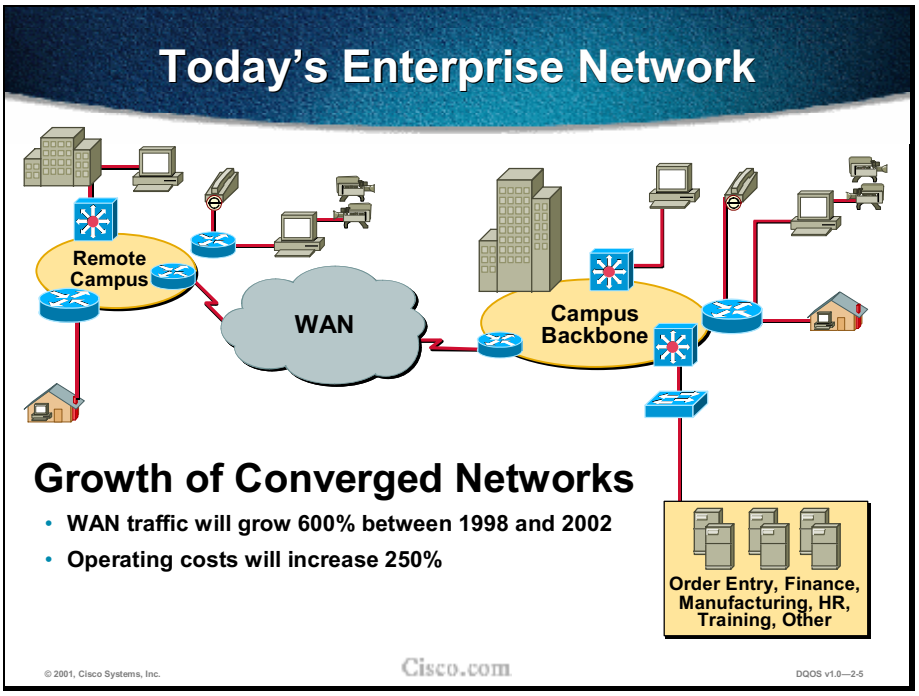
- List five benefits for implementing QoS in enterprise networks
- Describe how a converged network behaves without QoS
- Correctly describe the QoS framework
- Describe correctly what call admission control does
- Describe the difference between Integrated Services and Differentiated Services

Outline



Why Is QoS Necessary?

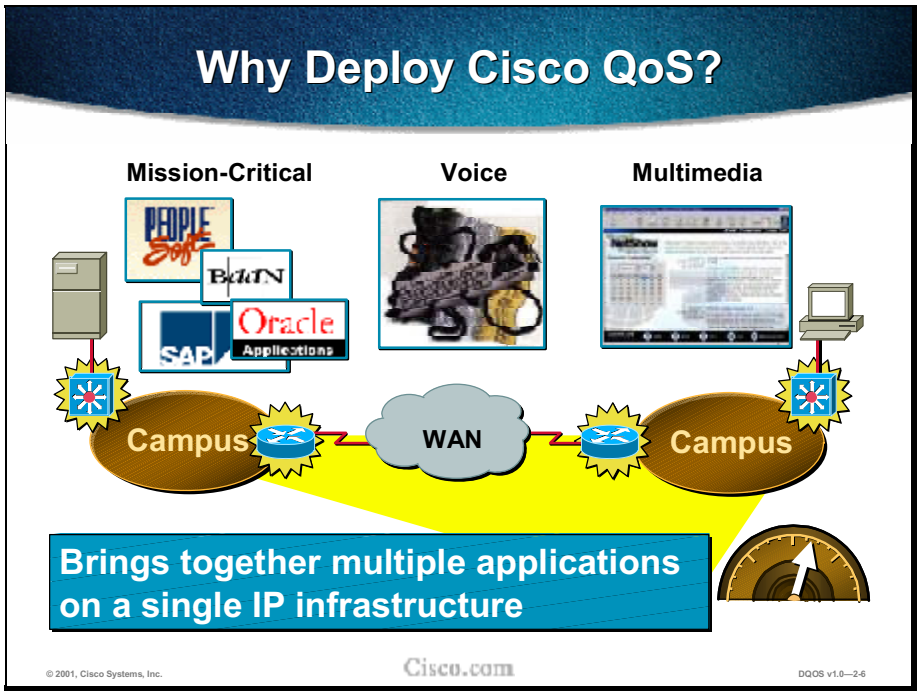


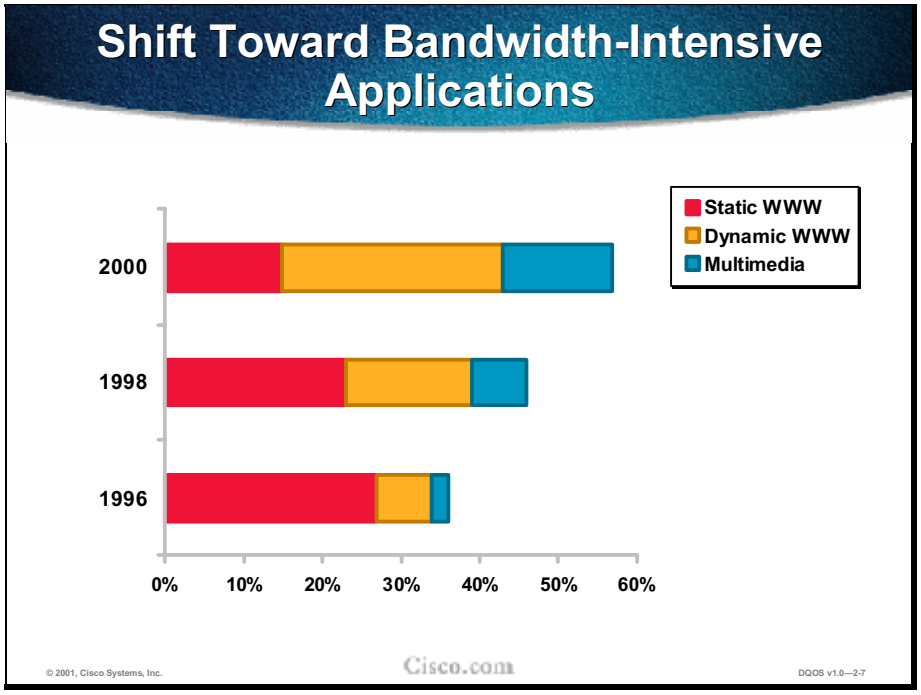


Today's enterprise network comprises multiple locations, server farms, remote access, and telecommuters. Increasing use of the network for mission-critical data, video, and Voice over IP requires a higher quality of service than data networks.

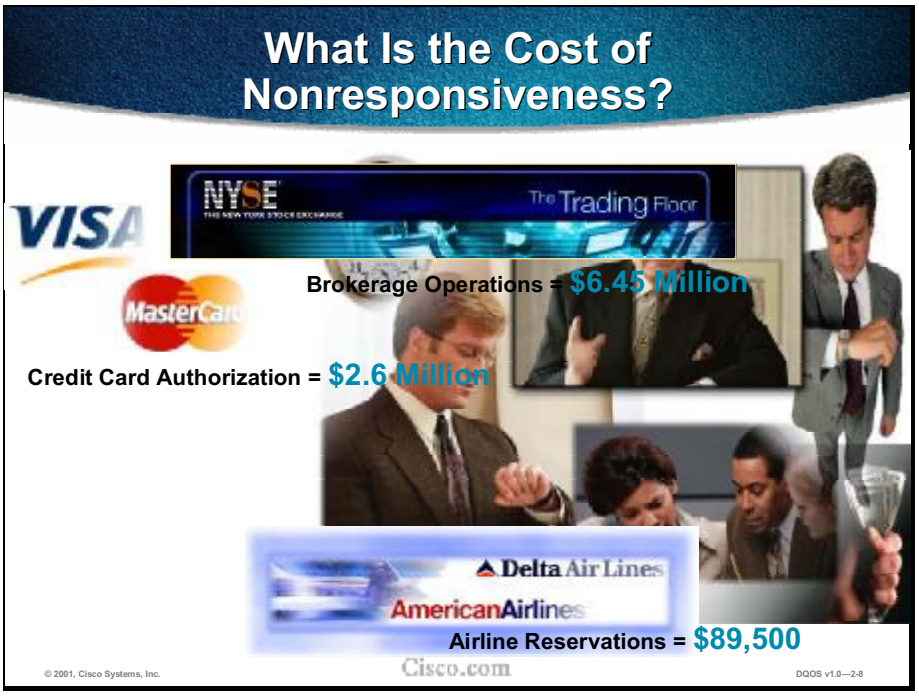
According to the Gartner Group, corporate WAN traffic is expected to grow approximately 600 percent between 1998 and 2002. Concurrent with this growth, operating costs are projected to increase by 250 percent. These operating-cost increases tax the ability to control expenditures. Network managers must balance budget constraints with the demand for more bandwidth, while accommodating traffic with different characteristics. Until now, they've addressed these differences with several parallel networks: one for voice, one for data, one for legacy mainframe applications, and one for video conferencing. Operating several networks both challenges management resources and can prove quite expensive.

Multiservice networking lets managers create a single, powerful network that is far easier and less expensive to manage and operate. But the benefit does not stop there. Consolidating data, voice, and video networks into a converged network enables managers not only to bring spiraling costs under control, but it also gives companies a new way to do business—a better way to compete. With a multiservice network, companies can develop and deploy a whole new class of powerful applications never before possible with separate networks.

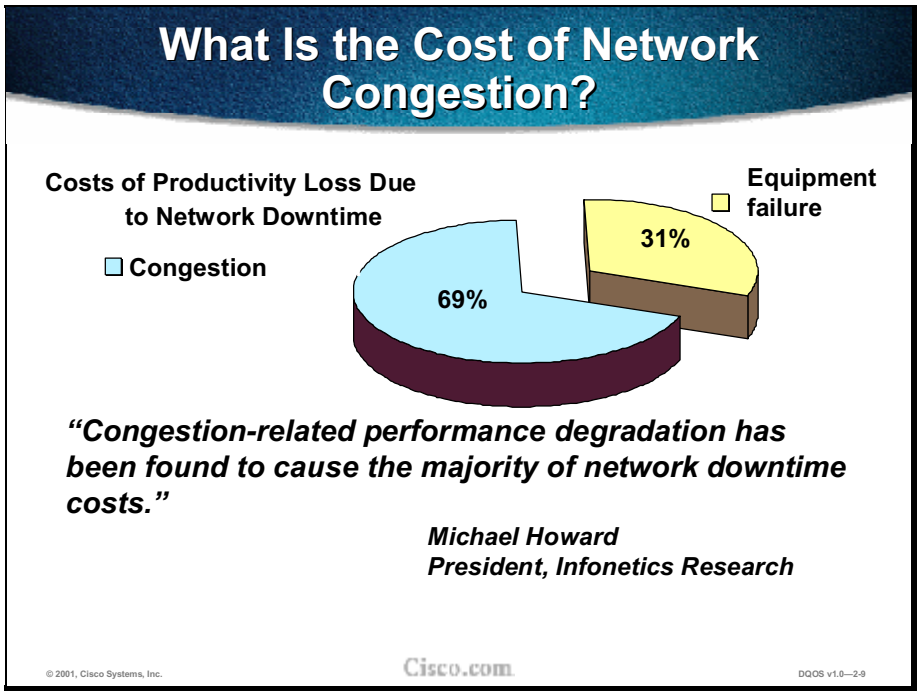




Bandwidth-intensive applications such as video-streaming, real-time audio, video conferencing, interactive communication, and peer-to-peer applications (Napster) increase the demand for high-speed networks.

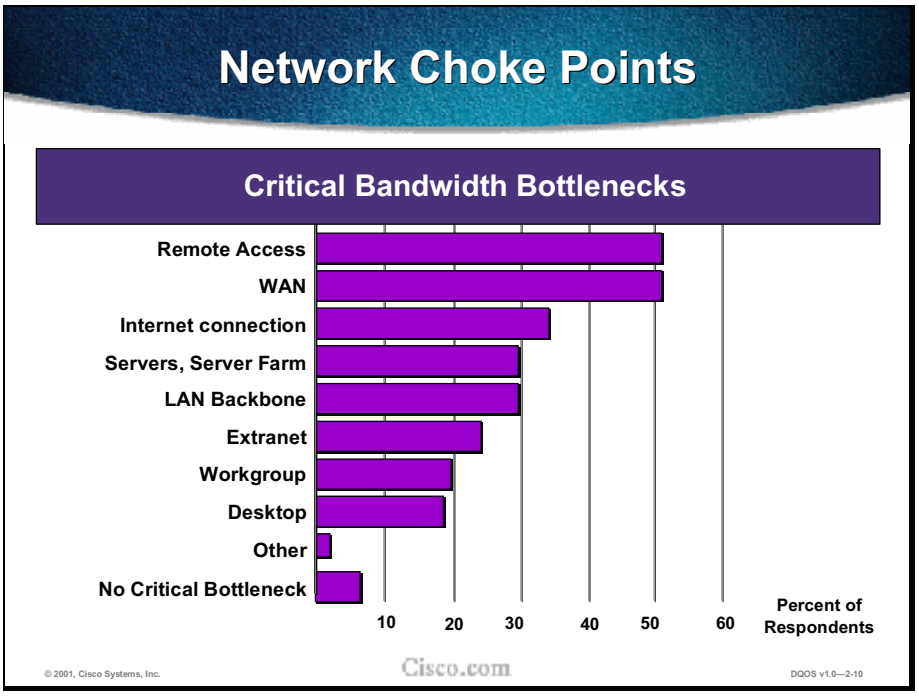


When networks are burdened with too much traffic, they become unreliable and unavailable. The consequence of network downtime is significant for businesses. There is always a cost in having networks go down, no matter what the business is. The figure above shows three examples of the cost of one hour of downtime. Brokerage operations can lose $6.45 million, credit card authorization services can lose $2.6 million, and airline reservations can lose $89,500.

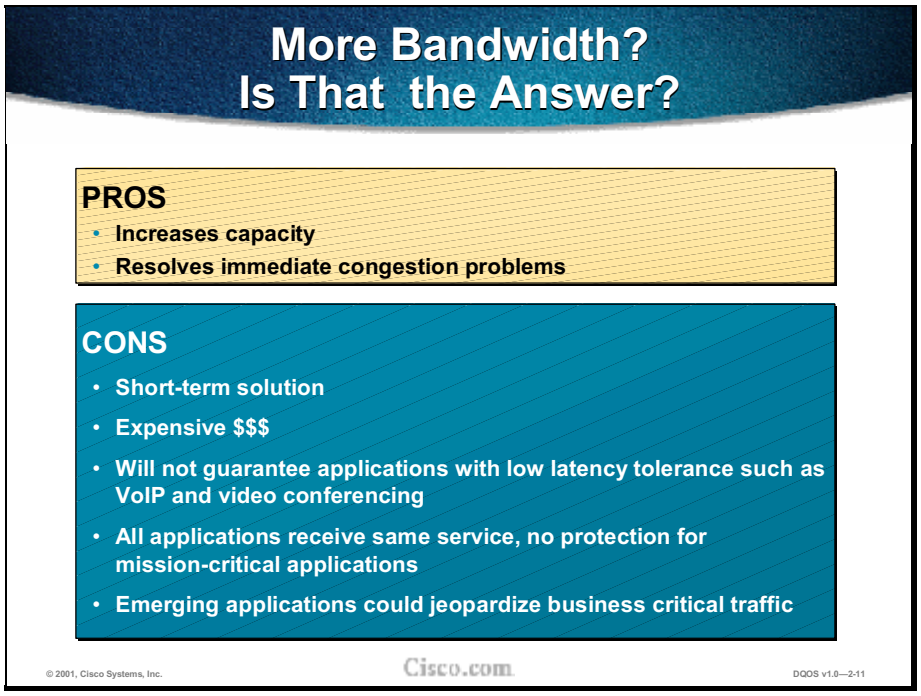


This slide shows that the source of network downtime is both equipment failure and congestion on the networks. In 1997, congestion on a data network cost more than twice as much in productivity loss than in equipment failure.

What does congestion look like? Consideration of the behavior of congested systems is not simple and cannot be dealt with in a simplistic manner, as traffic rates do not simply rise to a level, stay there a while, then subside. Periods of traffic congestion can be quite long, with losses that are heavily concentrated. A slight increase in the number of active connections can result in a large increase in the packet loss rate. This understanding of the behavior of congested networks suggests that because the level of busy-period traffic is not predictable, it would be difficult to efficiently size networks to reduce congestion adequately. Observers of network congestion report that in reality, traffic "spikes," which causes actual losses that ride on longer-term ripples and they in turn ride on still longer-term swells.

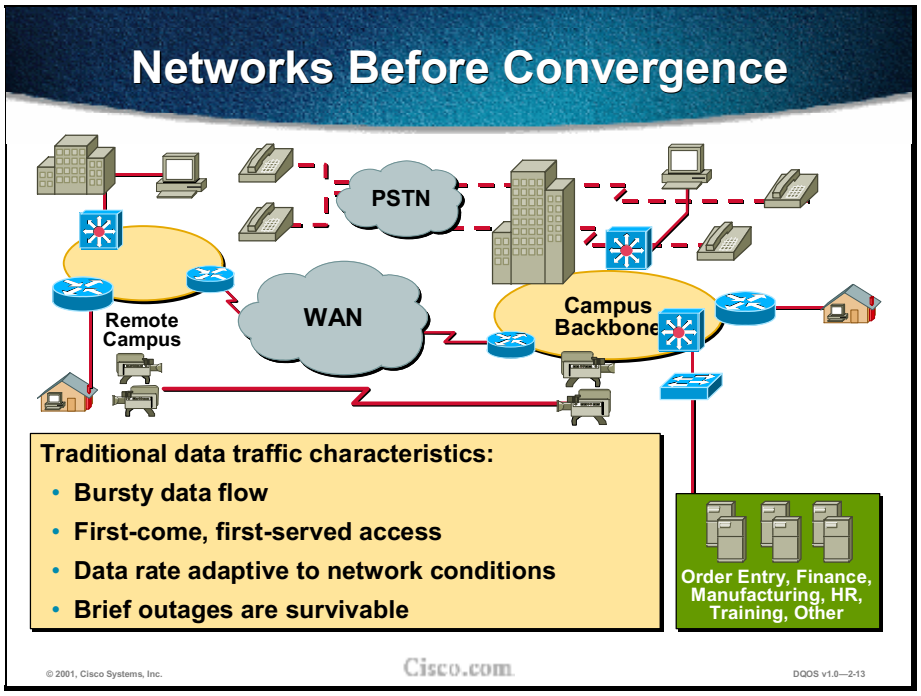


This is another way to look at the problem of increasing data traffic. When CIOs were asked in an *Information Week* survey (May 1999), "What are the critical bandwidth bottlenecks for your organization's network?" the top three responses were, "Remote access, WAN, and Internet connection." Each of these problems can be addressed by the deployment of the proper QoS tools.



Some might argue that the simple answer to resolving the problems of congestion and demand for increased capacity is to add bandwidth.

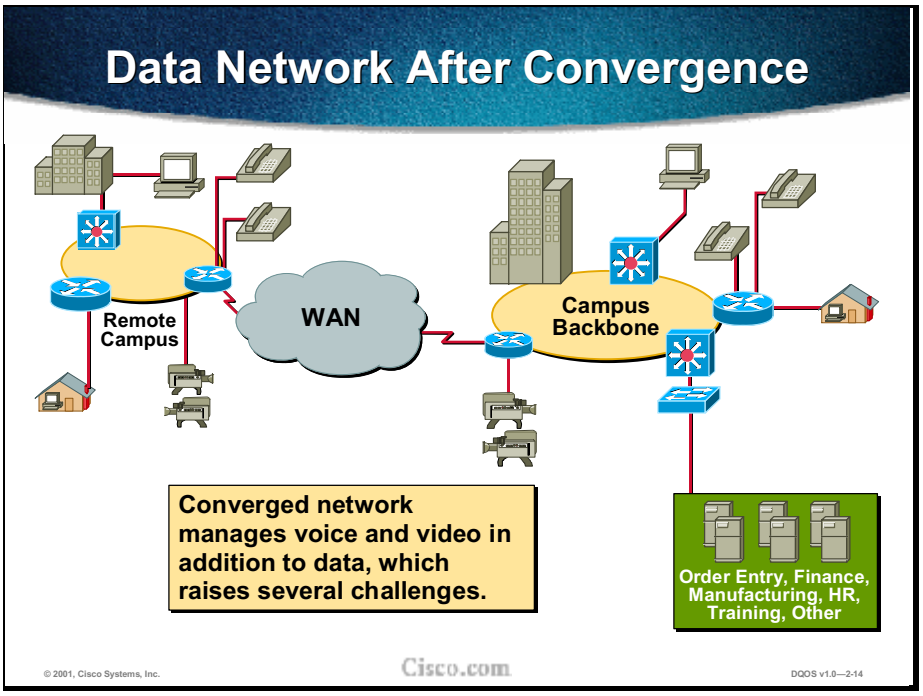
There are several problems with this approach: More bandwidth is often only a short-term solution. If users have more bandwidth, they use more bandwidth, and the mission-critical stuff or applications requiring low latency are back where they started. Bandwidth is expensive. Increased bandwidth does not guarantee that voice and video applications will perform effectively over data networks. Nor does it guarantee that mission-critical applications will get through. Nonessential applications such as Napster can jeopardize business-critical traffic when all traffic is treated in the same manner.

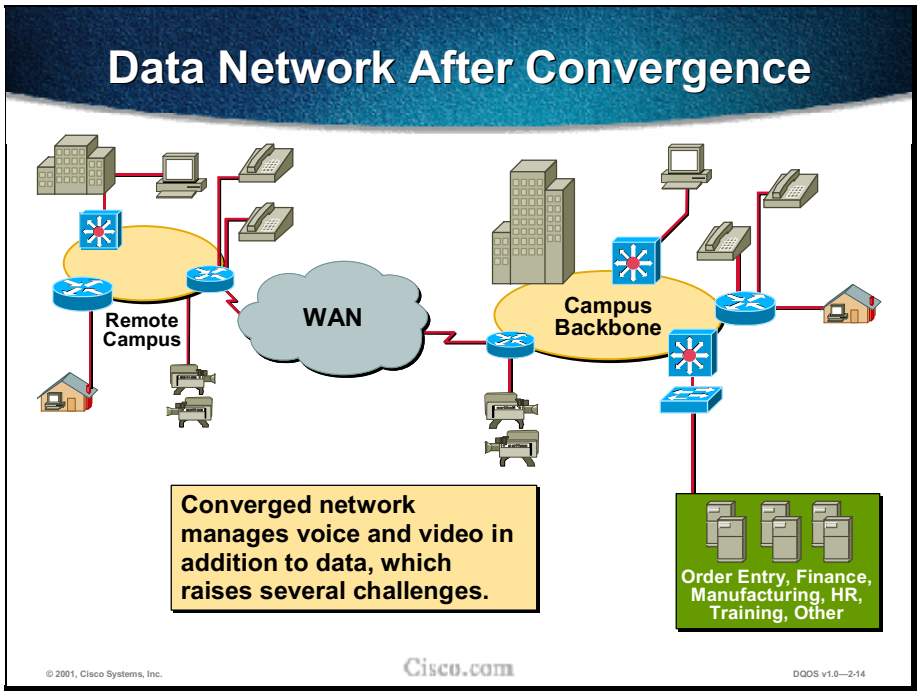


Historically, network engineering has been focused on connectivity. The rates at which data come onto the network resulted in bursty data flows. The data arrives in packets and tries to grab as much bandwidth as it can at any given time. The access is very egalitarian; it's a first-come, first-served basis, so whoever gets there first gets the bandwidth.

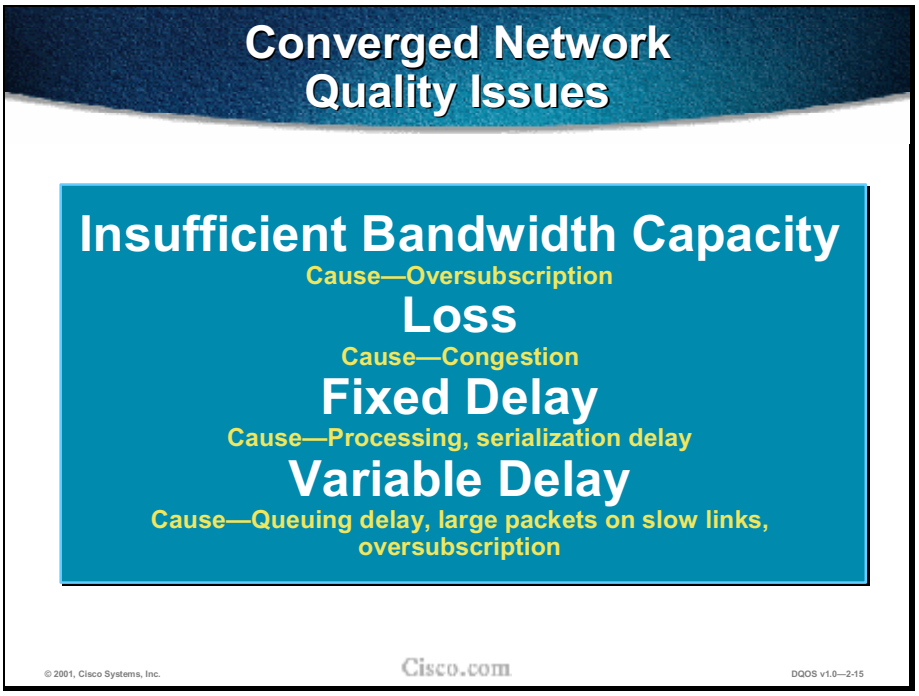
As a result of this somewhat anarchic way of attacking the network, the data rate is adaptive to network conditions.

The protocols that have been developed adapt to the bursty nature of data networks, and brief outages are survivable. Typically, if retrieving e-mail, a delay of a few seconds is generally not noticeable. A delay of minutes is annoying, but not serious.





The graphic above shows a converged network. The problems that arise in managing a converged network are listed in the next figure.



The big four problems facing enterprise networks are bandwidth capacity, loss, fixed delay, and variable delay. Variable delay is also called jitter.

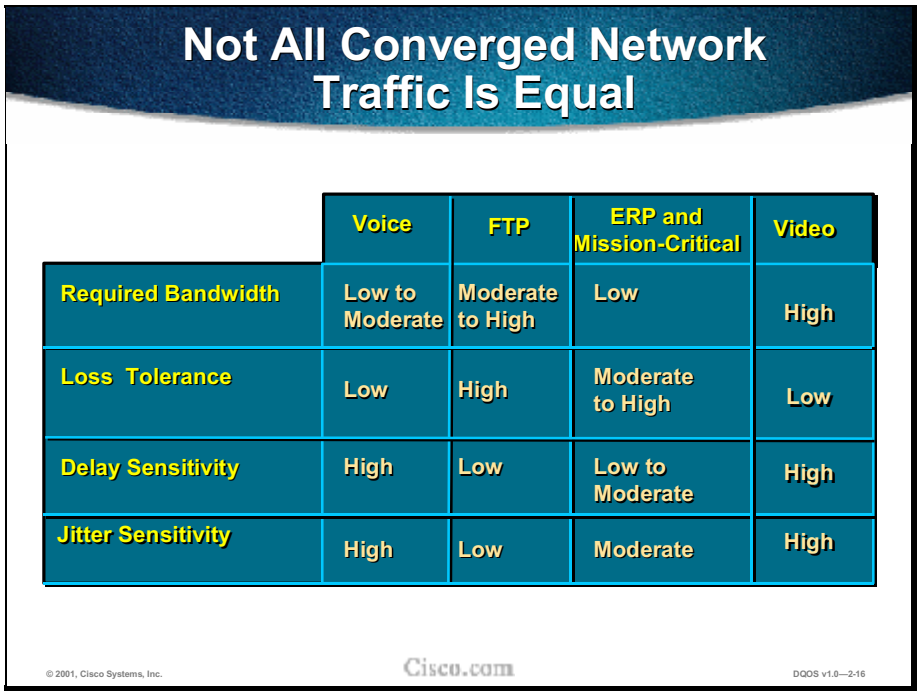
Large graphic files, multimedia uses, and increasing use for voice and video cause bandwidth capacity problems over data networks.

Loss of packets is usually caused by congestion in the WAN, resulting in speech dropouts or a stutter effect if the play-out side tries to accommodate by repeating previous packets.

Delay is the time it takes for a packet to reach the receiving endpoint after being transmitted from the sending endpoint. This time is termed the “end-to-end delay,” and it consists of two components: fixed network delay and variable network delay. Jitter is the delta, or difference, in the total end-to-end delay values of two voice packets in the voice flow.

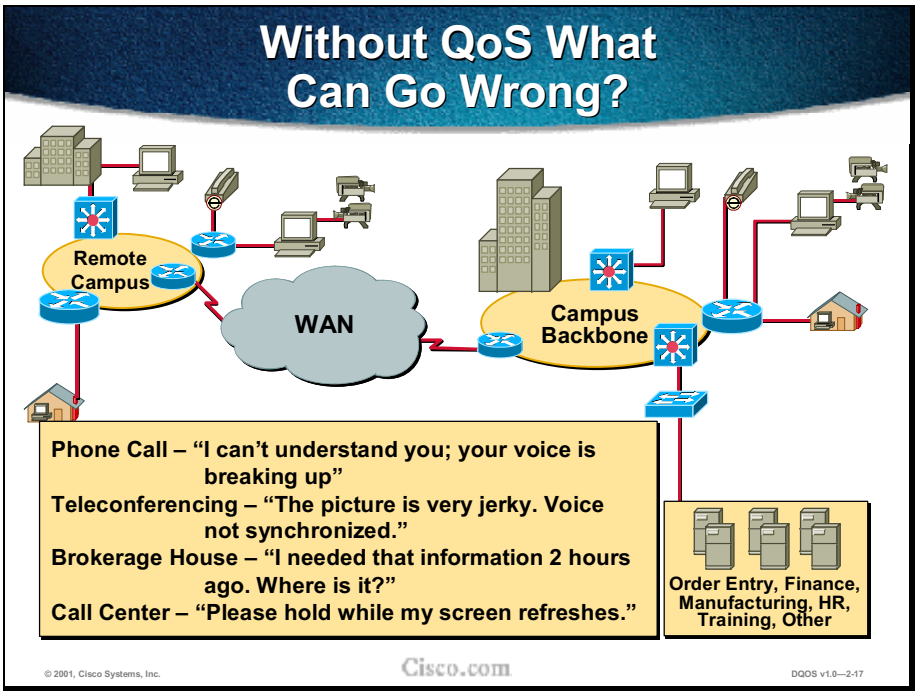
Two types of fixed delay are processing and serialization delays. A processing delay is the time required by a networking device to look up the route, change the header, and complete other switching tasks. In some cases, the packet also must be manipulated. For example, the encapsulation type or the hop count must be changed. Each of these steps can contribute to the processing delay.

Serialization is the process of placing bits on the circuit. The higher the circuit speed, the less time it takes to place the bits on the circuit. So the higher the speed, the less serialization delay.

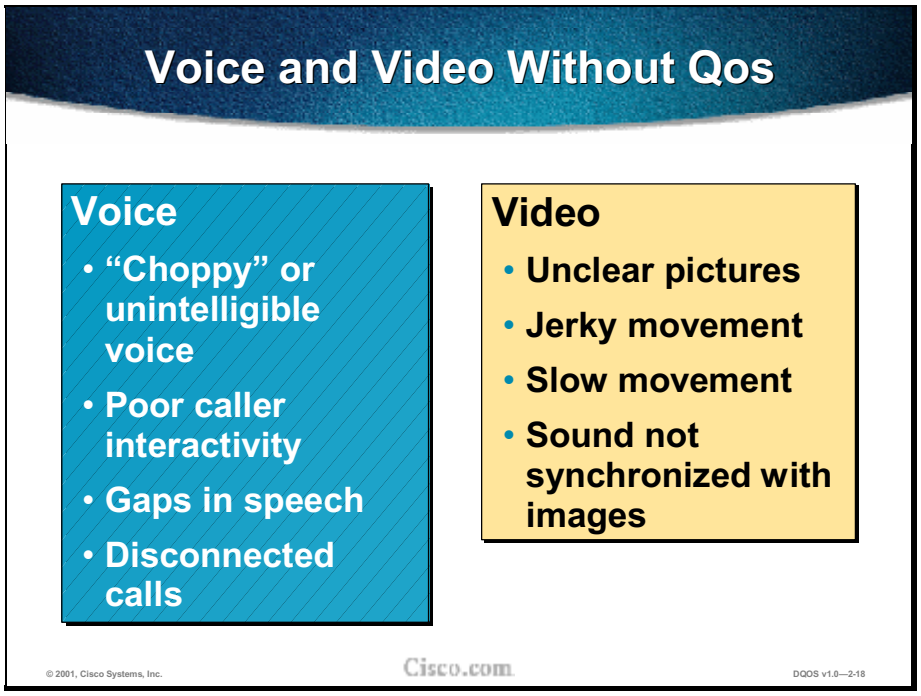


This slide compares the tolerance of different applications: voice, FTP (File Transfer Protocol), ERP (enterprise resource planning) and mission-critical, and video, to the demands of network traffic: bandwidth demand, packet loss, delay, and jitter.

Examination of this table shows that each type of network traffic has its own needs. Voice and video transmission are both highly sensitive to delay and jitter, but video requires much more bandwidth than voice. With different types of traffic on the converged network, QoS tools allow the network manager to treat the different types of traffic differently.



This slide shows an everyday enterprise network with concrete examples of what can go wrong. For telephone calls, the voice quality can be poor. Parts of sentences can be lost, the voice can be shaky, and there can be an echo on the line. For teleconferencing, the picture can be jerky and unfocused. Sound transmission may not be coordinated with the visual transmission. For mission-critical applications, transmissions can be delayed or lost when networks become congested.



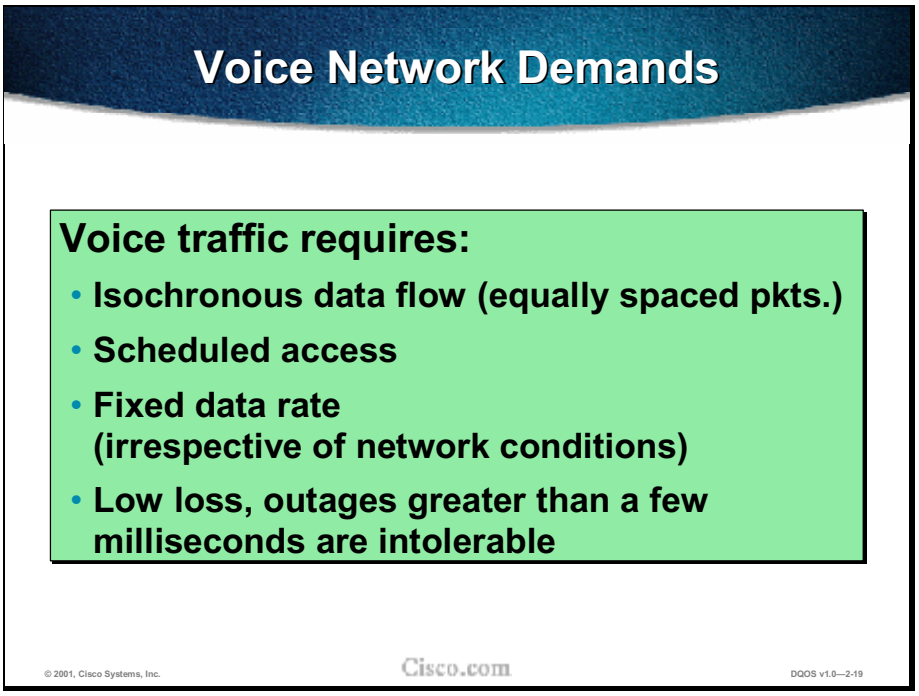
What happens if a data network is not prepared for voice and/or video to be run over it?

With inadequate preparation of the network, voice transmission is choppy or unintelligible. Gaps in speech are particularly troublesome where pieces of speech are interspersed with silence, and speech literally disappears. In voice-mail systems this silence is a problem. For example, you dial 68614. In a situation where the gaps in speech are actually gaps in the tone, 68614 becomes 66881114, because the gaps in speech are perceived as pauses in the touch-tones.

Poor caller interactivity is the consequence of delay. It causes two problems—echo and talker overlap. Echo is caused by the signal reflections of the speaker's voice from the far-end telephone equipment back into the speaker's ear. Talker overlap (or the problem of one talker stepping on the other talker's speech) becomes significant if the one-way delay becomes greater than 250 milliseconds. If bad, calls go to "walkie-talkie" mode.

Disconnected calls are the worst cases. If there are long gaps in speech, people hang up, or if there are signaling problems, calls are disconnected. Such events are completely unacceptable in the voice world yet are quite common for an inadequately prepared data network that's attempting to carry voice.

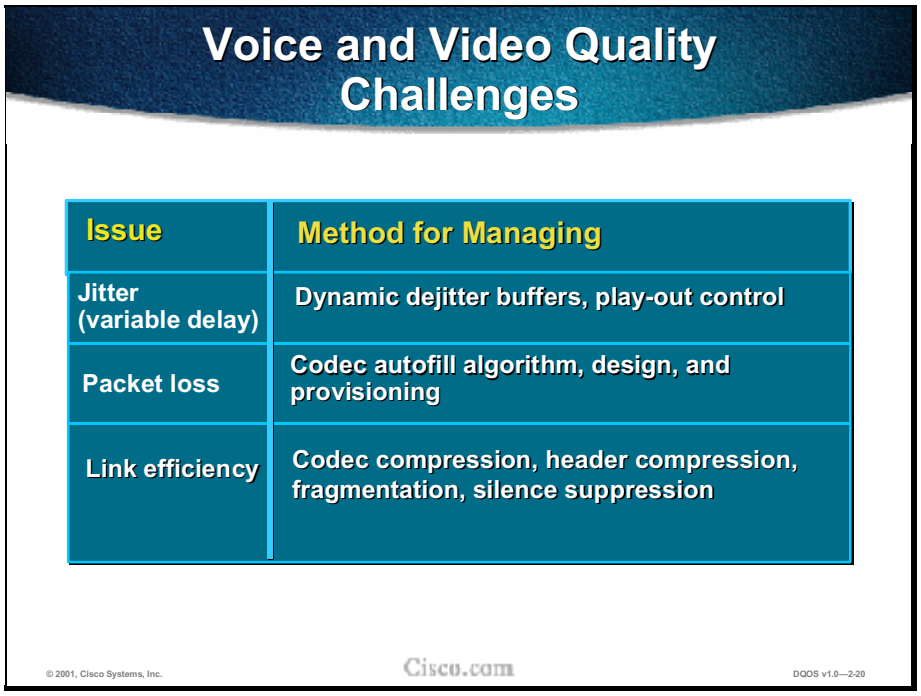
Multimedia streams, such as those used in IP telephony or video conferencing, may be extremely sensitive to delivery delays, creating unique quality-of-service demands on the underlying networks that carry them. When packets are delivered using the "best-effort" delivery model, they may not arrive in order, in a timely manner, or at all. The result is unclear pictures, jerky and/or slow movement, and sound out of synchronization with the image.



When voice is added to a data network, the equation changes. In voice/data networks, the data flow is isochronous, which means that packets are equally spaced. Voice data flow needs a specific amount of bandwidth, and it needs it at a specific time. Therefore access is scheduled, and it needs to be guaranteed. In traditional networks, access is arranged in time slots (time division multiplexing–TDM), and there is specific reserved bandwidth for voice data.

The data rate is not adaptive to network conditions. It needs how much it needs and it needs it then, and that's all there is to it.

Outages of greater than a few milliseconds (ms) are simply intolerable. Silences are not acceptable on phone calls nor is waiting for the data network to come back in order for voice to be available.



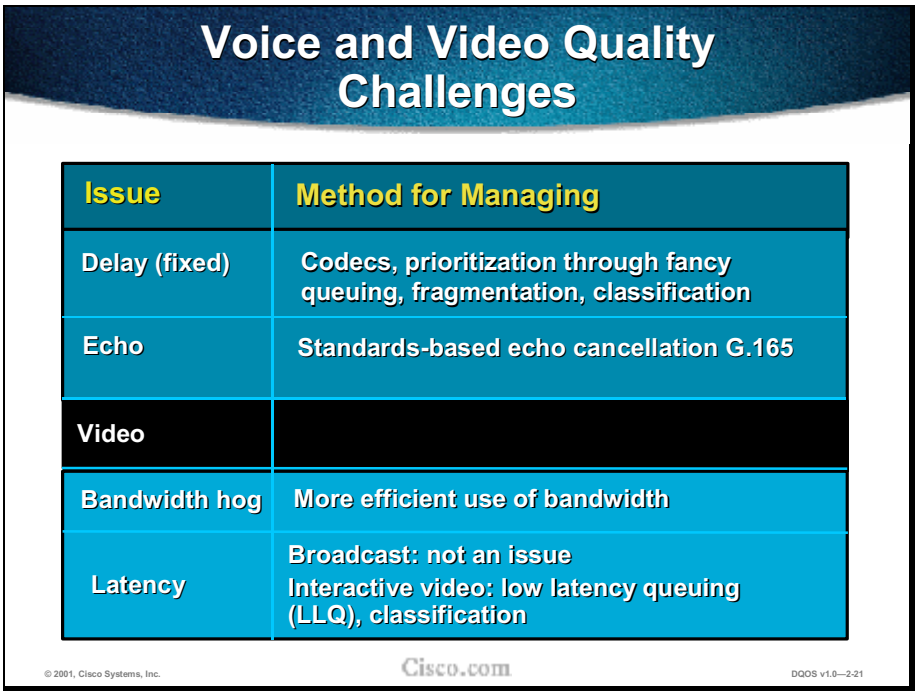
This slide lists three demands of voice and video over data lines. Some of the solutions listed here are not QoS solutions.

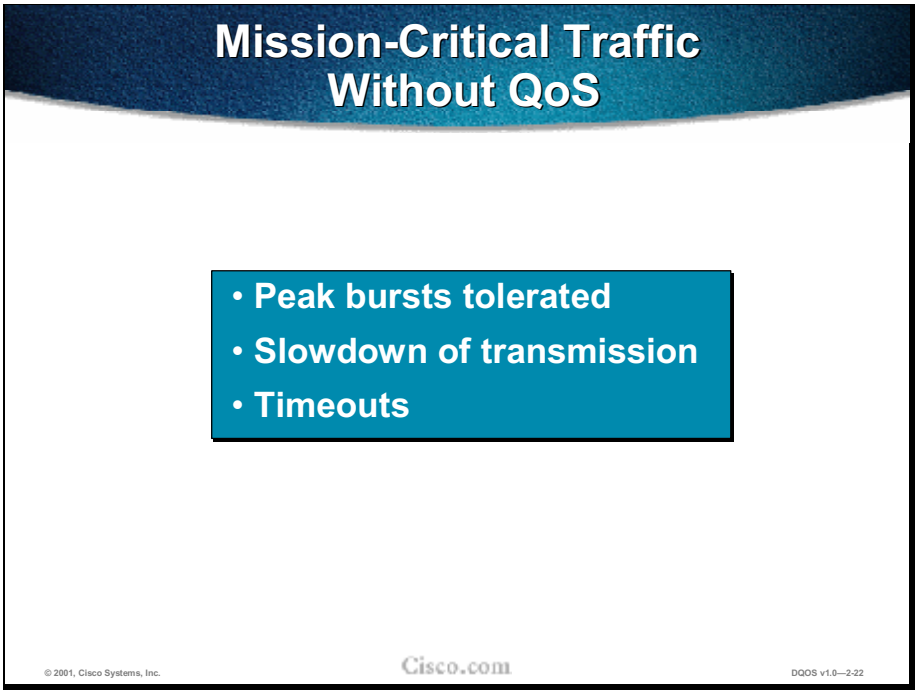
QoS Solutions—These solutions will be taught in the context of this class.

- RTP
- Queuing
- Fragmentation
- Precedence
- Header compression

Non-QoS Solutions—These solutions will not be taught in the context of this class.

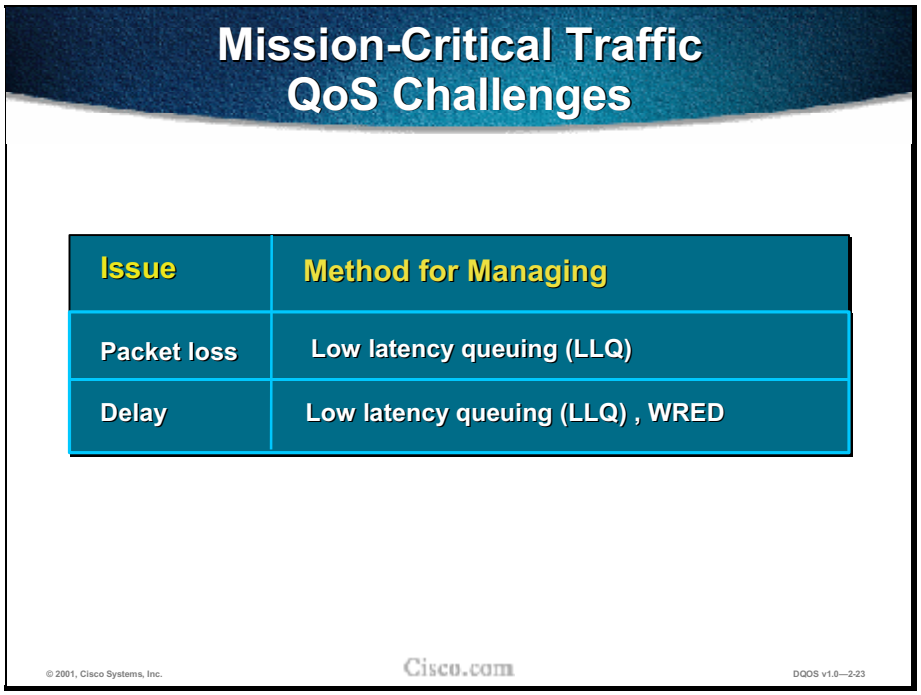
- Standards-based echo cancellation
- Codecs–Cisco IOS supports a list of codecs
- Silence suppression
- Dynamic dejitter buffers
- Play-out control
- Codec autofill algorithm



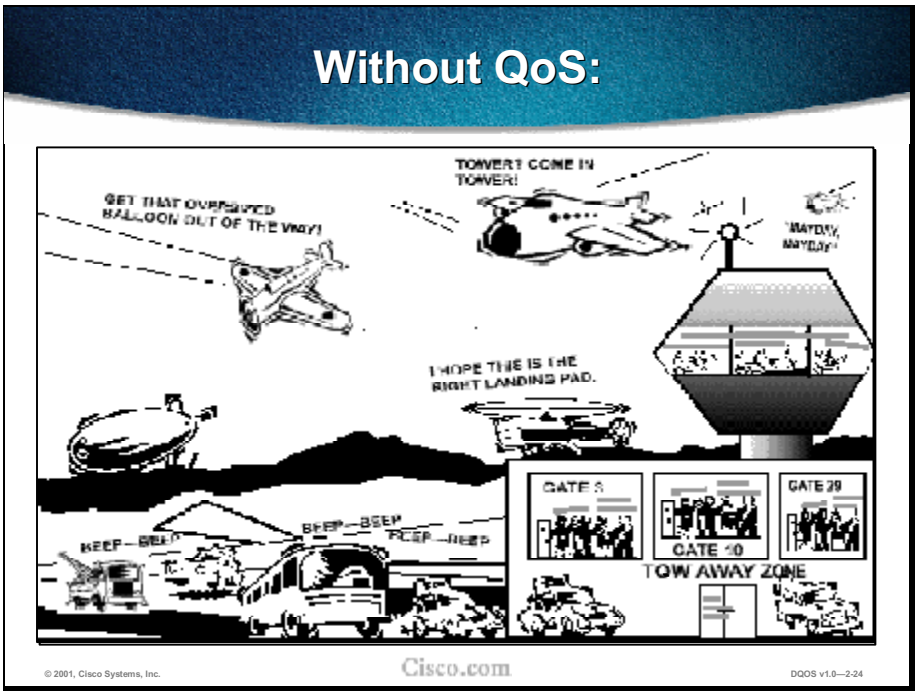


On enterprise networks mission-critical applications compete for bandwidth with network uses that are of lesser importance. Applications such as Oracle, SAP, and PeopleSoft can tolerate peak bursts that transcend capacity; however, other behaviors of the data transmission can be a problem for the data network. For instance, under Transmission Control Protocol (TCP), when the round-trip of a packet exceeds a certain threshold, the applications slow down the transmission, reducing traffic on the network.

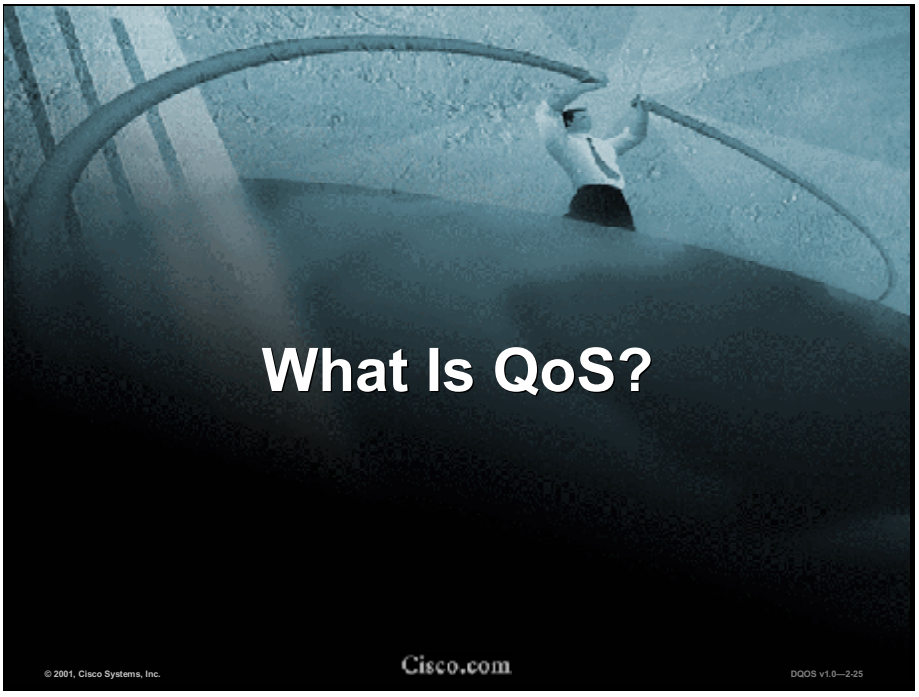
Another problem is transmission delays, which occur when there is packet loss. In TCP a small number of packet drops caused by local congestion signal to TCP that it should reduce its transmission speed. When several message segments are lost simultaneously however, TCP cannot determine how to recover, and it waits for a timeout before trying. Because timeouts can last for seconds, TCP operating in such environments tends to move very slowly.

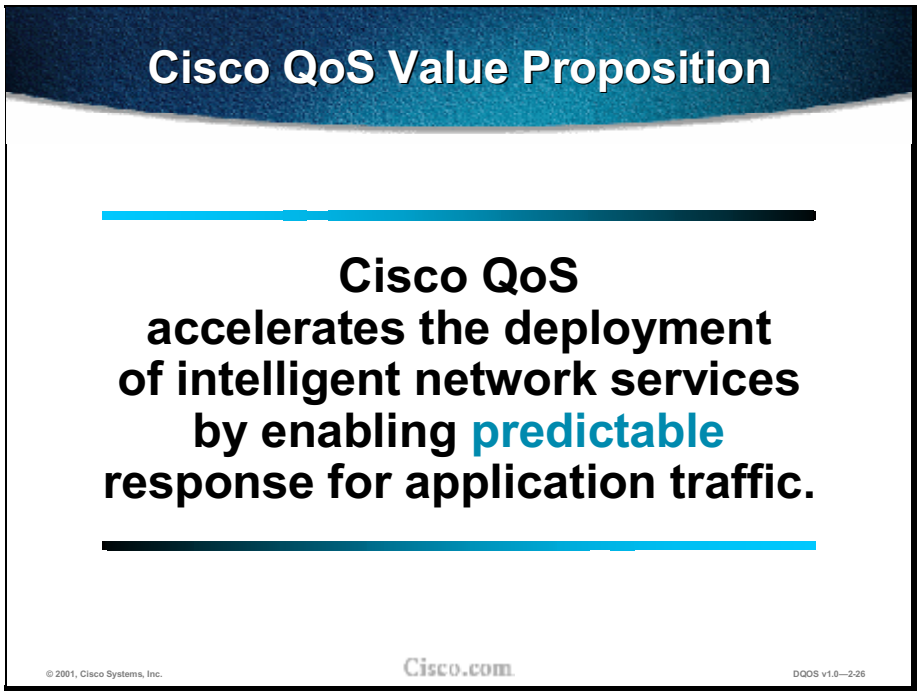


Enterprise resource planning (ERP) has fewer demands than voice and video. What is important to mission-critical applications is packet loss and delays. The “weapons” for managing these threats are low latency queuing(LLQ) and weighted random early detection (WRED).



An airport without QoS.



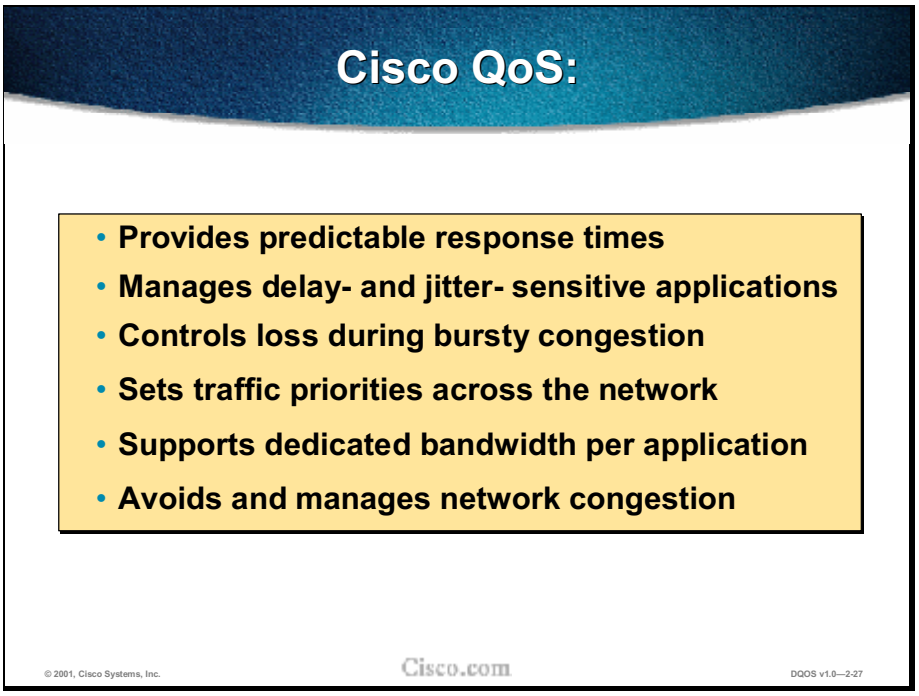


The IOS QoS features enable networks to control and predictably service a variety of networked applications and traffic types, allowing network managers to take advantage of a new generation of media-rich and mission-critical applications.

The goal of QoS is to provide better and more predictable network service by providing dedicated bandwidth, controlled jitter and latency, and improved loss characteristics. QoS achieves these goals by providing tools for managing network congestion, shaping network traffic, using expensive wide-area links more efficiently, and setting traffic policies across the network. QoS offers intelligent network services in the right places, just enough in each host, switch, or router.

Parekh and Gallagher, Infocomm ’93, defined what is meant by predictable:

- One must have at most a predictable amount of traffic in the network.
- One must have a predictable amount of delay in each network element.
- Given these, end-to-end delay of a host-to-host message is predictable.

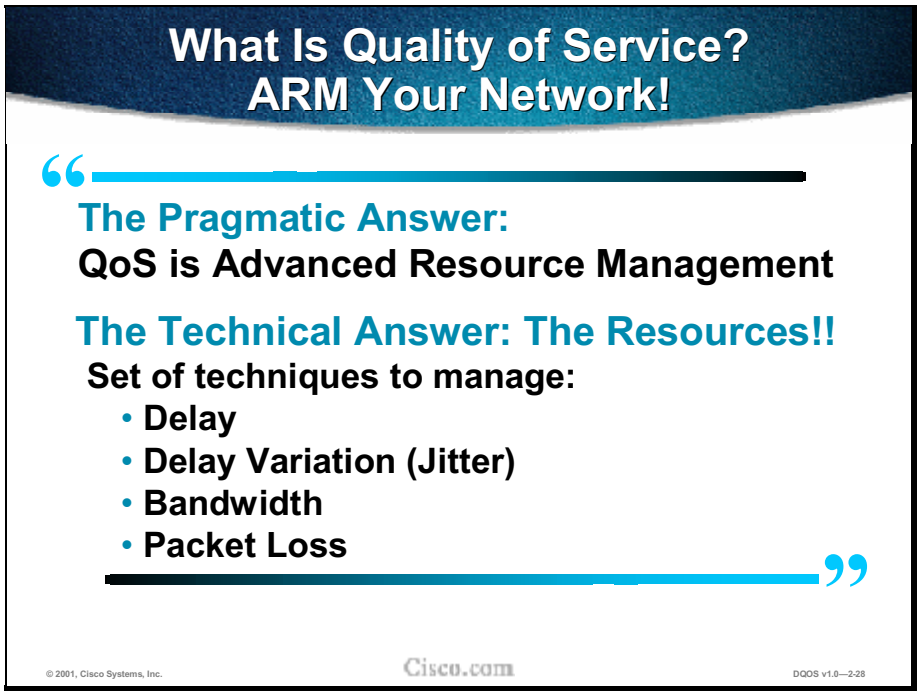


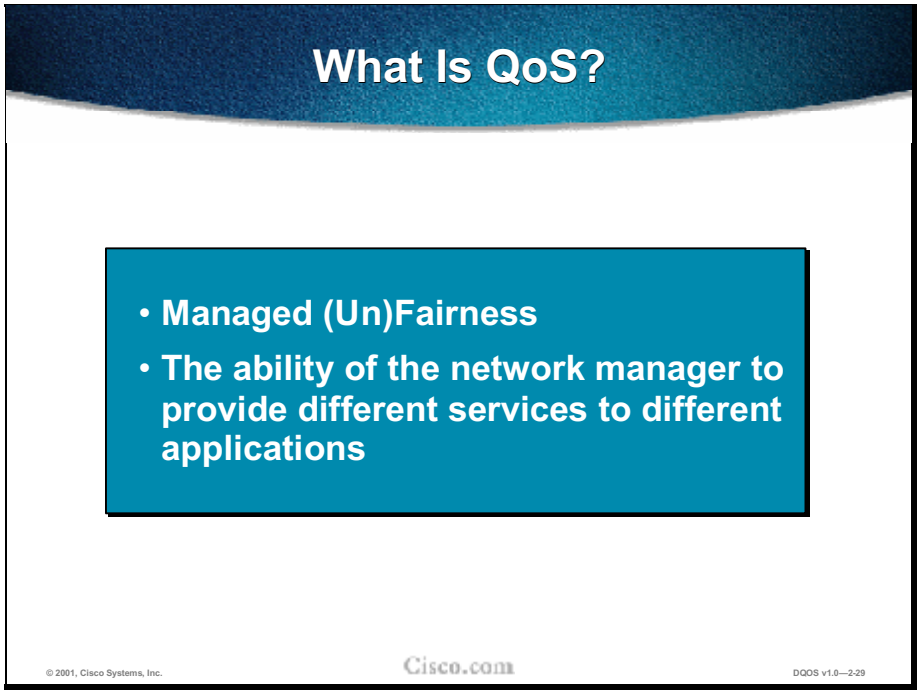
Applications are getting more demanding. Mission-critical applications deployed over IP networks increasingly require quality, reliability, and timeliness assurances. In particular, applications that use voice, video streams, or multimedia must be carefully managed within an IP network to preserve their integrity.

Managing QoS becomes increasingly difficult because many applications deliver unpredictable bursts of traffic. For example, usage patterns for web, e-mail, and file transfer applications are virtually impossible to predict, yet network managers need to be able to support mission-critical applications even during peak periods.

QoS technologies allow IT managers and network managers to:

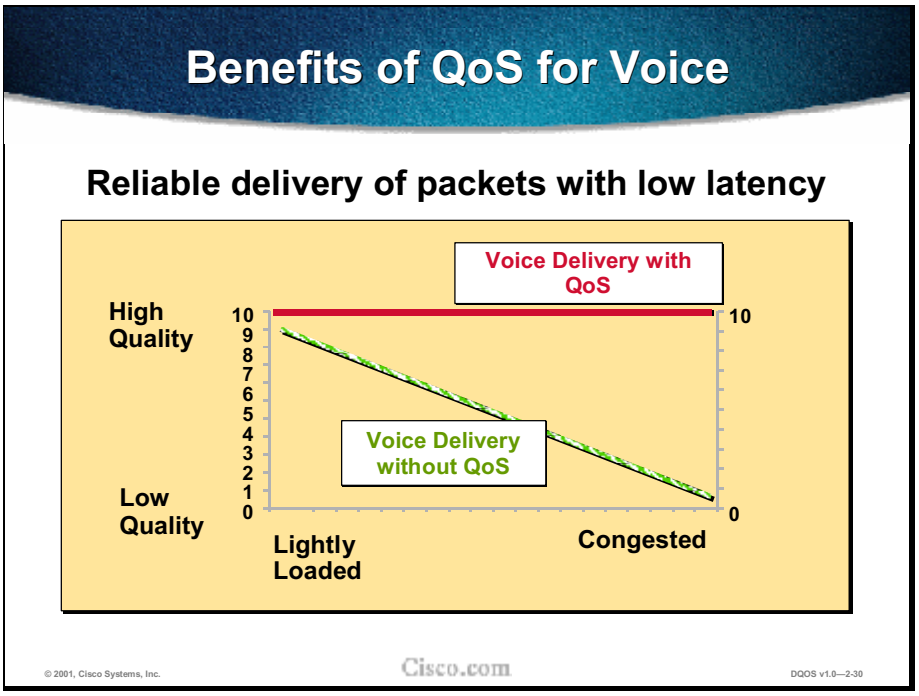
- Predict response times for end-to-end network services
- Manage jitter-sensitive applications, such as audio and video playbacks
- Manage delay-sensitive traffic, such as real-time voice
- Control loss in times of inevitable bursty congestion
- Set traffic priorities across the network
- Support dedicated bandwidth
- Avoid and manage network congestion





This slide raises the question, “Does QoS manage fairness or unfairness?” The answer is that it depends on your point of view. As the following slides show graphically, QoS treats different types of traffic differently, depending on the type and what it needs. Is that fair? Or is it unfair to allow certain types of traffic different treatment?

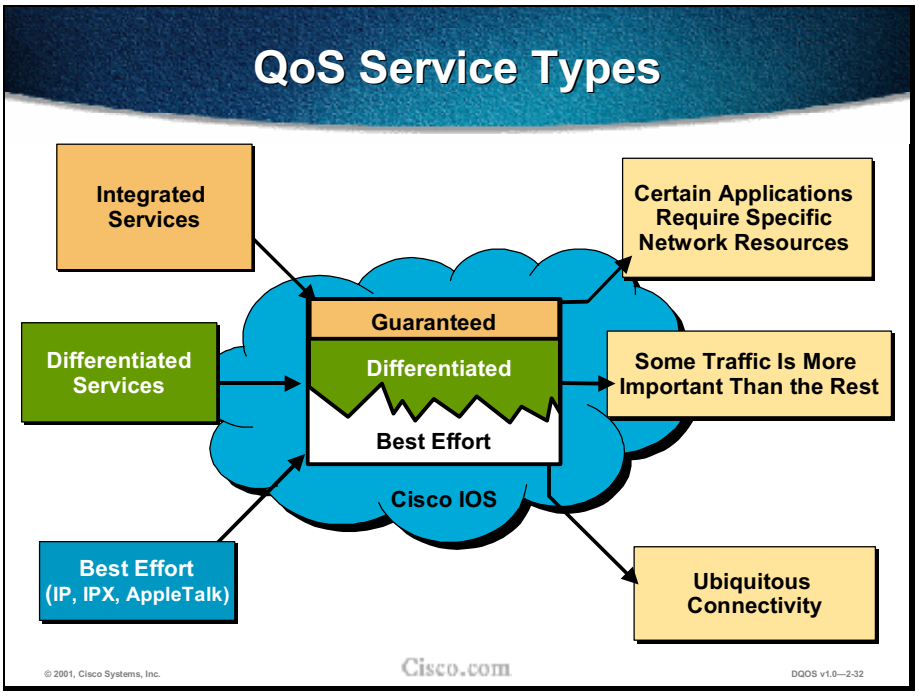
A different definition of QoS focuses on what the network manager is able to do. QoS tools allow voice, video, and data to be managed intelligently as it travels over the network.



QoS Features for Voice

Real-time applications such as voice applications have different characteristics and requirements from those of traditional data applications. Because they are real-time based, 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 unacceptably 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. Cisco IOS QoS features collectively embody these techniques, offering the means to provide priority service that meets the stringent requirements of voice-packet delivery.


QoS Service Types
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—2-31



End-to-end QoS is the ability of the network to deliver service required by specific network traffic from one end of the network to the other. Cisco IOS QoS software supports three types of service models: best effort, integrated, and differentiated services.

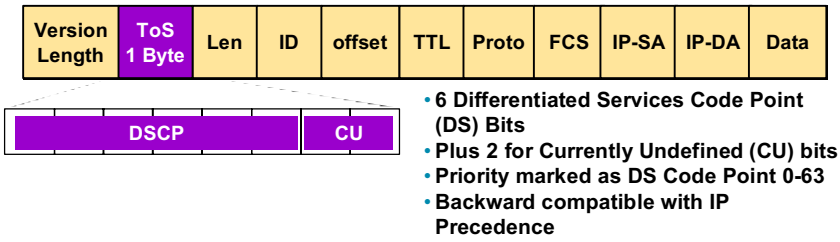
Best effort is a single-service model in which an application sends data whenever it must, in any quantity, and without requesting permission or first informing the network. For best-effort service, the network delivers data if it can, without any assurance of reliability, delay bounds, or throughput. The Cisco IOS QoS feature that implements best-effort service is first-in, first-out (FIFO) queuing. Best-effort service is suitable for a wide range of networked applications such as general file transfers or e-mail.

Integrated service is a multiple-service model that can accommodate multiple QoS requirements. In this model the application requests a specific kind of service from the network before sending data. The request is made by explicit signaling; the application informs the network of its traffic profile and requests a particular kind of service that can encompass its bandwidth and delay requirements. The application is expected to send data only after it gets a confirmation from the network. It is also expected to send data that lies within its described traffic profile.

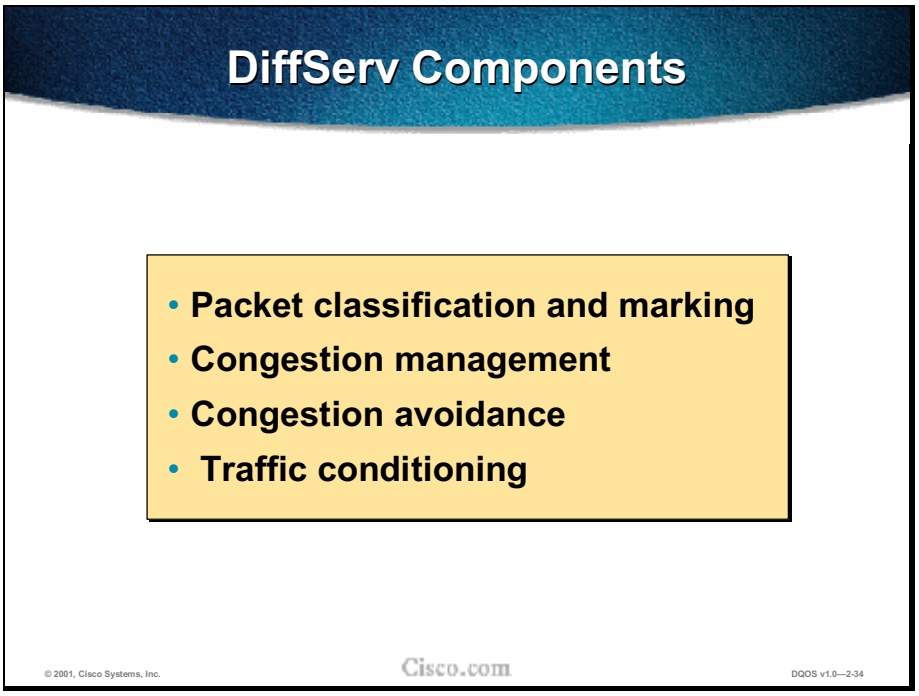
Differentiated service is a multiple-service model that can satisfy differing QoS requirements. For differentiated service, the network tries to deliver a particular kind of service based on the QoS specified by each packet. This specification can occur in different ways, for example, using the IP Precedence bit settings in IP packets or source and destination addresses. The network uses the QoS specification to classify, shape, and police traffic and to perform intelligent queuing.

Differentiated Services (DiffServ)

- Network defined service
- Multiple service model to satisfy differing requirements
- Implemented through 6 bit DSCP Field definitions



- 6 Differentiated Services Code Point (DS) Bits
- Plus 2 for Currently Undefined (CU) bits
- Priority marked as DS Code Point 0-63
- Backward compatible with IP Precedence

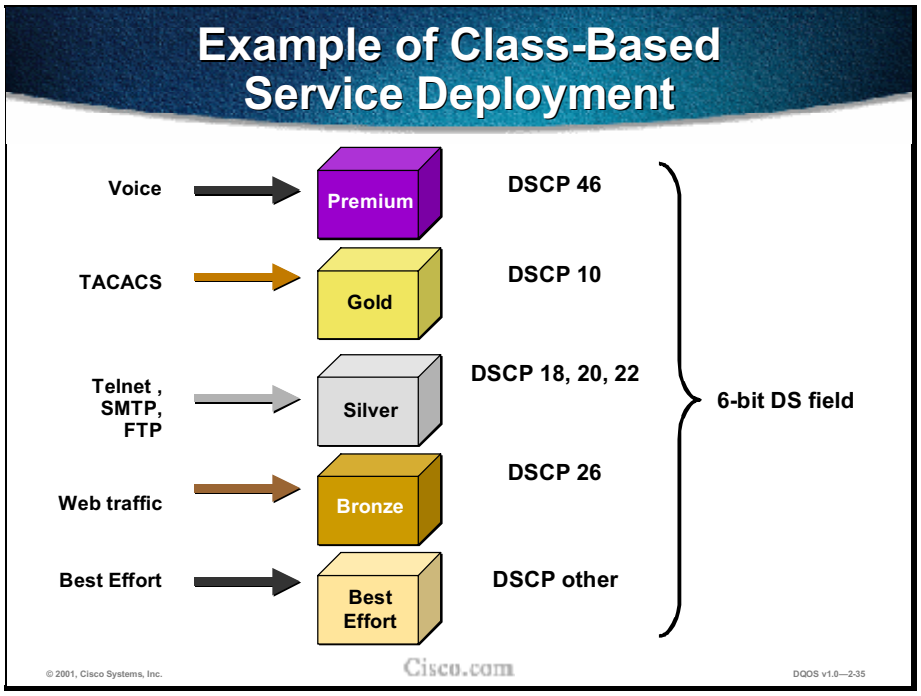


Differentiated Services (DiffServ) is used for several mission-critical applications and for providing end-to-end QoS. Typically, DiffServ is appropriate for aggregate flow because it performs a relatively coarse level of traffic classification.

A flow is defined as an individual, unidirectional data stream between two applications and is uniquely identified by the 5-tuple (source IP address, source port#, destination IP address, destination port#, and the transport protocol).

The DiffServ architecture is based on a simple model where traffic entering a network is classified and possibly conditioned at the boundaries of the network. The class of traffic is then identified with a differentiated services code point (DSCP) or bit marking in the IP header. Within the core of the network, packets are forwarded according to the per-hop behavior associated with the DSCP.

One of the primary principles of the differentiated service model is that one should mark packets as close to the edge of the network as possible. It is often a difficult and time-consuming task to understand to which class of traffic a given data packet belongs, so you want to classify the data as few times as possible. By marking the traffic at the network edge, core network devices and other devices along the forwarding path will be able to quickly determine the proper class of service (CoS) to apply to a given traffic flow.

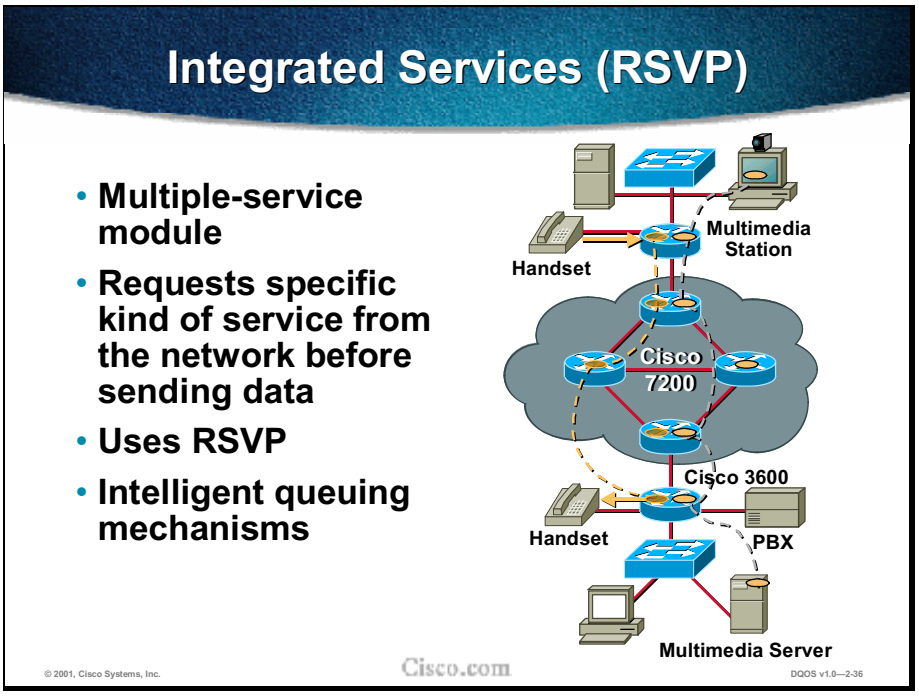


Traffic classes, along with the service level agreements (SLAs), for each traffic class in use on the sample DiffServ implementation are described as follows:

- Voice is considered premium class. The gold class of traffic consists of TACACS sessions, along with traffic marked with DSCP values 12 and 14. The silver traffic class consists of Telnet, Simple Main Transfer Protocol (SMTP), and File Transfer Protocol (FTP) sessions. The bronze traffic class consists of web traffic and traffic marked with DSCP values 28 and 30. Anything else is considered as belonging to the “best-effort” traffic class.

- The premium class should be forwarded with the lowest delay possible, up to a maximum of 500 kBps during periods of congestion. The gold class should be treated preferentially over the silver class, which in turn should be treated preferentially over the bronze class. The gold, silver, and bronze classes should have 35 percent, 25 percent, and 15 percent, respectively, of the interface bandwidth as the minimum bandwidth guarantees. Bronze class should be shaped to 320 kBps, and the best-effort class should be policed to 56 kBps.

- To provide for the various traffic classes, traffic needs to be classified based on DSCP values in a DiffServ domain. So that traffic can be classified based on DSCP values, the traffic should be premarked with the appropriate DSCP values at the time of entering the network.



IntServ and RSVP

Integrated service is a multiple-service model that can accommodate multiple QoS requirements. The Integrated Services (IntServ) model inherits the connection-oriented approach from telephony network design. Every individual communication must explicitly specify to the network its traffic descriptor as well as requested resources. 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 state for each individual communication.

The role of Resource Reservation Protocol (RSVP) in the Cisco QoS architecture is to provide resource admission control for VoIP networks. If resources are available, RSVP accepts a reservation and installs a traffic classifier in the QoS forwarding path. The traffic classifier tells the QoS forwarding path how to classify packets from a particular flow and what forwarding treatment to provide. The installation of a traffic classifier and flow treatment is the interface between RSVP and DiffServ. RSVP is a control plane feature that limits accepted VoIP load to what the network can support. Integration of resource-based admission control with DiffServ network (RSVP aggregation) aims at achieving scalable strict QoS guarantees for VoIP calls.

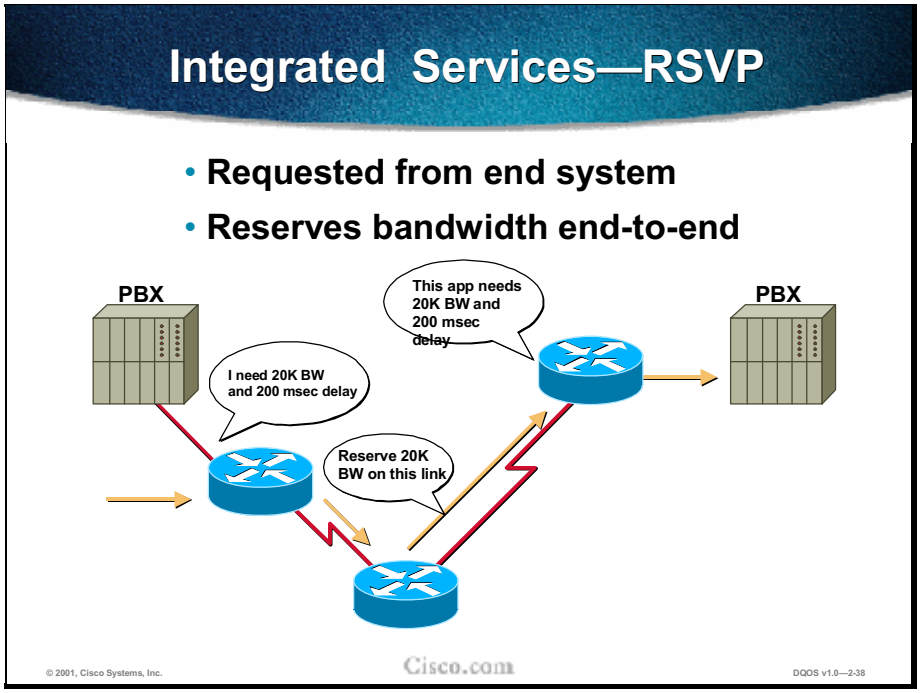
In this model the application requests a specific kind of service from the network before sending data. Explicit signaling makes the request; the application informs the network of its traffic profile and requests a particular kind of service that can encompass its bandwidth and delay requirements. The application is expected to send data only after it gets a confirmation from the network. It is also expected to send data that lies within its described traffic profile.

The network performs admission control based on information from the application and available network resources. It 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 per-flow state, then performing packet classification, policing, and intelligent queuing based on that state.

Cisco IOS QoS includes these features that provide controlled-load service, which is a kind of integrated service:

- Resource Reservation Protocol (RSVP) can be used by applications to signal their QoS requirements to the router.
- Intelligent queuing mechanisms can be used with RSVP to provide the following kinds of services:
- Guaranteed-rate service, which allows applications to reserve bandwidth to meet their requirements. For example, a Voice over IP (VoIP) application can reserve 32 Mbps end-to-end using this kind of service. Cisco IOS QoS uses weighted fair queuing (WFQ) with RSVP to provide this kind of service.
- Controlled-load service, which allows applications to have low delay and high throughput even during times of congestion. For example, adaptive real-time applications such as playback of a recorded conference can use this service. Cisco IOS QoS uses RSVP with weighted random early detection (WRED) to provide this kind of service.

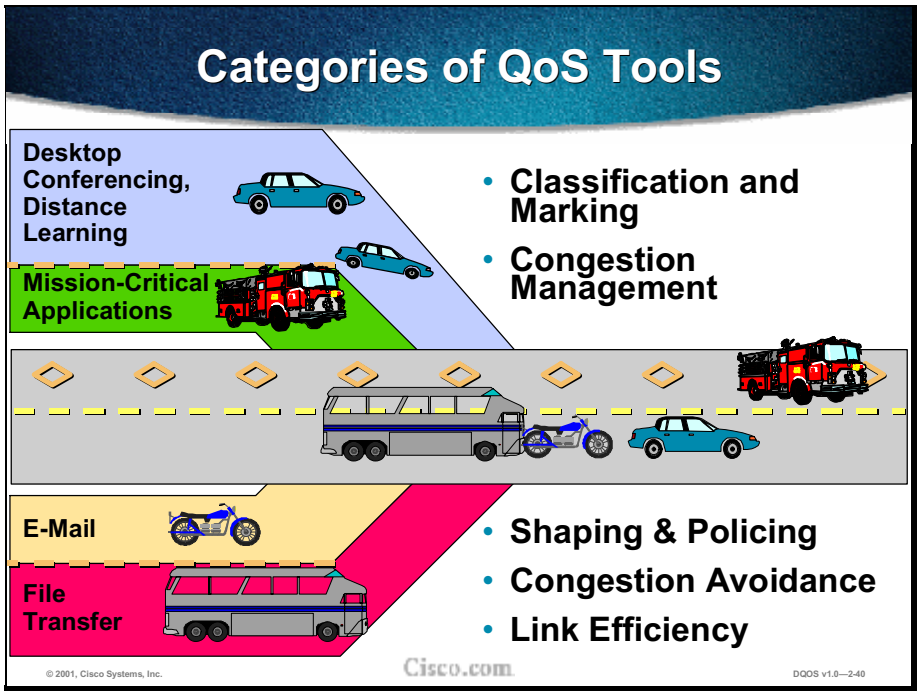


Resource Reservation Protocol (RSVP) is an IP service that allows end systems or hosts on either side of a router network to establish a reserved-bandwidth path between them to predetermine and ensure QoS for their data transmission. RSVP currently has the only standard signaling protocol designed to guarantee network bandwidth from end to end for IP networks.

RSVP is an IETF Internet standard (RFC 2205) protocol for allowing an application to dynamically reserve network bandwidth. RSVP enables applications to request a specific QoS for a data flow, as shown in the slide. Cisco's implementation also allows RSVP to be initiated within the network, using configured proxy RSVP. Network managers can thereby take advantage of the benefits of RSVP in the network, even for non-RSVP-enabled applications and hosts.

Hosts and routers use RSVP to deliver QoS requests to the routers along the paths of the data stream and to maintain router and host state to provide the requested service, usually bandwidth and latency. RSVP uses a mean data rate, the largest amount of data the router will keep in queue, and minimum QoS to determine bandwidth reservation.

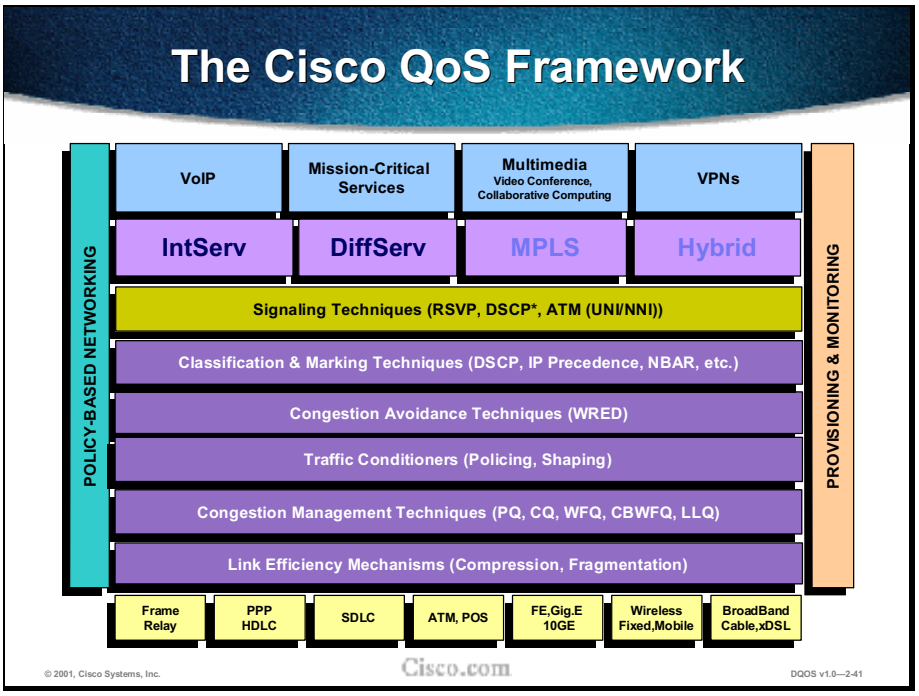
Weighted fair queuing (WFQ) or WRED acts as the workhorse for RSVP, setting up the packet classification and scheduling required for the reserved flows. Using WFQ, RSVP can deliver an Integrated Services guaranteed service. Using WRED, it can deliver a controlled-load service. WFQ continues to provide its advantageous handling of nonreserved traffic by expediting interactive traffic and fairly sharing the remaining bandwidth between high-bandwidth flows, and WRED provides its commensurate advantages for non-RSVP flow traffic. RSVP can be deployed in existing networks with a software upgrade.



The basic categories of tools for QoS are:

- Congestion Management
- Shaping and Policing
- Congestion Avoidance
- Link Efficiency

The following section describes each of these categories in more detail.



This framework shows the QoS tools available to deploy, provision, and monitor end-to-end QoS.

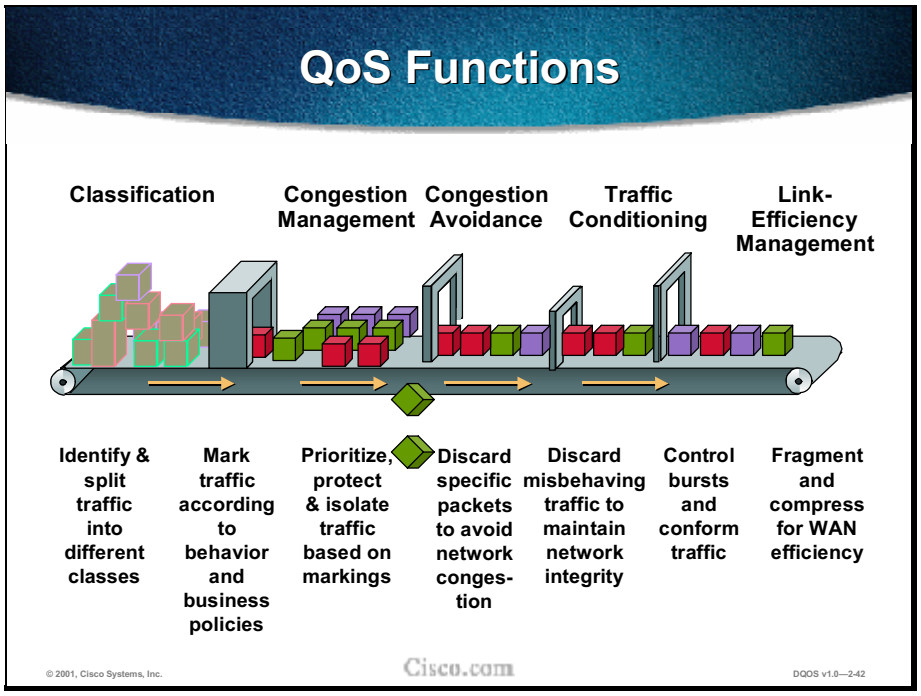
At the bottom are the various transport protocols.

The middle section lists the tools that are used in deploying the QoS: Classification and Marking, Congestion Avoidance, Traffic Conditioners, Congestion Management, and Link Efficiency. Each of these is taught in this class.

The top row shows the different applications that benefit from QoS.

Below the top line are IntServ and DiffServ, which have been defined in the section on QoS services. In the next bar are RSVP and DSCP, the two marking tools of IntServ and DiffServ respectively. The Integrated Services standard defines fine-grained (flow-based) methods of performing IP traffic admission control that uses RSVP. The Differentiated Services standard defines methods of classifying IP traffic into coarse-grained service classes and defines forwarding treatment based on these classifications.

MPLS (Multiprotocol Label Switching) and Hybrid are not part of this course.



This slide shows the five main categories of tools and describes in layman's terms how they contribute to QoS.

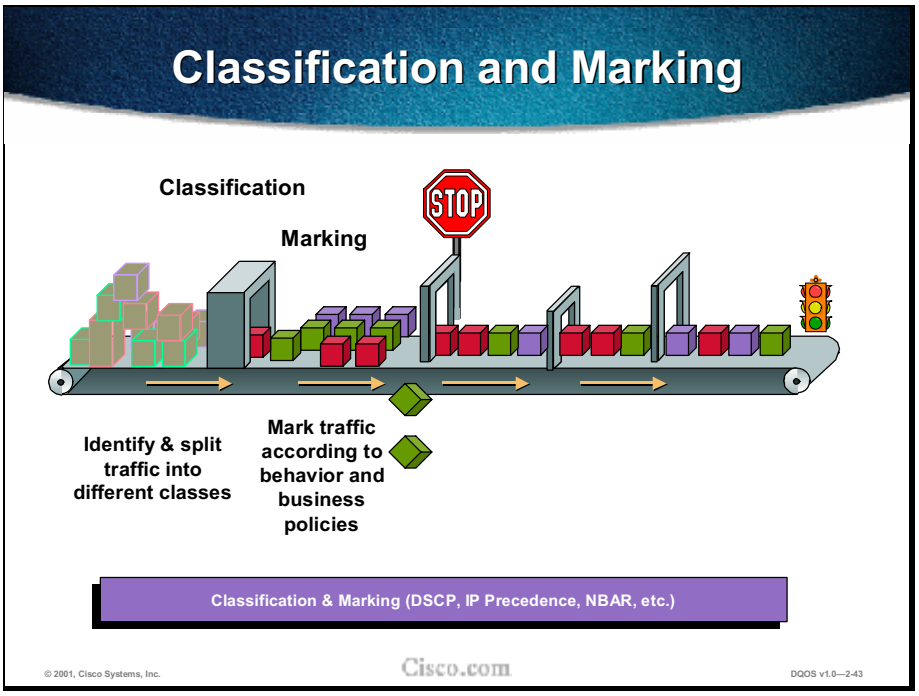
Classification and Marking is the identifying and splitting of traffic into different classes and the marking of traffic according to behavior and business policies. Tools include NBAR, PBR, ACL/Route Map, CAR, and MQC (Modular Quality of Service command-line interface).

Congestion management is the prioritizing, protection, and isolation of traffic based on markings. Tools include IP RTP Priority, CBWFQ, and LLQ.

Congestion avoidance discards specific packets based on markings, to avoid network congestion. Tools include WRED.

Traffic conditioning polices the traffic discard misbehaving traffic to maintain network integrity. It also shapes traffic to control busts. Tools include GTS/FRTS.

Link efficiency management fragments traffic to speed transmission and compresses headers to improve WAN efficiency. Fragmentation tools include LFI and FRF.12. Compression tools include CRTP.



Classification is the identifying and splitting of traffic into different classes.

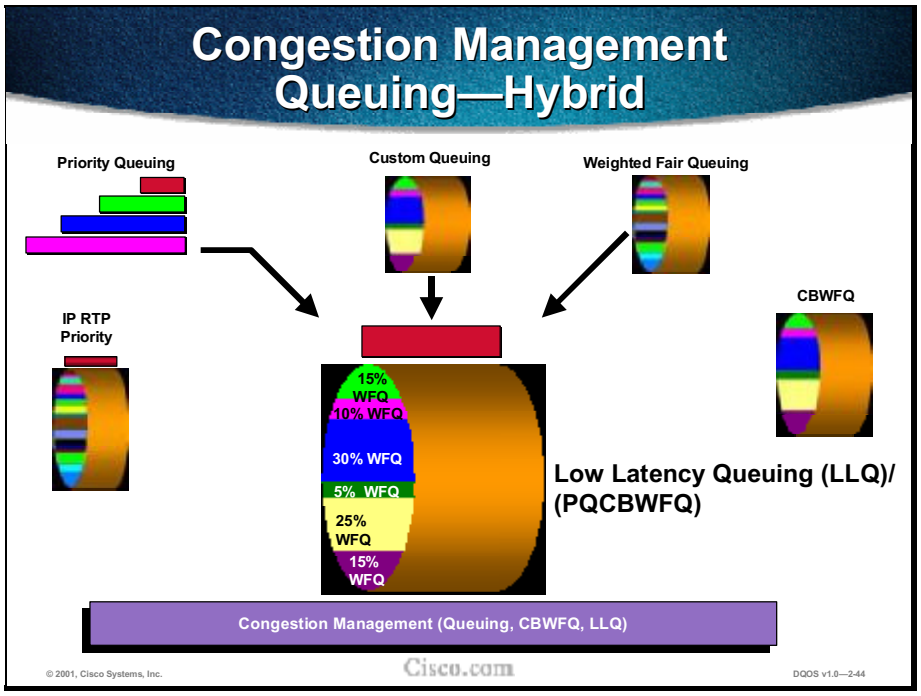
Marking, which is also known as coloring, is done with any of three methods: Modular QoS command-line interface (MQC), policer, and CAR as legacy.

DSCP has been briefly described in the section on Differentiated Services.

Networked-based application recognition (NBAR) is a new classification engine that can recognize a wide variety of applications, including web-based and client/server applications that dynamically assign Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) port numbers. Once the application is recognized, the network can invoke specific services for that particular application. NBAR currently works with QoS features to ensure that the network bandwidth is best used to fulfill client objectives. These features include the ability to guarantee bandwidth to critical applications, limit bandwidth to other applications, drop selective packets to avoid congestion, and mark packets appropriately so that the client network and the service provider's network can provide the proper QoS from end to end.

IP Precedence is one of the Differentiated Services methods for allocating network resources. Three bits in the IP header designated as the type of service (ToS) field can be manipulated to inform network devices that a priority should be given to the IP packet as it traverses the network. This designation is also called coloring a data stream because it causes IP packets with ToS set to stand out from other IP packets.

In Chapter 3, the tools for classification and marking are covered in detail.

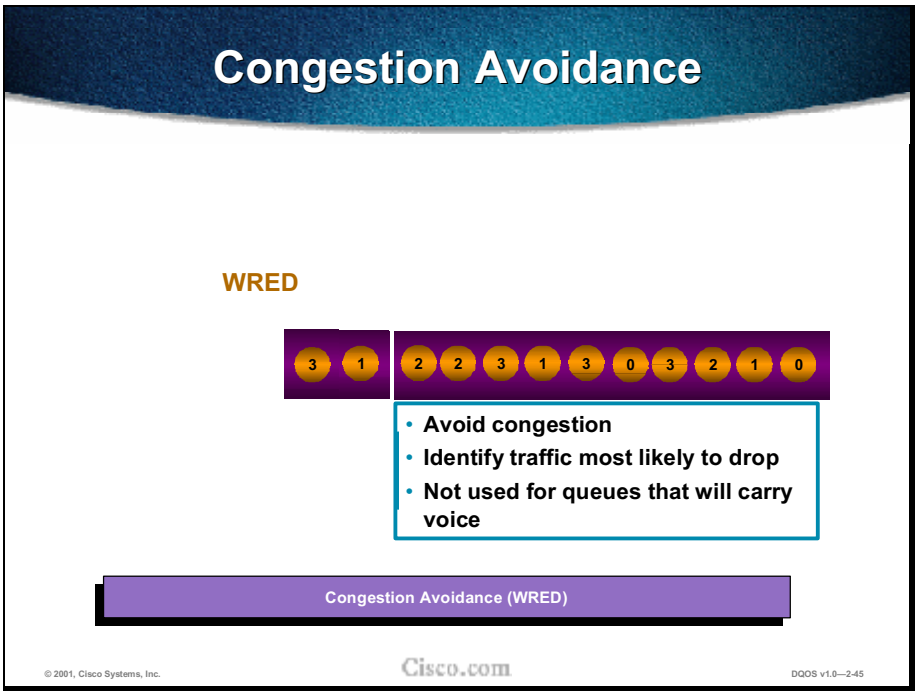


Congestion management features operate to control congestion once it occurs. One way that network elements handle an overflow of arriving traffic is to use a queuing algorithm to sort the traffic, then determine some method of prioritizing it onto an output link. Each queuing algorithm was designed to solve a specific network traffic problem and has a particular effect on network performance.

The Cisco IOS software features for congestion management, or queuing, include:

- FIFO (first-in, first-out)
- PQ (priority queuing)
- CQ (custom queuing)
- WFQ (flow-based weighted fair queuing) [not required]
- CBWFQ (class-based WFQ) [not required]
- IP RTP Priority (also known as PQ/WFQ)
- Frame Relay IP RTP Priority
- LLQ (low latency queuing)

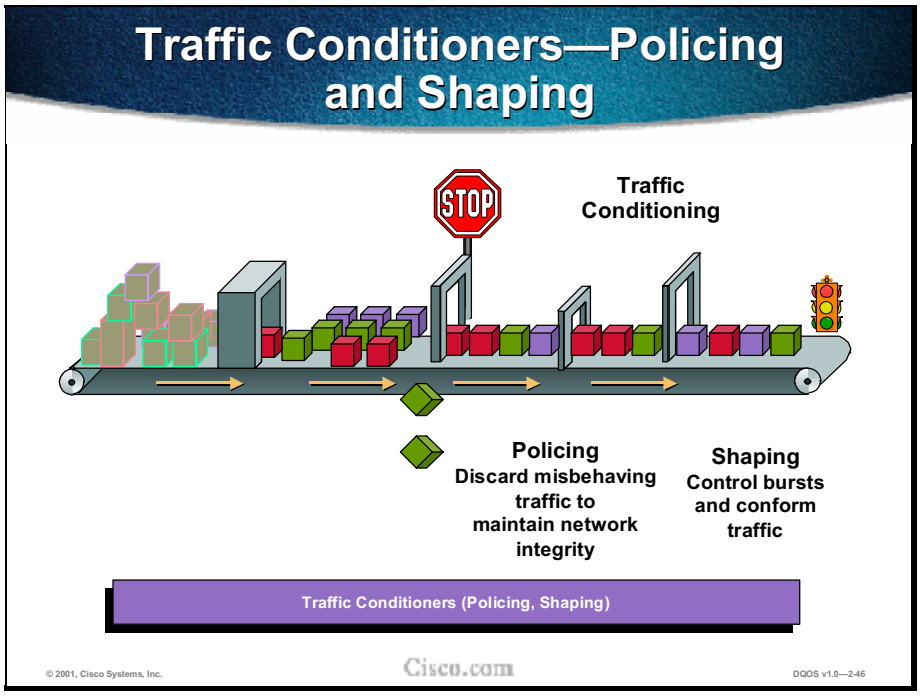
LLQ (low latency queuing) is now the preferred method. It is a hybrid of previous queuing methods developed in the Cisco IOS. As the slide above indicates, LLQ combines priority queuing (PQ), custom queuing (CQ), and weighted fair queuing (WFQ).



Congestion-avoidance techniques monitor network traffic loads in an effort to anticipate and avoid congestion at common network bottlenecks. Congestion avoidance is achieved through packet dropping.

Weighted random early detection (WRED) is Cisco's basic congestion-avoidance technique. WRED increases the probability that congestion is avoided by dropping low-priority packets rather than high-priority packets.

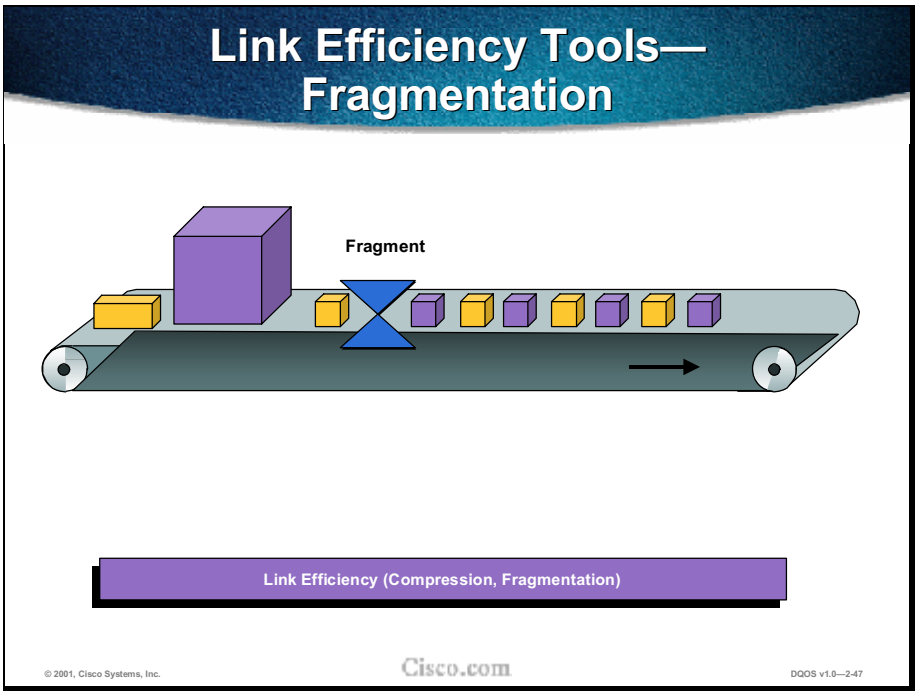
Note that WRED is not recommended for voice queues. The network should not be designed to drop voice packets.



Traffic entering a network can be conditioned by using a policer or shaper.

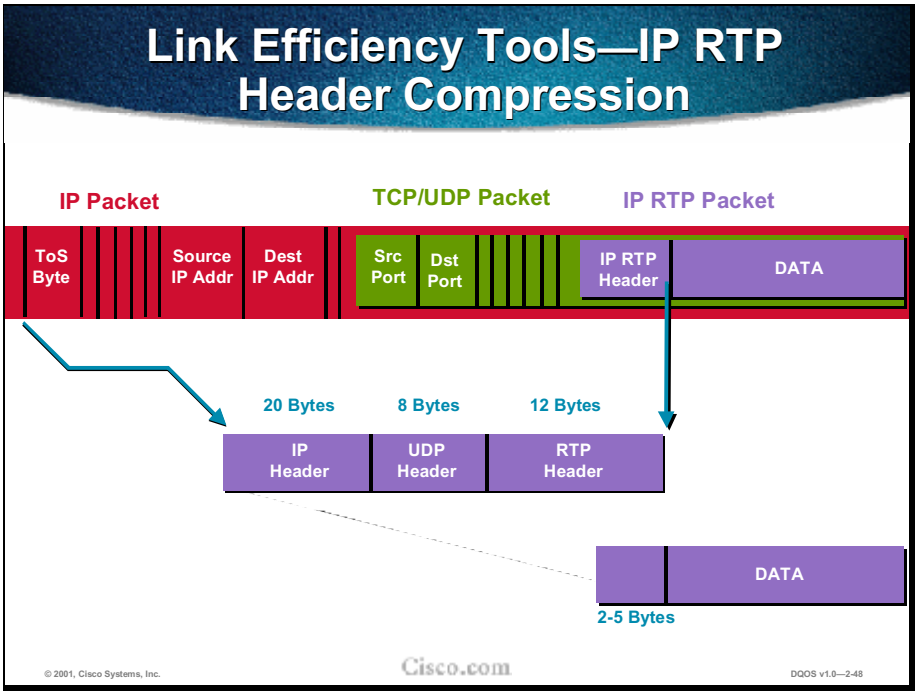
Policing is the ability to control bursts and conform traffic to ensure that certain types of traffic get certain types of bandwidth.

Shaping avoids delays by smoothing out speed mismatches in the network and limiting transmission rates. Cisco's QoS software solutions include two traffic shaping tools—generic traffic shaping (GTS) and Frame Relay traffic shaping (FRTS)—to manage traffic and congestion on the network.



Cisco IOS QoS software offers two link-efficiency mechanisms that work in conjunction with queuing and traffic shaping to manage existing bandwidth more efficiently and predictably: Link fragmentation and interleaving (LFI) and Compressed Real-Time Protocol (CRTP).

Interactive traffic, such as Telnet and 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. Cisco IOS QoS LFI reduces delay and jitter on slower-speed links by breaking up large datagrams and interleaving low-delay traffic packets with the resulting smaller packets.

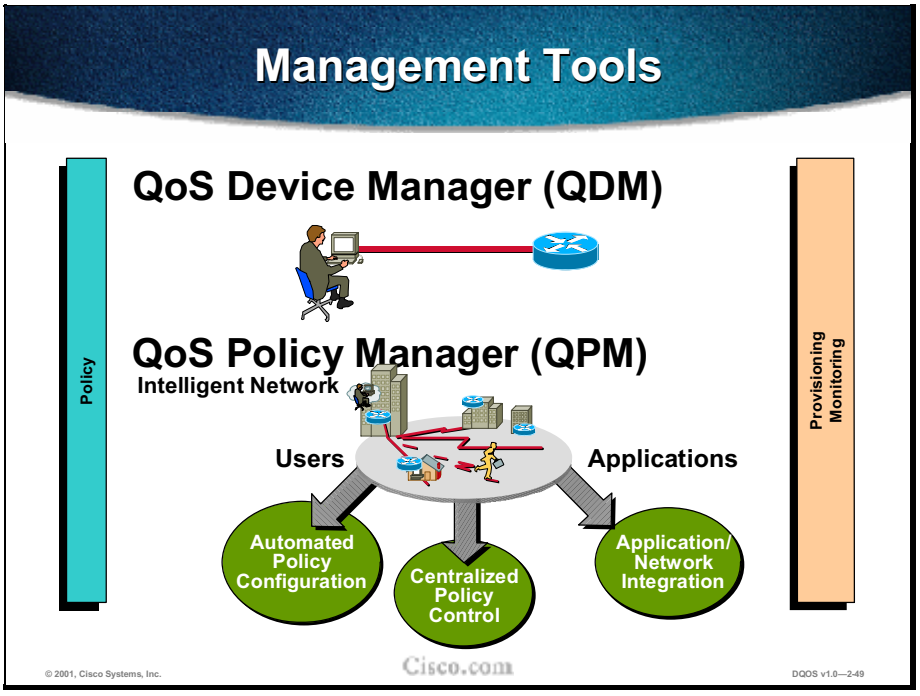


Compressed Real-Time Transport Protocol

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

To avoid unnecessary consumption of available bandwidth, the RTP header compression feature—referred to as CRTP—is used on a link-by-link basis. RTP header compression provides 200 percent compression, allowing to squeeze down and gain significant best possible utilization that we can.

In the slide above, the IP header is 40 bytes. Compression squeezes the header to 2 to 5 bytes.

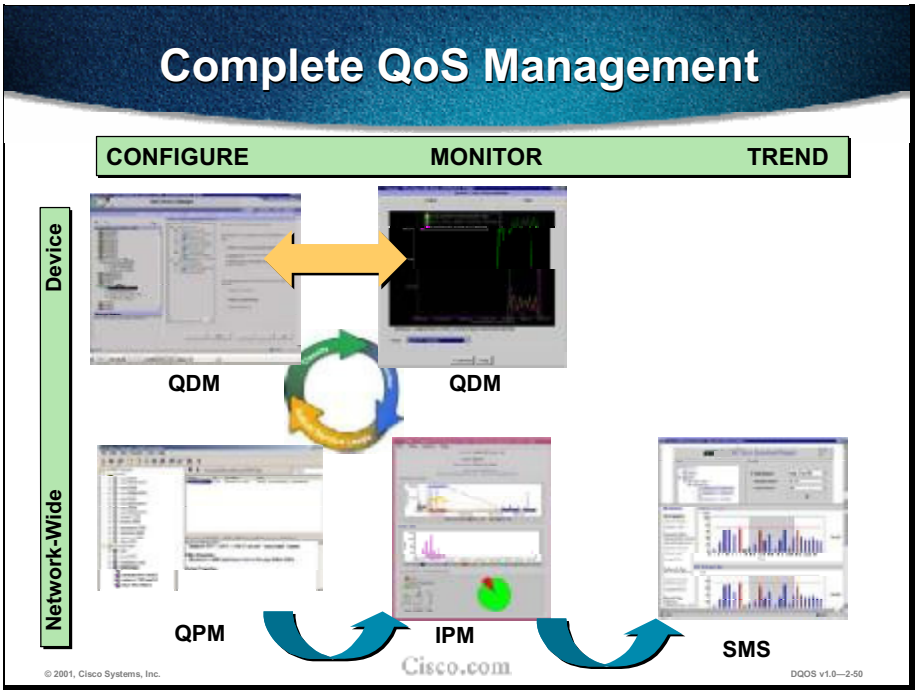


There are several tools to help manage the QoS features for IOS-based devices on the network.

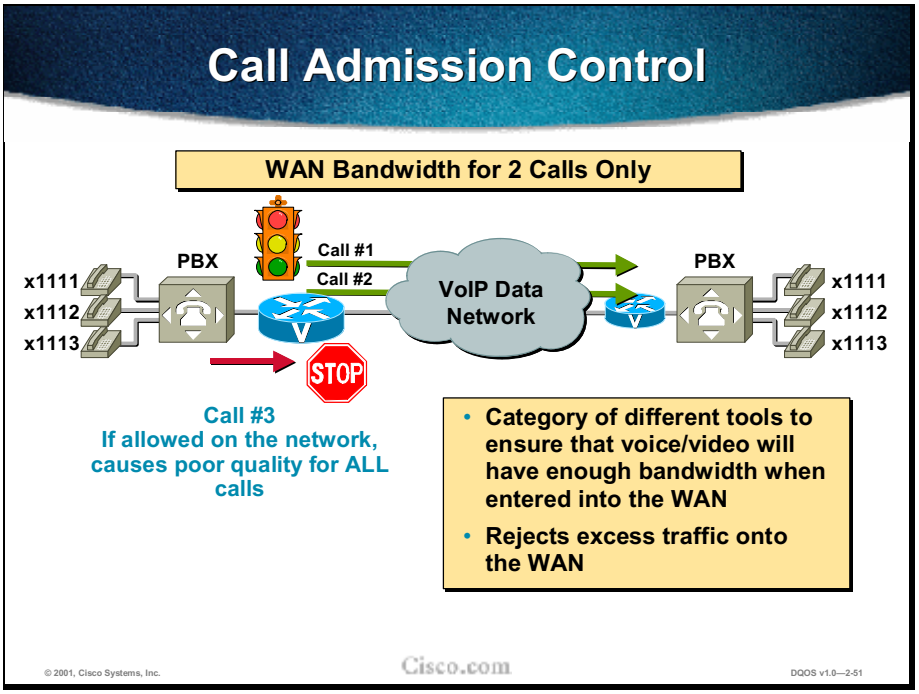
QoS Device Manager (QDM) is a tool for managing individual devices on the network. QDM has the ability to both configure and monitor QoS functionality on a Cisco router from a simple, web-based interface. Furthermore, QDM delivers this functionality within a web-based architecture that requires no client or server setup. Without QDM, it is not possible to easily monitor QoS device-level statistics, such as traffic rate per QoS class or packet drop rate per QoS class. Furthermore, without QDM, it is much more complicated to configure QoS classes and enforcement mechanisms on the router because it requires detailed knowledge of the Cisco IOS commands related to QoS.

QoS Policy Manager (QPM) is a software program that allows the network administrator to manage all the devices on the network. The network administrator can configure devices

according to established policies and propagate these configurations throughout the network, providing centralized policy control and application/network integration.



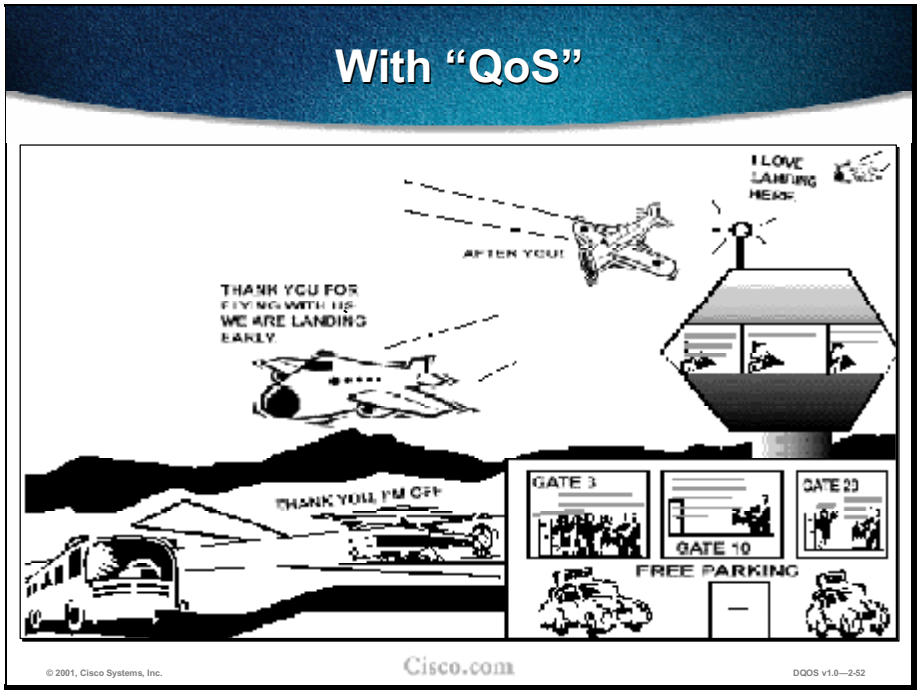
This figure shows the relationship of various management tools. In addition to QDM and QPM, Cisco offers IPM (Internet Performance Manager) and SMS (Service Management Solution) for additional monitoring and trending analysis.

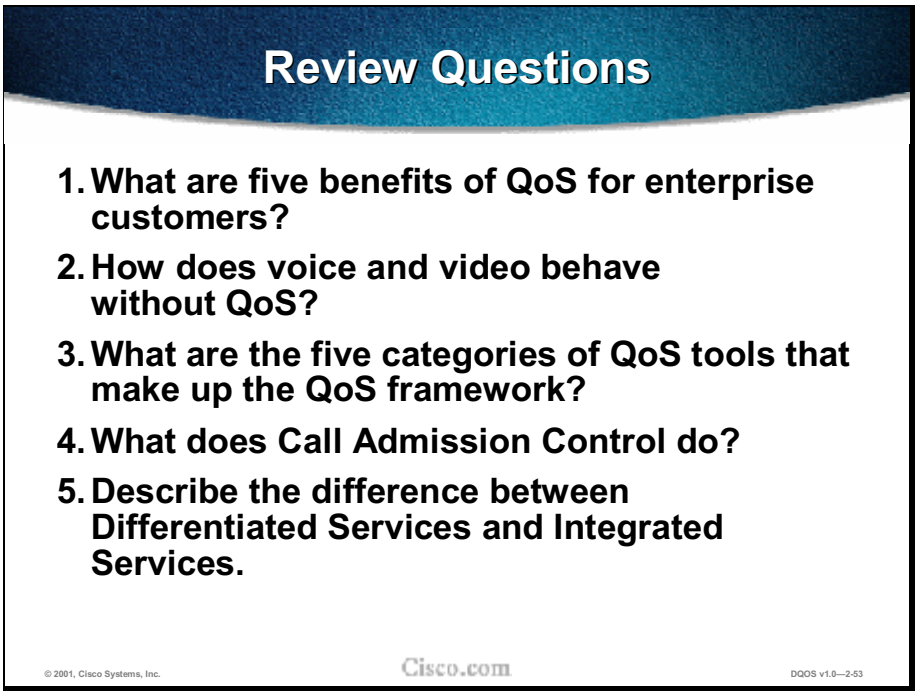


Call admission control (CAC) is a category of tools for ensuring that there is sufficient bandwidth for voice and video before traffic enters the network. This effectively prevents too much voice or video (which typically has fixed bandwidth requirements) from being sent down a wide-area link that does not have enough bandwidth to handle it. In the case of voice, for example, if there is enough bandwidth to carry five voice calls between locations A and B, and a sixth call is placed, this call should not be admitted across the link because the quality of all calls could suffer, regardless of the QoS techniques used. If this call is not admitted, it is automatically rerouted via some other path, such as the PSTN links that are available between sites.

The tools that will be taught include local configuration options, measurement-based options, and resource-based options. The local configuration options are physical DS0 limitations, max connections, voice bandwidth for FR, trunk conditioning, and local voice busyout (LVBO). The measurement-based options are advanced voice busyout (AVBO), resource availability indication (RAI), gatekeeper (GK) zone bandwidth, public switched telephone network (PSTN) fallback. Resource-based options are Resource Reservation Protocol (RSVP).

Location-based admission control of IP telephony can be implemented using current versions of the CallManager application.





1. What are five benefits of QoS for enterprise customers?

Answer: Reliability; control of bandwidth, jitter, and delay; ability to classify services; ability to tailor and shape network traffic

2. How does voice and video behave without QoS?

Answer: Jitter, delay, bandwidth congestion, and dropped packets

3. What are the five categories of QoS tools that make up the QoS framework?

Answer: Classification, congestion management, congestion avoidance, policing and shaping, link efficiency. Set of tools to manage different types of data traffic according to the needs of the traffic and the business.

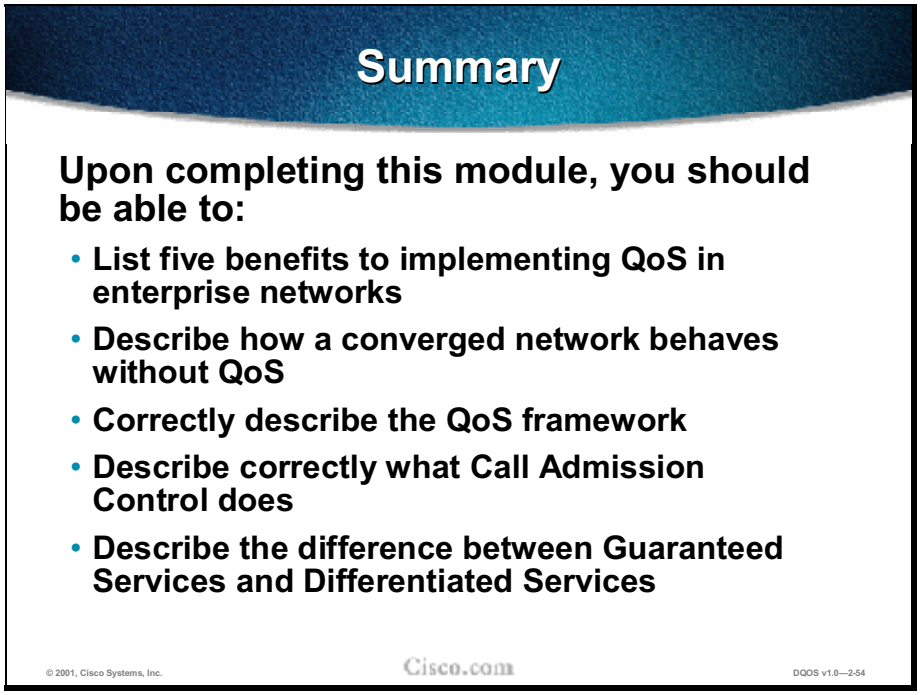
4. What does call admission control do?

Answer: For connection traffic a method of controlling egress onto the WAN if bandwidth is not available.

5. Describe the difference between Differentiated Services and Guaranteed Services.

Answer: Differentiated Services is a set of tools to manage the flow of traffic and thus to differentiate how different traffic is to be forwarded. Guaranteed Services is a means to reserve bandwidth to ensure data transmission.

Summary



3

Classification and Marking

Overview

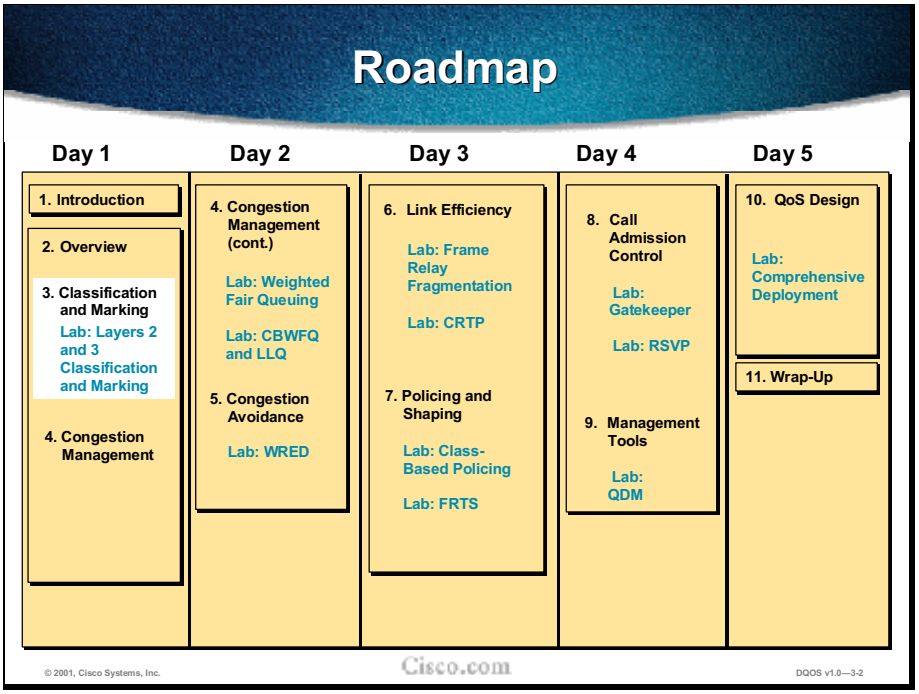
This chapter explains the tools and techniques for classification and marking. Topics covered include class of service (CoS), IP Precedence, DiffServ classification, Modular QoS CLI, class-based marking, Layer 2 – Layer 3 mapping, network-based application recognition (NBAR), policy-based routing (PBR), and committed access rate (CAR).

Objectives

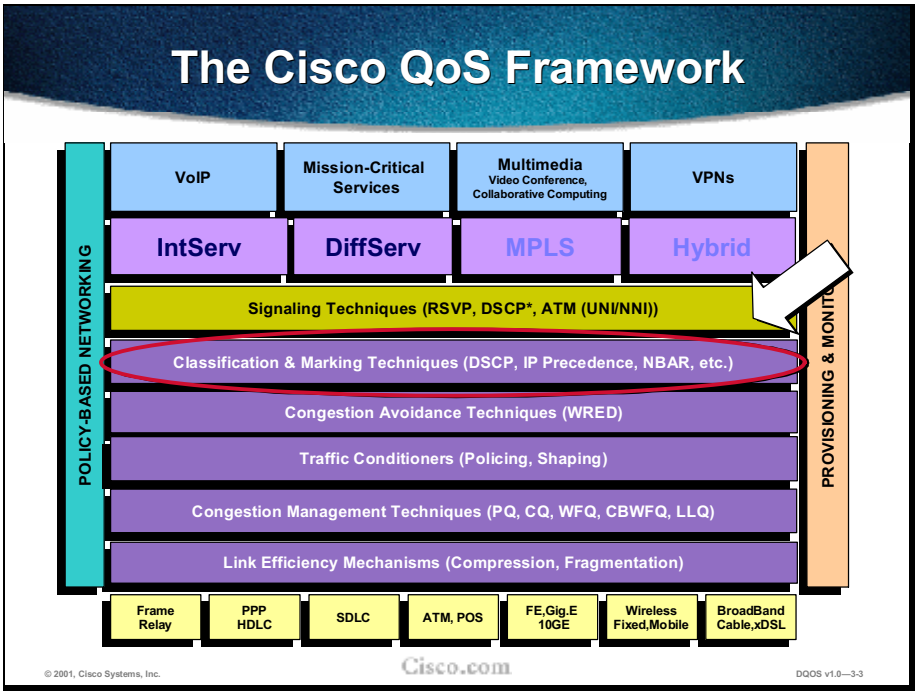
Upon completing this chapter, you will be able to:

- Explain the reason for classification and marking
- Explain the difference between classification and marking
- Explain class of service (CoS), IP Precedence, and DiffServ code points
- Configure QoS policy using Modular QoS CLI
- Explain the role of network-based application recognition (NBAR)
- Classify and mark traffic

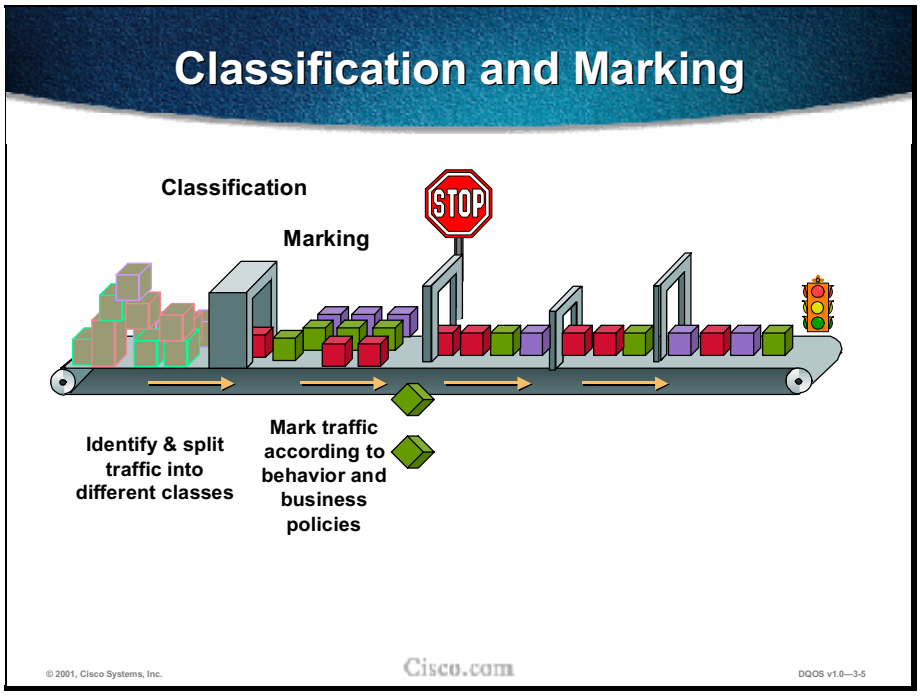
Outline



The figure above shows the plan for the week. With the exception of the Overview, each chapter has a lab for using the materials that have just been taught.



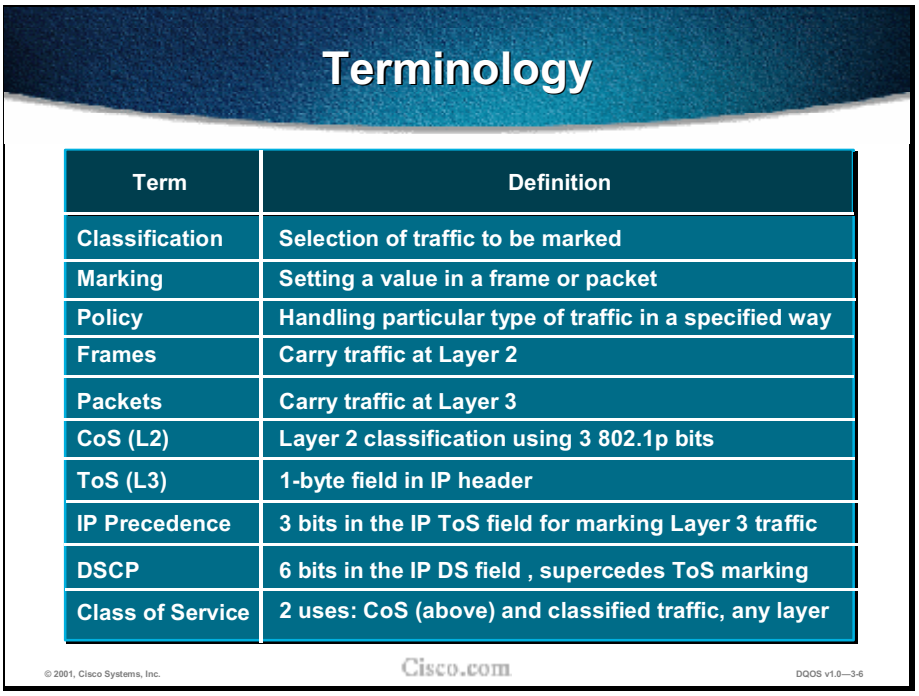
The diagram shows where Classification and Marking fit into the big QoS picture.



For QoS methods to be used in the network, traffic must be classified with different priorities. Each classification must be marked for identification so QoS methods can be applied. This chapter covers various aspects of classification and marking.

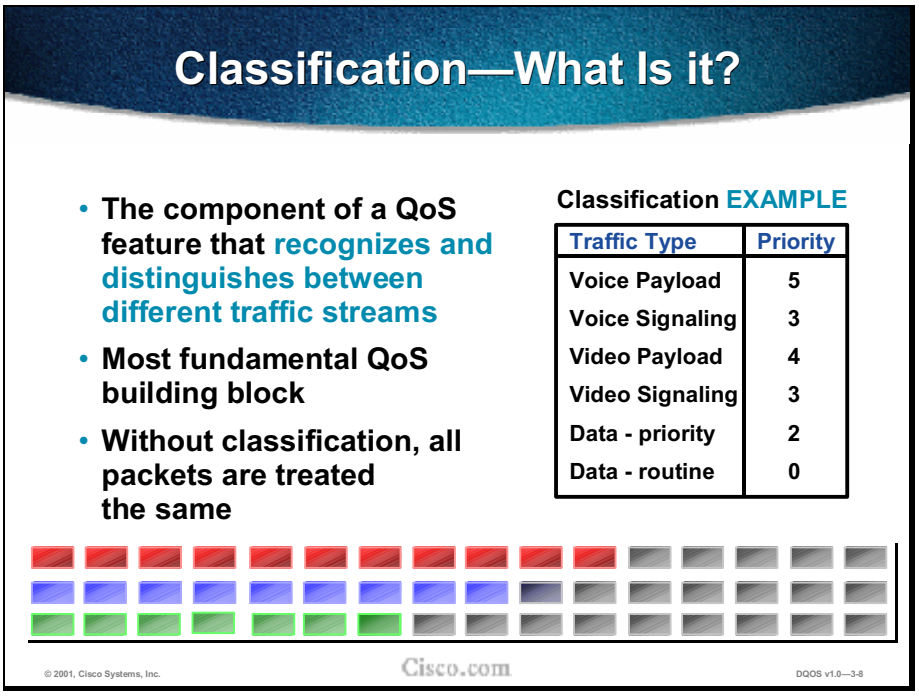
Topics include:

- Traffic classification
- Marking methods
- Introduction to Modular QoS command-line interface (MQC), an IOS tool used to configure, display, and monitor many QoS features including classification and marking features
- NBAR—network-based application recognition, an IOS protocol-discovery and classification capability
- PBR—policy-based routing
- Classification using access control lists (ACLs) and route maps
- Classification using dial peers
- Classification capabilities of committed access rate (CAR)



It is useful before proceeding to provide working definitions for terms commonly encountered in classification and marking.


Classification
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—3-7



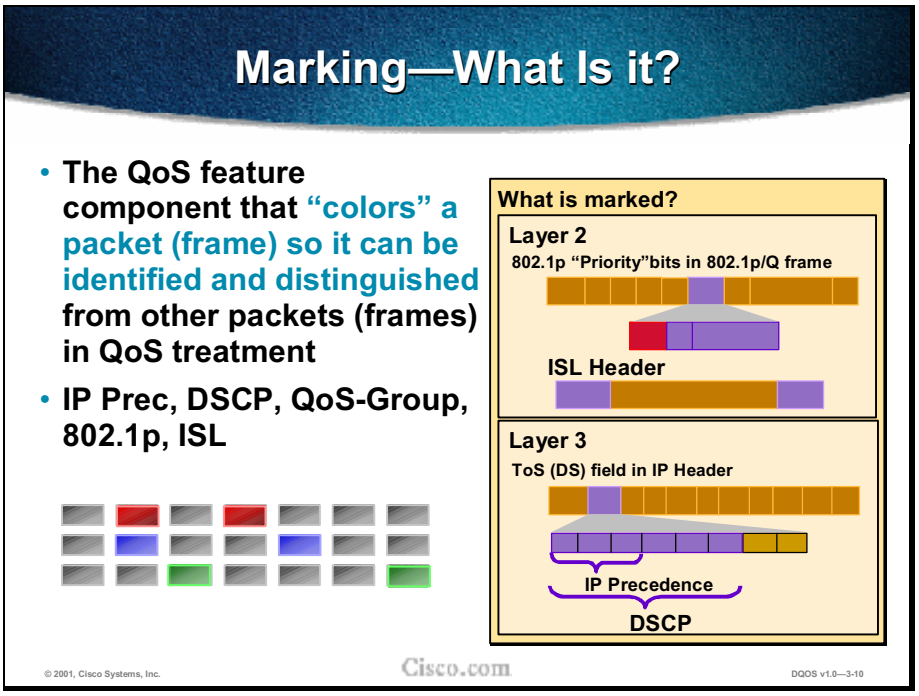
Packet classification uses a traffic descriptor (for example, the IP Precedence or DSCP, each described elsewhere in this chapter) to categorize a packet within a specific group in order to define that packet. After the packet has been defined (that is, classified), the packet is then accessible for QoS handling on the network.

Using packet classification, you can partition network traffic into multiple priority levels or classes of service. When traffic descriptors are used to classify traffic, the source agrees to adhere to the contracted terms and the network promises a quality of service. Traffic policers, traffic shapers, and queuing techniques use the traffic descriptor of the packet (that is, the classification of the packet) to ensure adherence to that agreement.

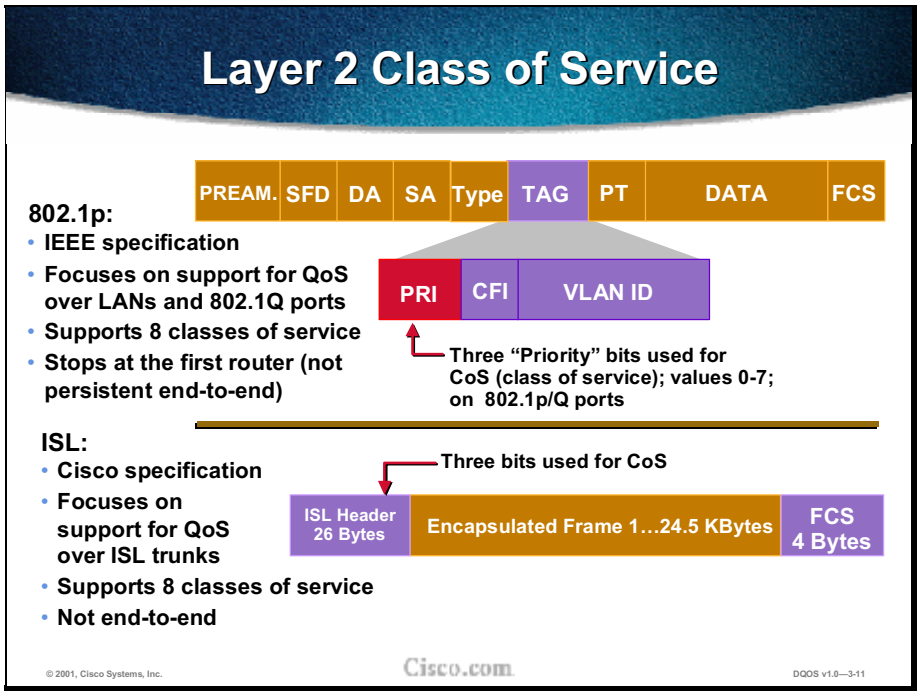
For an overview of classification see *Classification Overview* at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcprt1/qcdclass.htm


Packet Marking
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—3-9



Marking is related to classification. Marking lets you classify a packet or frame based on a specific traffic descriptor. Marking a packet or frame with its classification lets you set information in the Layer 2, 3, or 4 headers, or even set information within the payload of a packet, so that the packet or frame can be identified and distinguished from other packets (or frames). This section describes marking Layer 2 and Layer 3 headers.



The IEEE 802.1p standard is a supplement to IEEE 802.1D, "Standard for Local Area Network MAC (Media Access Control) Bridges," which is a standard for connecting LANs through MAC bridges. The 802.1p standard supports eight classes of service (CoS). A CoS value of 0 means routine service (no priority).

The 802.1Q standard is an IEEE specification for implementing virtual LANs (VLANs) in Layer 2 LAN switches.

An Ethernet frame might or might not be 802.1p/Q compliant. If it is, then a 4-byte Tag field is inserted in the Ethernet frame. The Tag field serves the purposes of both standards (802.1p and 802.1Q). The first three bits, known as the user priority bits, are 802.1p bits. The rest of the Tag field is made up of the CFI (Canonical Format Identifier) and VLAN ID fields, which are 802.1Q fields. The Tag field is sometimes called the 802.1p/802.1Q field, or the 802.1p/Q field.

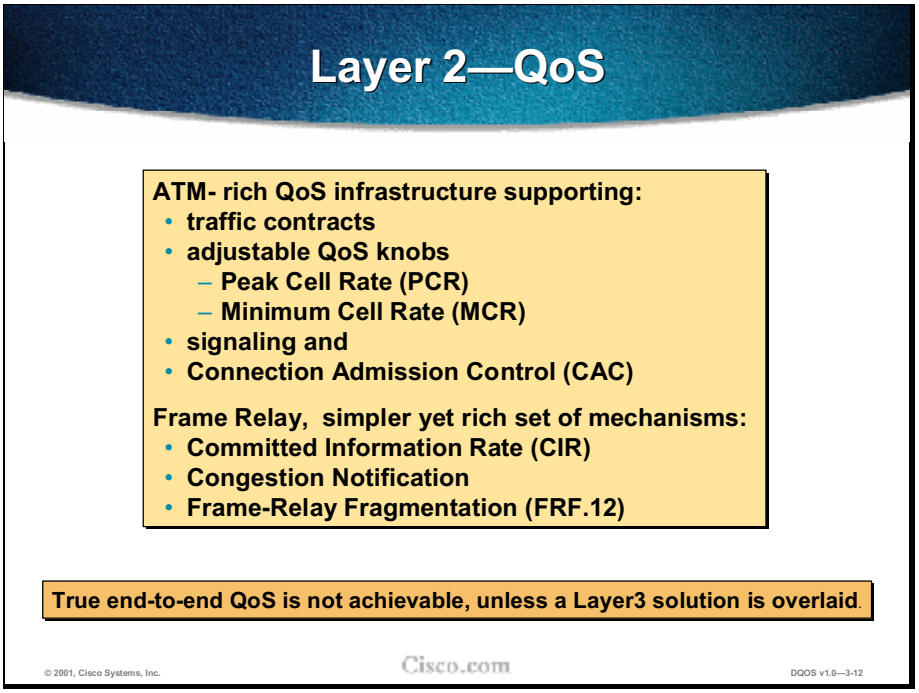
So when a Layer 2 Ethernet frame is marked for priority, the three 802.1p user priority bits are set to a value 0-7.

For additional information on 802.1p/Q marking see the following URLs:

- http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t3/dtbridge.htm#xtocid114535
- http://www.cisco.com/cpress/cc/td/cpress/design/topdown/td0512.htm

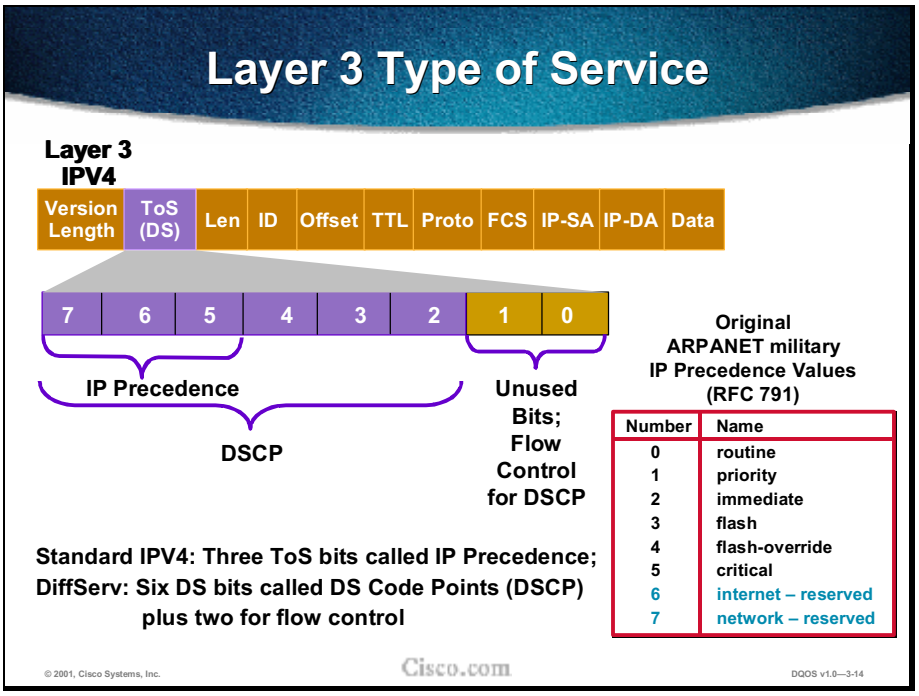
Inter-Switch Link (ISL) is a proprietary Cisco protocol for interconnecting multiple switches and maintaining VLAN information as traffic goes between switches. ISL is 802.1p compliant. The ISL frame header has a 1-byte User field that carries an IEEE 802.1p CoS value in the three least significant bits. When an ISL frame is marked for priority, the three 802.1p CoS bits are set to a value 0-7.

- http://www.cisco.com/univercd/cc/td/doc/product/software/ios113ed/113ed_cr/switch_c/xcisl.htm
- http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_5_5/cnfg_gd/qos.htm



Before the IETF defined IP (Layer 3) QoS methods, the ITU-T (International Union for Telecommunications, Telecommunications), the Asynchronous Transfer Mode (ATM) Forum, and the Frame Relay Forum (FRF) had already arrived at standards to do Layer 2 QoS in ATM and Frame Relay networks. The ATM standards define a very rich QoS infrastructure by supporting traffic contracts, many adjustable QoS knobs (such as peak cell rate [PCR], minimum cell rate [MCR], and so on), signaling, and Connection Admission Control (CAC). Frame Relay, on the other hand provides for a simpler yet rich set of mechanisms to provide for a committed information rate (CIR), congestion notification, and the recently introduced Frame Relay fragmentation (FRF.12).

Though these rich QoS mechanisms exist in Layer 2 transport technologies, true end-to-end QoS is not achievable, unless a Layer 3 solution is overlaid. Service providers offering both ATM/Frame Relay and IP services want to provide robust QoS solutions to customers. Mapping Layer 3 QoS to Layer 2 QoS is the first step toward achieving a complete solution that does not depend on any specific Layer 2 technology. Both IntServ and DiffServ can be implemented over QoS-aware transports such as ATM and Frame Relay. For example, the IntServ controlled-load service can be implemented using RSVP, over an ATM VBR-rt (variable bit rate, real-time) switched virtual circuit (SVC). With DiffServ, packets marked with different ToS-byte values can be sent over different ATM PVCs or SVCs. As an example, high-priority traffic may go over a VBR-nrt VC, and all other traffic may go over an available bit rate (ABR) VC, with the VBR VC capable of preempting the ABR VC in case of congestion or failure. Similarly, Frame Relay traffic shaping (FRTS) (slowing down the rate of transmission by buffering, in response to congestion notification by the FR switches), FRF.12 (packet fragmentation and interleaving on low-speed FR links), and other mechanisms can be used to complement IP QoS.Thus, a true end-to-end QoS solution comprises both Layer 3 and Layer 2 QoS and is media independent. Introduction of a Gigabit Ethernet link somewhere along the packet's path poses no problem to deliver QoS, as the Layer 3 QoS is still preserved and can even be enhanced by mapping to the 802.1p (User Priority) QoS mechanism on Ethernet (RFC-1349). Cisco IOS QoS focuses on delivering exactly this model—interoperability/mappings between Layer 2 and Layer 3 QoS over IP, ATM, Frame Relay, packet over SONET (POS), Ethernet, etc.



IP Precedence uses the three precedence bits in the IPv4 header's ToS (type of service) field to specify class of service for each packet, as shown above. You can partition traffic in up to six classes of service using IP Precedence (settings 6 and 7 are reserved for internal network use). The queuing technologies throughout the network can then use this signal to provide the appropriate expedited handling.

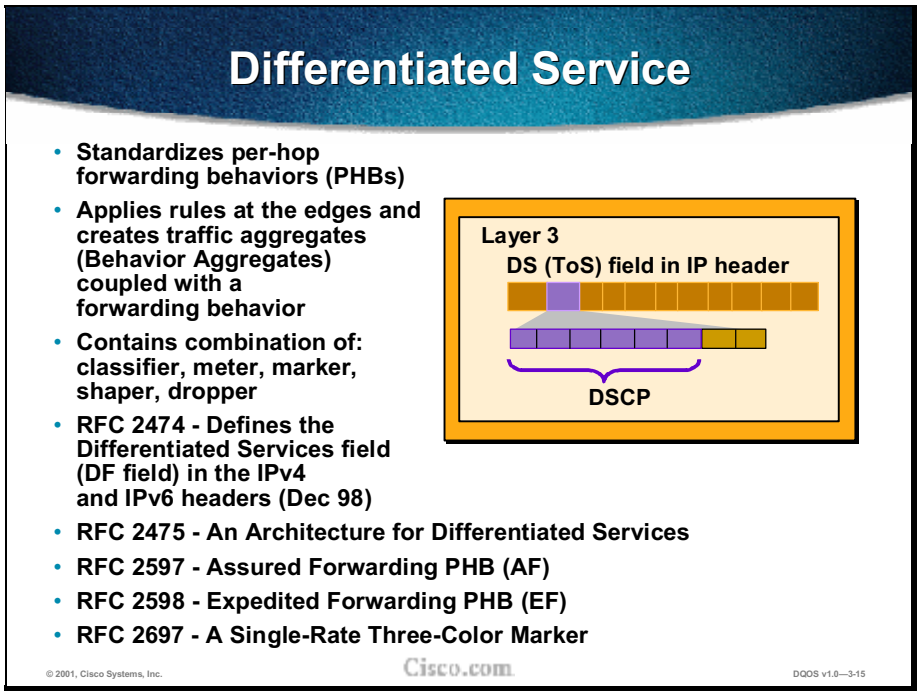
Features such as policy-based routing, class-based marking, and committed access rate (CAR) can be used to set precedence based on extended access-list classification. This allows considerable flexibility for precedence assignment, including assignment by application or user, or by destination and source subnet, and so on. Typically this functionality is deployed as close to the edge of the network (or administrative domain) as possible so each subsequent network element can provide service based on the determined policy.

IP Precedence can also be set in the host or network client, and this signaling can be used optionally; however, this can be overridden by policy within the network.

IP Precedence enables service classes to be established using existing network queuing mechanisms (for example, WFQ or WRED) with no changes to existing applications or complicated network requirements. Note that this same approach is easily extended to IPv6 using its priority field.

DiffServ is a new model that supersedes, and is backward compatible with, IP Precedence. DiffServ uses six prioritization bits, which permits classification of up to 64 values (0 to 63). A DiffServ value is called a differentiated services code point (DSCP).

With DiffServ, the DS field supersedes the IPv4 ToS octet and the IPv6 Traffic Class octet. Six bits of the DS field are used as the DS code points to select the per-hop behavior (PHB) at each interface. A currently unused (CU) two-bit field is reserved for explicit congestion notification (ECN). The value of the CU bits is ignored by DS-compliant interfaces when determining the PHB to apply to a received packet.



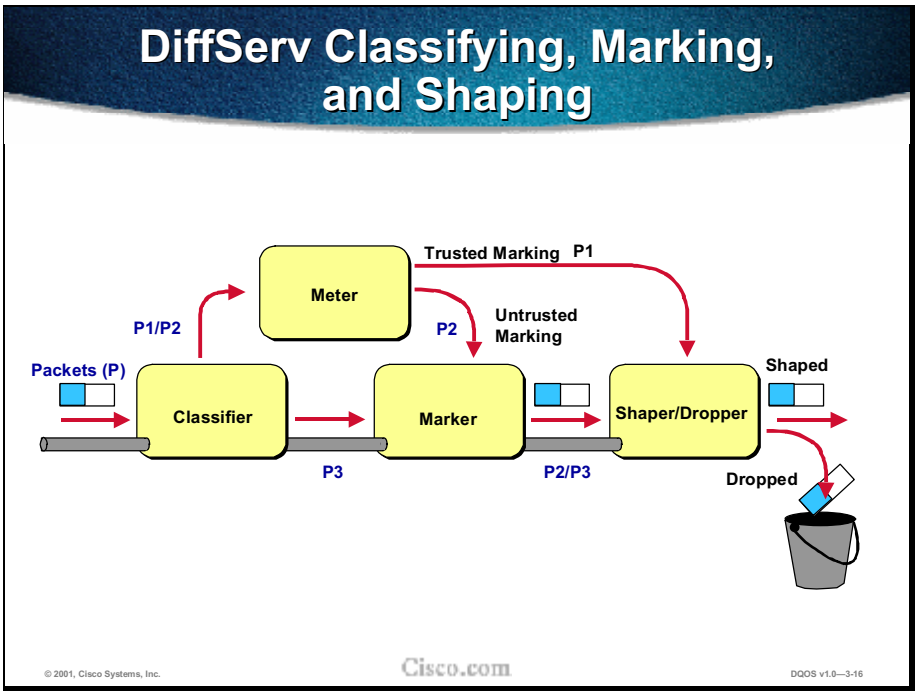
Differentiated Services (DiffServ) describes a set of end-to-end quality-of-service (QoS) capabilities.

As stated in the IETF DiffServ working group objectives: “The differentiated service approach to providing quality of service in networks employs a small, well-defined set of building blocks from which a variety of aggregate behaviors may be built. A small bit pattern in each packet, in the IPv4 ToS octet or the IPv6 Traffic Class octet, is used to mark a packet to receive a particular forwarding treatment, or per-hop behavior, at each network node. A common understanding about the use and interpretation of this bit pattern is required for interdomain use, multivendor interoperability, and consistent reasoning about expected aggregate behaviors in a network. Thus, the working group has standardized a common layout for a six-bit field of both octets, called the DS field.” [http://www.ietf.org/html.charters/diffserv-charter.html]

In order to deliver end-to-end QoS, this architecture (RFC-2475) has two major components—packet marking using the IPv4 ToS byte, and per-hop behaviors (PHBs). Each of these behaviors is defined in the next pages. Typically, a PHB represents the scheduling and discard priorities a packet should receive on a router interface.

For a discussion of DiffServ, see *White Paper: DiffServ—The Scalable End-to-End QoS Model* at the following URL:

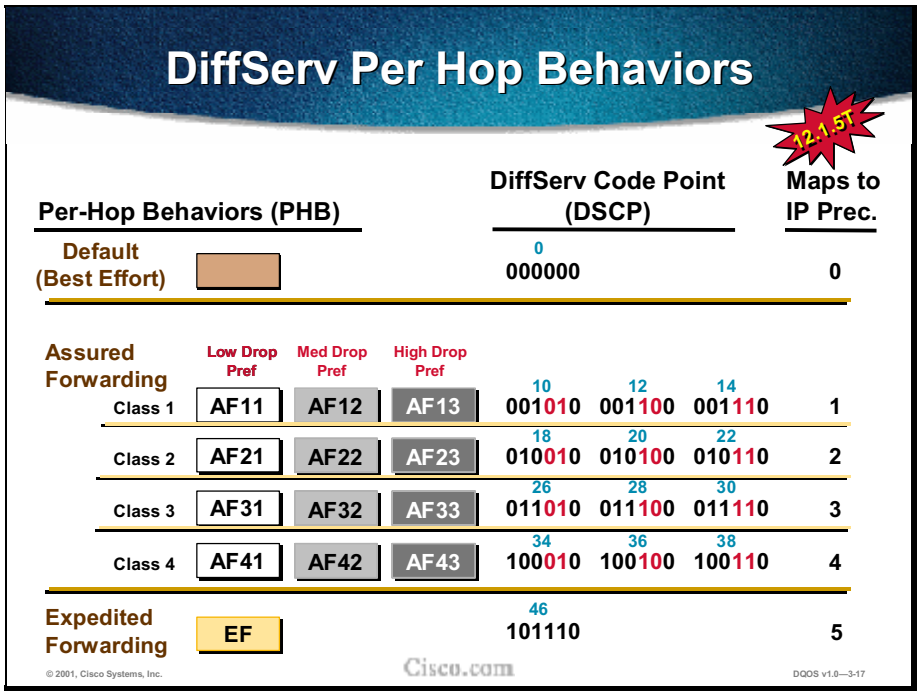
http://www.cisco.com/warp/public/cc/pd/iosw/ioft/iofwft/prodlit/difse_wp.htm



With DiffServ, you can use packet classification to partition network traffic into multiple priority levels or classes of service. When traffic descriptors are used to classify traffic, the source agrees to adhere to the contracted terms and the network promises a quality of service. Traffic policers and traffic shapers use the traffic descriptor of the packet (that is, the classification of the packet) to ensure adherence to that agreement. Packet classification uses the DSCP traffic descriptor to categorize a packet within a specific group in order to define that packet. After the packet has been defined (that is, classified), the packet is then accessible for QoS handling on the network.

Packet marking is related to packet classification. Packet marking allows you to classify a packet based on the DSCP value. This classification can then be used to apply user-defined differentiated services to the packet and to associate a packet with a local QoS group. Associating a packet with a local QoS group allows users to associate a group ID with a packet. The group ID can be used to classify packets into QoS groups based on prefix, autonomous system, and community string. A user can set up to 64 DSCP values and 100 QoS group markings.

A traffic conditioner finally shapes (buffers to achieve a target flow rate) or drops the packet in case of congestion. The slide illustrates the typical traffic conditioner at the edge of a DiffServ domain. A DiffServ “internal node” enforces the appropriate PHB by employing policing or shaping techniques and sometimes re-marking out of profile packets, depending on the policy or the SLA.



A per-hop behavior (PHB) is the externally observable forwarding behavior applied at a DiffServ-compliant node to a DiffServ behavior aggregate (BA).

With the ability of the system to mark packets according to DSCP setting, collections of packets — each with the same DSCP setting and sent in a particular direction — can be grouped into a BA. Packets from multiple sources or applications can belong to the same BA.

In other words, a PHB refers to the packet scheduling, queuing, policing, or shaping behavior of a node on any given packet belonging to a BA, as configured by a service level agreement (SLA) or a policy map.

The four available standard PHBs are:

- Default PHB (as defined in RFC 2474)
- Assured forwarding (AFny) PHB (as defined in RFC 2597)
- Expedited forwarding (EF) PHB (as defined in RFC 2598) Class-selector PHB (as defined in RFC 2474)

The default PHB specifies that a packet marked with a DSCP value of 000000 (recommended) receive best-effort service from a DS-compliant node. If a packet arrives at a DS-compliant node and the DSCP value is not mapped to any other PHB, the packet is mapped to the default PHB.

Assured forwarding PHB defines four AF*ny* classes (*n*=1-4: AF1, AF2, AF3, and AF4). Each class is assigned a specific amount of buffer space and interface bandwidth, according to the

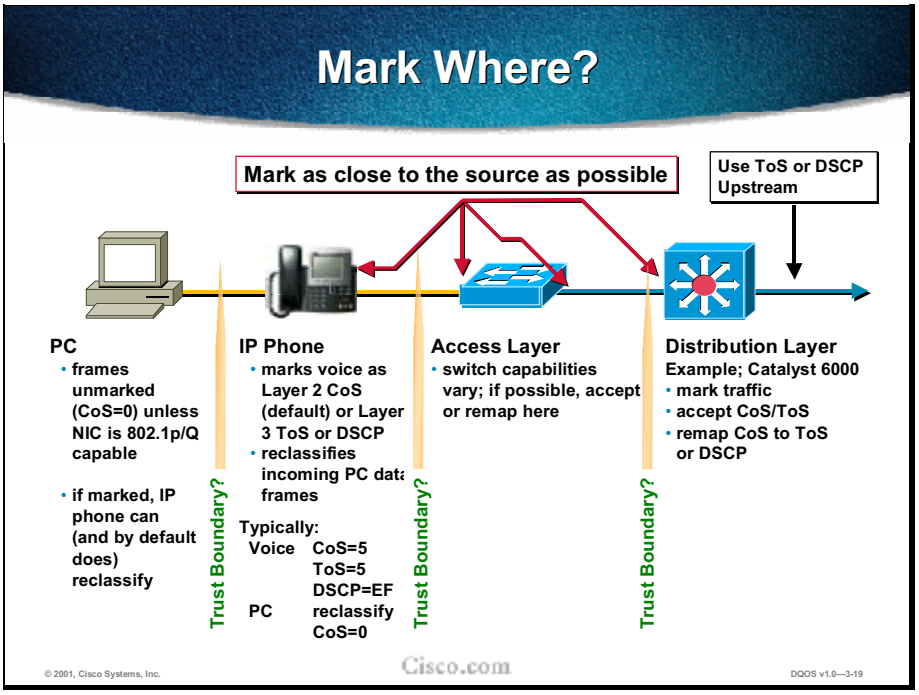
service level agreement (SLA) with the service provider or policy map. Within each AF class, you can specify three drop precedence values (*y*=1-3: 1, 2, and 3). Packets in the AF13 class are dropped before packets in the AF12 class, which in turn are dropped before packets in the AF11 class. The figure lists the DSCP value and corresponding drop precedence value for each AF PHB class.

The EF PHB provides low-loss, low-latency, low-jitter, and assured bandwidth service. It is comparable to the Integrated Services Resource Reservation Protocol (RSVP), which provides guaranteed bandwidth service. EF PHB should be reserved for the most critical applications since, in instances of traffic congestion, it is not feasible to treat all or most traffic as high priority. EF PHB is ideally suited for applications such as VoIP that require low bandwidth, guaranteed bandwidth, low delay, and low jitter. The recommended DSCP value for EF PHB is 101110.

The class-selector PHB is the PHB associated with a class-selector code point. DSCP values called class-selector code points (in the form xxx000, where x is either 0 or 1) preserve backward-compatibility with IP Precedence. (The DSCP value for a packet with default PHB 000000 is also called the class-selector code point.) For example, packets with a DSCP value of 110000 (the equivalent of the IP Precedence-based value of 110) have preferential forwarding treatment (for scheduling, queuing, and so on), as compared to packets with a DSCP value of 100000 (the equivalent of the IP Precedence-based value of 100).

For a detailed discussion of DiffServ classification and marking, see *Implementing DiffServ for End-to-End Quality of Service* at the following URL:

http://cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/dtdfsv.htm



Classification should take place at the network edge, typically in the wiring closet or within video endpoints or IP phones themselves. The slide demonstrates this with an IP telephony example. Packets can be marked as important by using Layer 2 class of service (CoS) settings in the user priority bits of the 802.1p portion of the 802.1p/Q field or the IP Precedence/differentiated services code point (DSCP) bits in the ToS/DS field in the IPv4 header. Cisco IP Phones can mark voice packets as high priority using CoS as well as ToS. By default, the phone sends 802.1p tagged packets with the CoS and ToS set to a value of 5. Because most PCs do not have an 802.1Q capable network interface card (NIC), they send the packets untagged. This means that the frames do not have an 802.1p field. Also, unless the applications running on the PC send packets with a specific CoS value, this field is zero. A special case is where the TCP/IP stack in the PC has been modified to send all packets with a ToS value other than zero. Typically this does not happen, and the ToS value is zero.

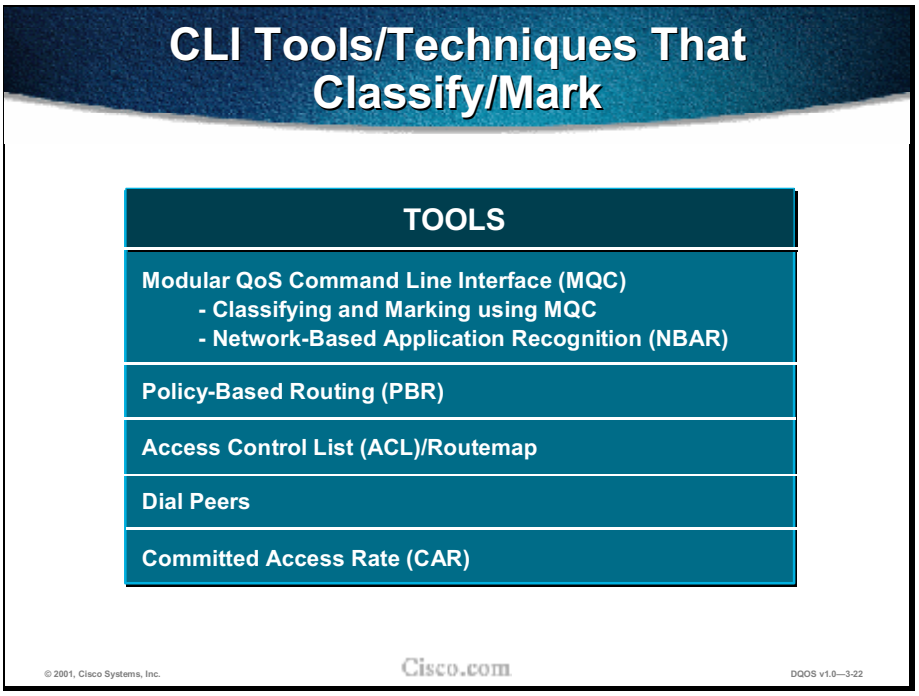
Even if the PC is sending tagged frames with a specific CoS value, Cisco IP Phones can zero out this value before sending the frames to the switch. This is the default behavior. Frames coming from the phone have a CoS of 5 and frames coming from the PC have a CoS of 0. When the switch receives these frames, it can take into account these values for further processing, based on its capabilities.

The switch uses its queues (available on a per-port basis) to buffer incoming frames before sending them to the switching engine (it is important to remember that input queuing comes into play only when there is congestion). The switch uses the CoS value(s) to put the frames into appropriate queues. The switch can also employ mechanisms such as weighted random early detection (WRED) to make intelligent drops within a queue (also known as congestion avoidance) and weighted round-robin (WRR) to provide more bandwidth to some queues than to others (also known as congestion management).

Trust Boundaries The concept of trust is an important and integral one to deploying QoS. Once the end devices have set CoS or ToS values, the switch has the option of trusting them. If the switch trusts the values, it does not need to do any reclassification; if it does not trust the values, then it must perform reclassification for the appropriate QoS.

The notion of trusting or not trusting forms the basis for the trust boundary. Ideally, classification should be done as close to the source as possible. If the end device is capable of performing this function, the trust boundary for the network is at the access layer in the wiring closet. If the device is not capable of performing this function, or the wiring closet switch does not trust the classification done by the end device, the trust boundary might shift. How this shift happens depends on the capabilities of the switch in the wiring closet. If the switch can reclassify the packets, the trust boundary remains in the wiring closet. If the switch cannot perform this function, the task falls to other devices in the network, going toward the backbone. In this case, the rule of thumb is to perform reclassification at the distribution layer. This means that the trust boundary has shifted to the distribution layer. It is more than likely that there is a high-end switch in the distribution layer with features to support this function. If possible, try to avoid performing this function in the core of the network.


Tools and Techniques
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—3-21



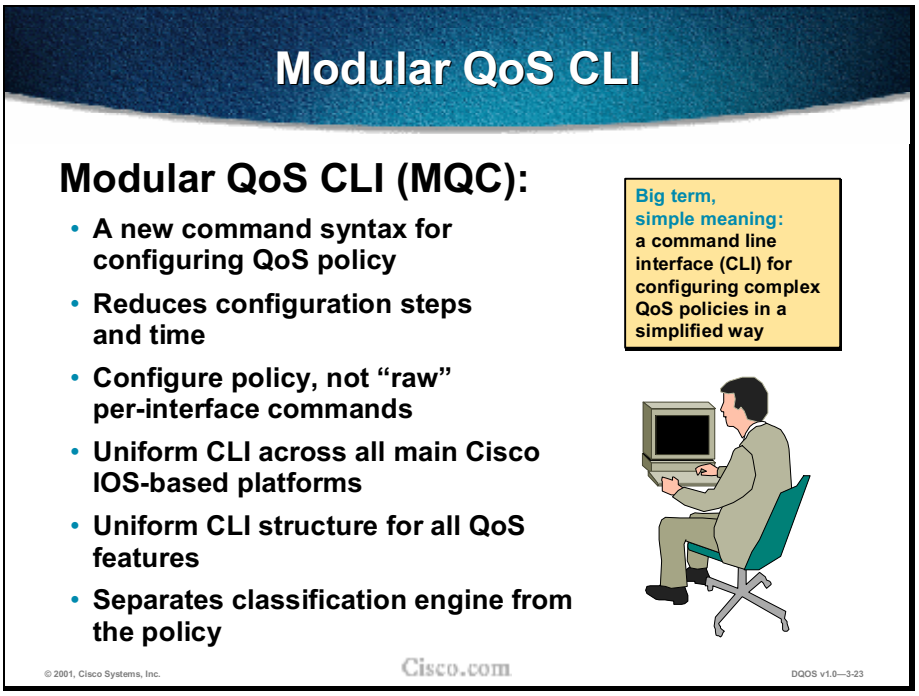
The features listed in the slide can be used to classify/mark traffic.

- **MQC**—Cisco's Modular QoS command-line interface (MQC) provides a convenient and efficient user interface for configuring QoS.
  - **Classification and Marking**: MQC provides an efficient interface for a basic, straightforward, three-step structure for implementing a QoS policy.
  - **NBAR**: Network-based application recognition (NBAR) is an MQC-enabled classification and protocol-discovery feature. NBAR can determine the mix of traffic on the network, which is important in isolating congestion problems.
- **PBR**–Policy-based routing (PBR) can be used with a route map on an interface to classify and mark packets. The route map does the classification (matches patterns in packet addressing or other characteristics), then the **set** condition does the marking (alters the IP Precedence).

- **ACL**—An access control list (commonly used to permit or deny access to a line or interface) and a route map (commonly used to permit or deny access to a receiving interface) can be used together in implementing QoS policies and features.

- **Dial Peers**—To give real-time voice traffic a higher priority than other network traffic, you can weight the voice data traffic associated with a particular VoIP dial peer by using IP Precedence.

- **CAR**—In addition to its policing capability, committed access rate (CAR) can classify and mark traffic.

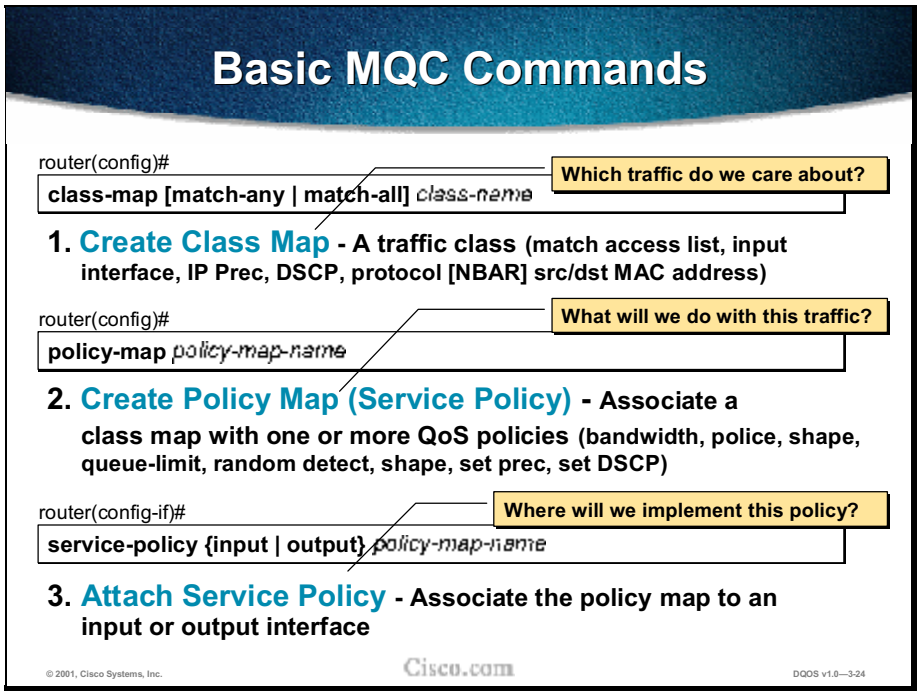


Cisco's Modular QoS command-line interface (MQC) provides a convenient and efficient user interface for configuring QoS. Because MQC can be used to configure various QoS features covered throughout this course, a brief introduction is given here.

MQC is available across all main Cisco IOS-based platforms, beginning with Cisco IOS Release 12.0(5)T. MQC is a new, more advanced way of configuring QoS.

For a detailed discussion of MQC, see *Modular Quality of Service Command-Line Interface* at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120xe/120xe5/mqc/mcli.htm



These commands provide the basic three-step structure for implementing a QoS policy.

1. The **class-map** command defines a traffic class (class map). **Match** commands specify various criteria for classifying packets. Packets that match the specified criteria are forwarded according to the QoS specifications set in the service policy. Packets that fail to meet any of the matching criteria are classified as members of the default class.

   – A **class-map match-any** command does a logical OR of the match conditions following it. A **class-map match-all** command does a logical AND. If neither **match-all** nor **match-any** is specified, the traffic class behaves in a manner consistent with **class-map match-all** command.

2. The **policy-map** command specifies the QoS policies (policy map) to apply to traffic classes defined by a class map. The actual policies are covered in more detail in other chapters.

3. The **service-policy** command attaches a policy map to an input or output interface.

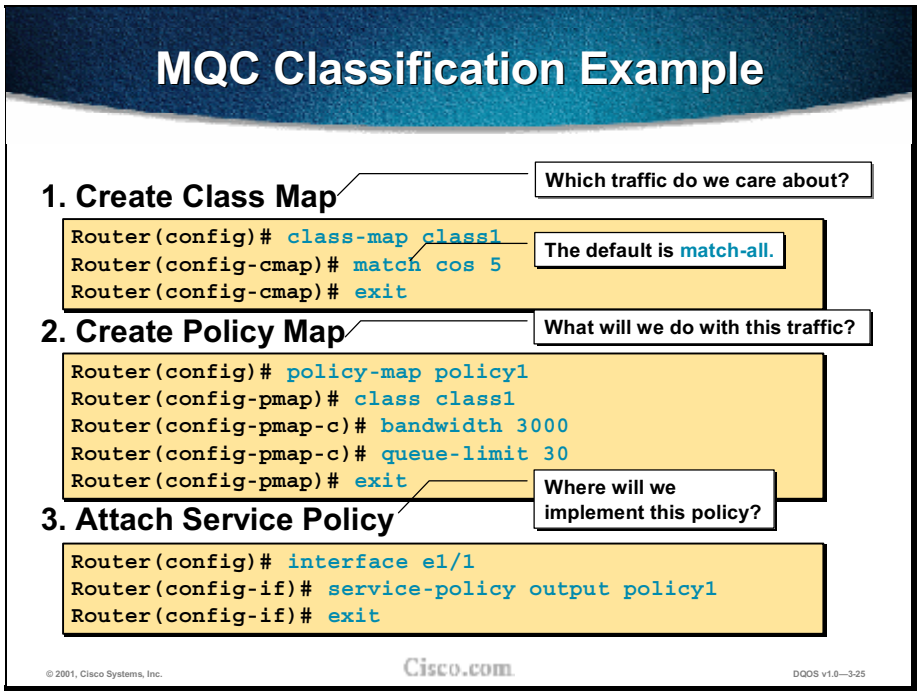
The **show policy-map** command displays the configuration of a policy map and its associated class maps.

See *Modular Quality of Service Command-Line Interface* for a complete descriptions and syntax of all MQC commands. It is located at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120xe/120xe5/mcli.htm

See also:

- http://cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_r/qrdcmd3.htm
- http://cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_r/qrdcmd1.htm

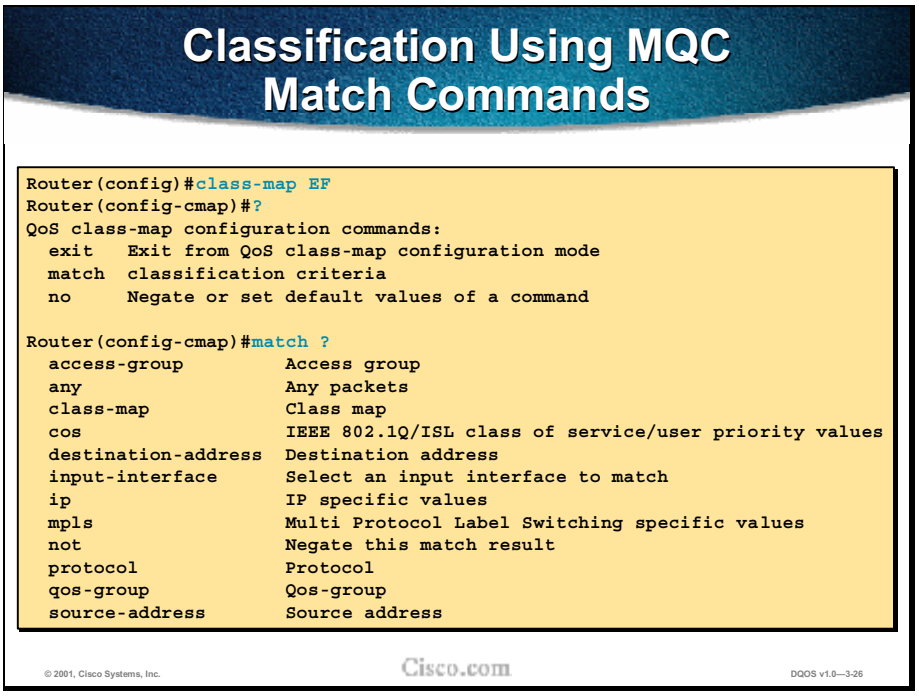


In Step 1 of the example, a traffic class called class1 is created. Its one match criterion is to use packets that have the 802.1p class of service value set to 5. Packets that meet the match criterion belong to the class.

In Step 2, a service policy named policy1 is defined. The **class** command associates the policy map with a previously defined class map (in this case *class1*). That is, policy map policy1 contains policy specifications for the class class1. The **class** command must be entered immediately after entering policy map configuration mode, which is indicated by the prompt (*config-pmap*).

The policy named policy1 includes a bandwidth allocation request and a maximum packet count limit for the queue reserved for the class.

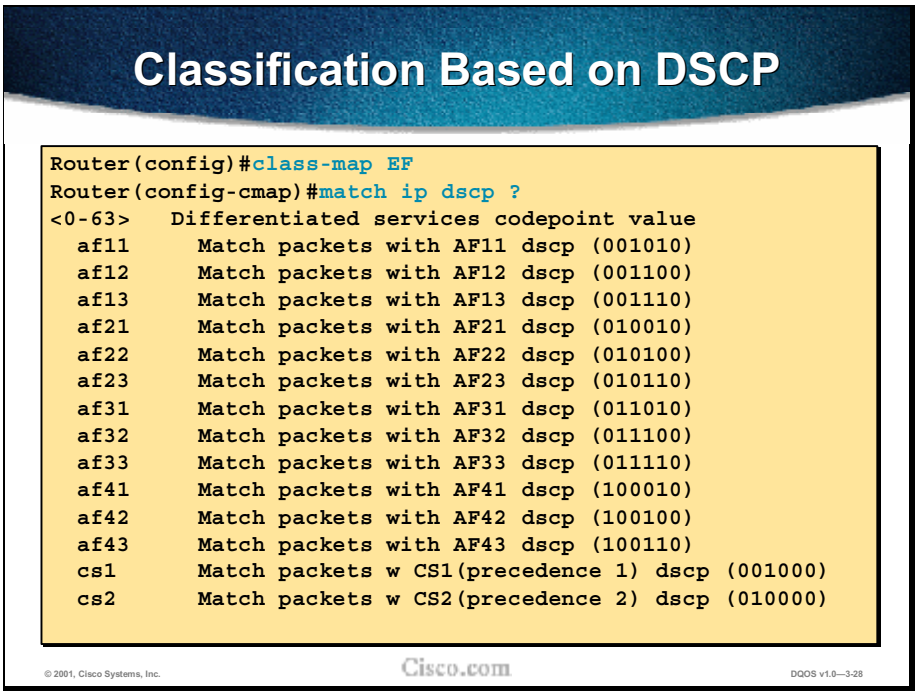
Step 3 shows how to attach an existing service policy to an interface. After you define a service policy with the **policy-map** command, use the **service-policy** command in interface configuration mode to attach the service policy. In this way you specify the service policy for those interfaces. Although you can assign the same service policy to multiple interfaces, each interface can have only one service policy attached at the input and only one service policy attached at the output.



The slide shows various MQC match criteria available for defining a class of service.

- **class-map match-all** *class-name*: Specifies a logical AND operator for all matching statements under this traffic class. When neither **match-all** nor **match-any** is specified, the default is **match-all**.
- **class-map match-any** *class-name*: Specifies a logical OR operator for all matching statements under this traffic class.
- **match input-interface** *interface-name*: Specifies the name of the input interface used as a match criterion against which packets are checked to determine if they belong to the class.
- **match source-address mac** *address*: Specifies the name of the source MAC address used as a match criterion against which packets are checked to determine if they belong to the class.
- **match destination-address mac** *address*: Specifies the name of the destination MAC address used as a match criterion against which packets are checked to determine if they belong to the class.
- **match access-group** *access-list-number*: Specifies the numbered access list against whose contents packets are checked to determine if they belong to the class.
- **match ip dscp** *number*: Specifies up to eight differentiated services code point (DSCP) values used as match criteria. The value of each service code point is from 0 to 63.
- **match cos**: Specifies from one to four Layer 2 class of service 802.1p/ISL values (0-7).

- **match ip rtp**: Specifies the Real-Time Transport Protocol (RTP) port as the match criterion.
- **match mpls experimental**: Specifies the value of the EXP field as a match criterion.
- **match ip precedence** *number*: Specifies up to eight IP Precedence values used as match criteria.
- **match qos-group** *number*: Specifies the number of the QoS group index used as a match criterion against which packets are checked to determine if they belong to the class.
- **match protocol** *protocol*: Specifies the name of the protocol used as a match criterion against which packets are checked to determine if they belong to the class (NBAR).
- **match class-map** *class-name*: Specifies the name of a traffic class to be used as a matching criterion (for nesting traffic classes [nested class maps] within one another).
- **match any**: Specifies that all packets are matched.
- **match not match-criteria:** Specifies a match criterion value that prevents packets from being classified as members of a specified traffic class. All other values of that particular match criterion belong to the class.

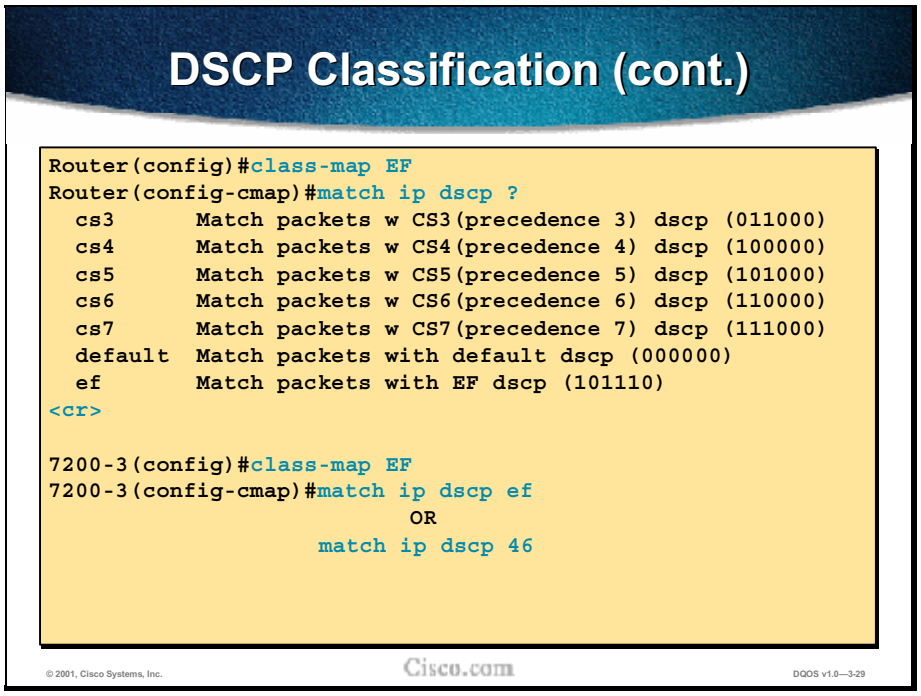


This example begins with creation of a class map named EF. Before continuing with this example on the next page, note in the slide that entering a question mark (?) after the command **match ip dscp** results in a display of possible arguments for the command. As shown in the first column of the help (?) output, you can enter either a number (0 to 63) for the DSCP value or an alphanumeric string that matches one of the standard DiffServ PHBs. Selecting a standard PHB sets the DSCP value to the value for that PHB; for example, setting the DSCP value to AF31 is the same as setting the DSCP value to 26.

Note also that the possible arguments listed so far in this slide include the assured forwarding PHBs and the beginning of a list of class-selector PHBs.

Recall from the earlier DiffServ discussion that the assured forwarding (AF) PHB comprises four predefined AF classes for prioritizing traffic and that within each class there are three additional predefined prioritizations (drop precedences). Thus, the AF PHB predefines 12 traffic classes.

Recall also that the class-selector code points (in the form xxx000, where x is either 0 or 1) preserve backward-compatibility with IP Precedences and that the class-selector PHB is the PHB associated with a class-selector code point. The slide lists the first two of the seven class-selector code points, which map to IP Precedence 1 and IP Precedence 2.

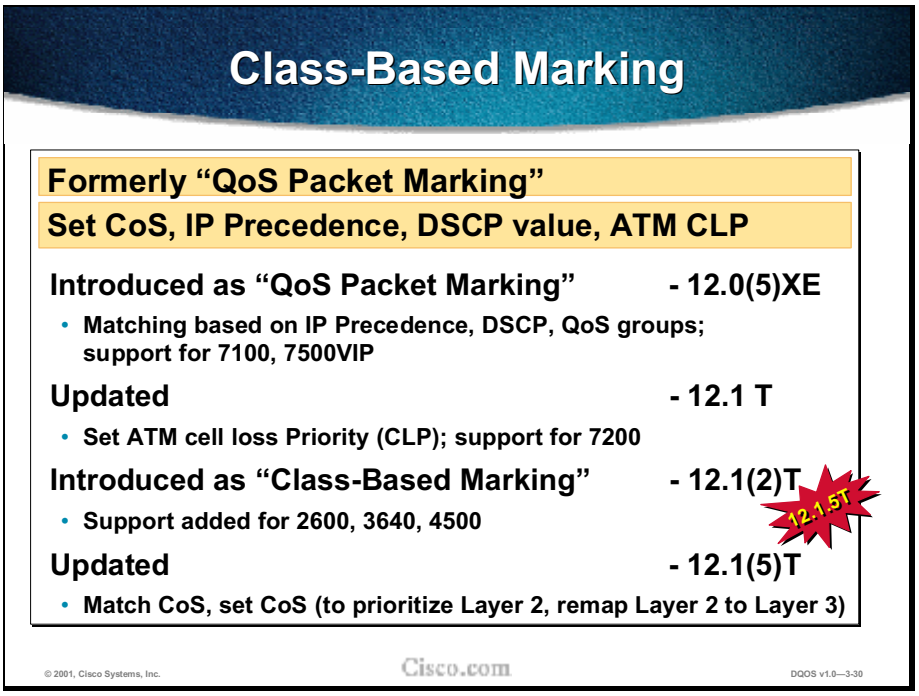


Continuing with the example of DSCP classification using MQC, notice that following the remaining class-selector code points, the help (?) display lists the default and expedited forwarding (EF) PHBs.

Recall from the DiffServ discussion that the default PHB specifies that a packet marked with a DSCP value of 000000 receives best-effort service from a DS-compliant node.

Recall also that the EF PHB is comparable to the integrated services Resource Reservation Protocol (RSVP), which provides guaranteed bandwidth service, and should be reserved for the most critical applications, such as VoIP, that require low bandwidth, guaranteed bandwidth, low delay, and low jitter.

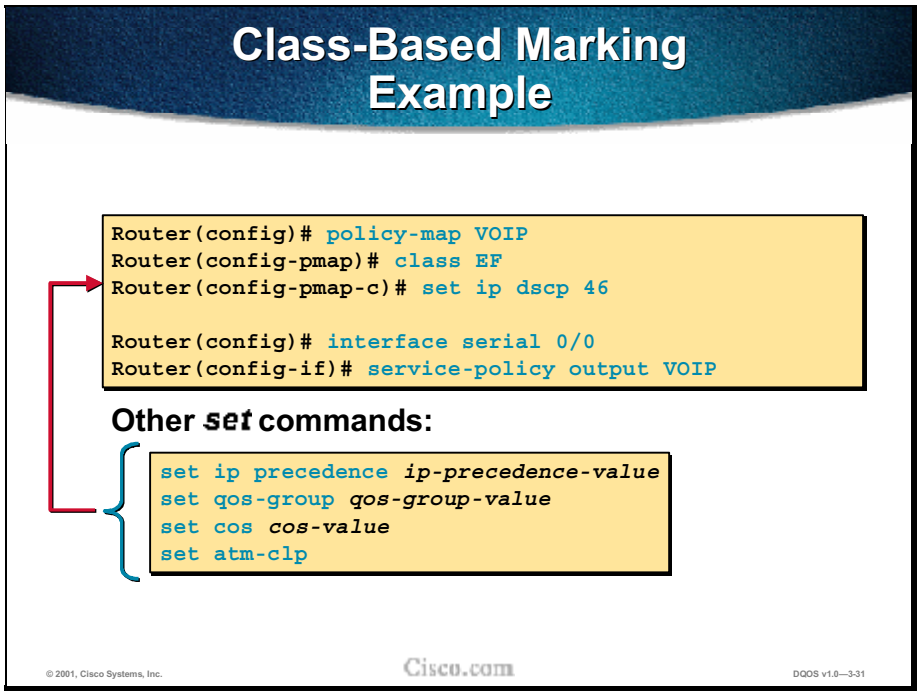
Now, to complete the example, which began with creation of a class map named EF (this first step repeated in the slide for convenience), the second step is to specify that the class map includes traffic that has the DSCP set to EF (which is the same as setting the DSCP to 46). Thus, the traffic in this example is classified as traffic with the DSCP set to EF.



Class-based packet marking is an MQC capability that was introduced originally by Cisco as "QoS Packet Marking." Class-based packet marking provides a user-friendly command-line interface (CLI) for efficient packet marking, including:

- Layer 3 marking—IP Precedence, DSCP
- Layer 2 marking—CoS for 802.p/Q, ISL
- ATM marking—to reset the cell loss priority (CLP) bit in the ATM packet header from 0 to 1
- Associating a local quality of service (QoS) group value (0-99) with a packet

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/cbpmark2.htm#xtocid253640



To mark packets using class-based marking:

1. Create a policy map: **policy-map** *policy-name*Associate the policy map with a previously defined class map: **class** *class-name*

3. Enter one of the following set commands:

   - **set ip precedence** ip-precedence-value
   - **set ip dscp** ip-dscp-value
   - **set qos-group** qos-group-value
   - **set cos** cos-value
   - **set atm-clp**

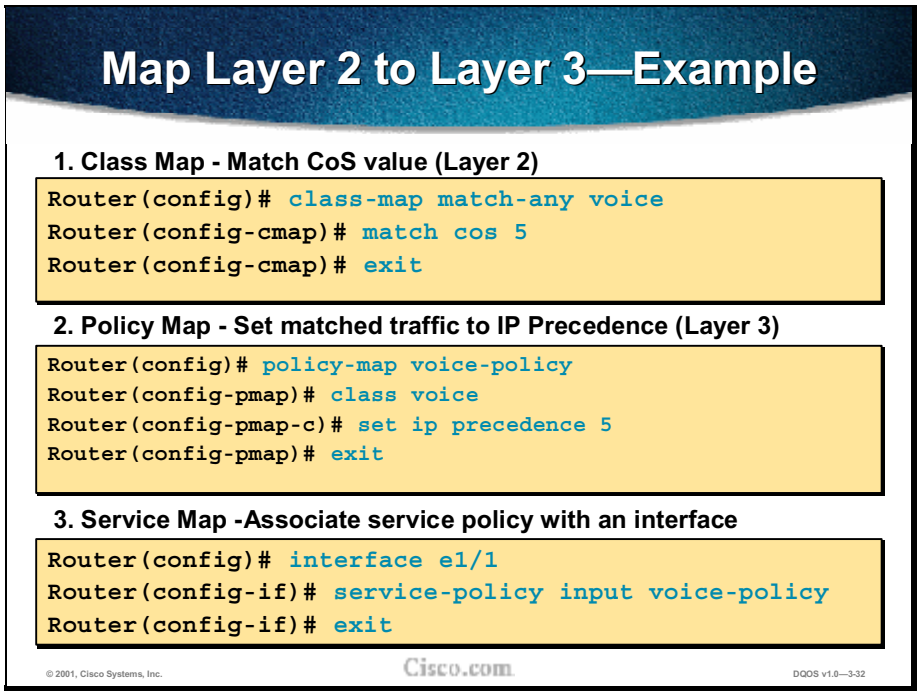
Note that this is simply the MQC policy map creation step. In the example, the policy map VOIP is created.

Before packet marking, create the class map with which the policy map is to be associated. In the example, the previously created class map is named EF.

Following packet marking, associate the policy map with an interface (that is, create a service policy). This is done in the last two lines of the example.

For a detailed discussion of class-based marking, see *Class-Based Marking* at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/cbpmark2.htm



Using the 802.1p bits within the 802.1p/Q field provides the desired QoS results at Layer 2. When traffic has to cross a Layer 3 boundary, however, it becomes imperative to implement these mechanisms using Layer 3 parameters, such as IP Precedence or DSCP.

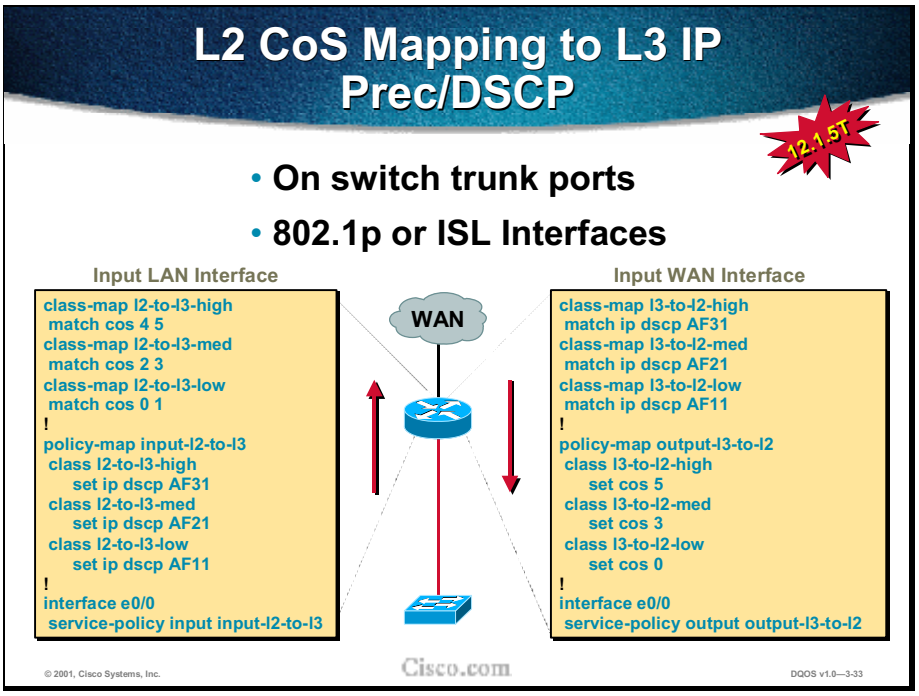
Traffic crosses a Layer 3 boundary when packets are routed between subnets by Layer 3 switches or routers. Traffic also crosses a Layer 3 boundary when packets need to go out of the campus network onto the WAN through edge routers. When this happens, Layer 2 classification does not help. All of the QoS techniques employed by the routers (including the very important WAN QoS) rely on Layer 3 classification.

Layer 3 classification can be achieved by using the appropriate platforms in the campus. Beginning with IP phones, for example, packets are already presented to the switch with CoS=ToS=5. This Layer 3 classification is preserved even if the packets travel all the way through to the WAN edge router, where the Layer 2 header is removed. If the trust boundary is at the source (for example, at the IP phone), voice traffic has the ToS bits set to 5 and is presented to the network devices for appropriate treatment. WAN routers can use this classification to employ any of the queuing techniques. If the trust boundary is not at the source and packets need to be reclassified, the device performing this function should be capable of doing it at Layer 3 before it can cross a Layer 3 boundary.

In the example, the class map voice is configured to match traffic that has the Layer 2 CoS value set to 5. The policy map voice-policy sets the IP Precedence for matching traffic to 5, which maps this traffic for end-to-end QoS. Finally, the policy is attached to one or more interfaces.

For further discussion of reclassification of Layer 2 traffic to Layer 3, see, for example, *Campus Infrastructure Considerations* at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/voice/ip_tele/network/dgcampus.htm

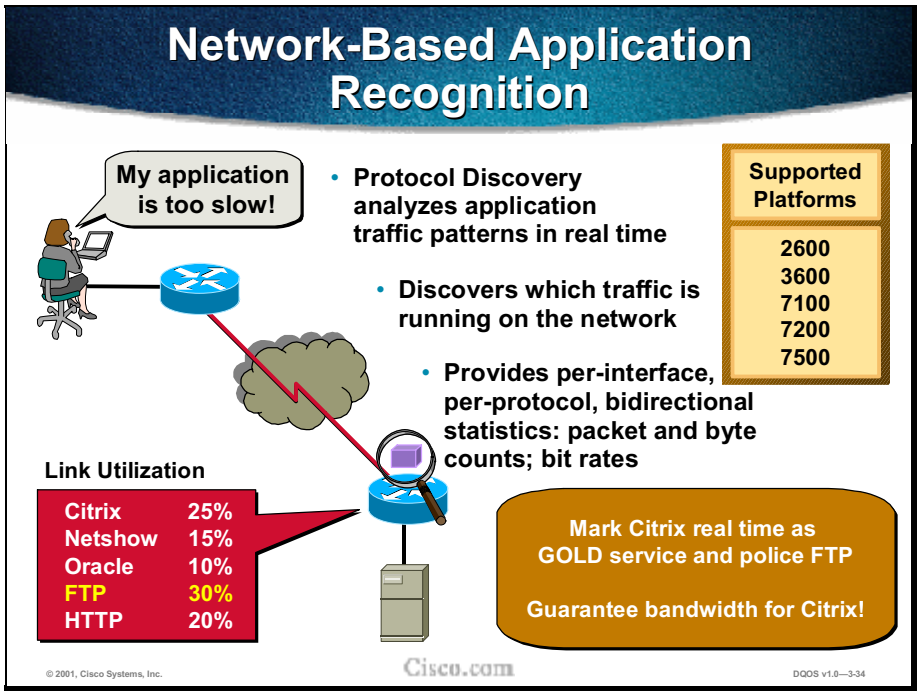


This slide provides a complex mapping example. At the input LAN interface, Layer 2 CoS is mapped to Layer 3 DSCP to prepare CoS-marked traffic for output to the WAN interface. At the input WAN interface the opposite occurs; Layer 3 DSCP values are mapped to Layer 2 CoS values to prepare DSCP-marked traffic for campus QoS handling.

Note that in each example three classes of traffic are set up and that three policy maps are configured, one for each class of traffic (low, medium, and high priority). Note also that more than one value can be specified in a **match** command. Up to four values can be specified in a **match cos** or a **match ip precedence** command; up to eight values can be specified in a **match ip dscp** command.

In the input LAN interface example, traffic set to CoS 4 or 5 is classified as high priority in the class map named *l2-l3-high* (think of the class map name as reading, "*Layer 2 to Layer 3, High Priority*"). Traffic set to CoS 2 or 3 is classified as medium priority in the class map named *l2-l3-med*. Traffic set to CoS 0 or 1 is classified as low priority in the class map named *l2-l3-low*. The policy map named *input-l2-to-l3* (read as, "*Input: Layer 2 to Layer 3*") sets high-priority traffic (CoS 4 or 5) to a DSCP setting of AF11 (for example, to the PHB that sets DSCP to 10) medium-priority traffic (CoS 2 or 3) to AF21 (DSCP=18), and low-priority traffic (CoS 0 or 1) to AF31 (DSCP=26). Finally, this policy is attached to the input LAN interface.

On the input WAN interface, the policy map *output-l3-to-l2* (read as, "*Output: Layer 3 to Layer 2*") essentially reverses the mapping process for traffic that is coming from the WAN and is destined for the LAN and that matches any of the three class maps defined for this interface. Notice that AF11 traffic is set to CoS=5, AF21 traffic is set to CoS=3, and AF31 traffic is set to CoS=0.

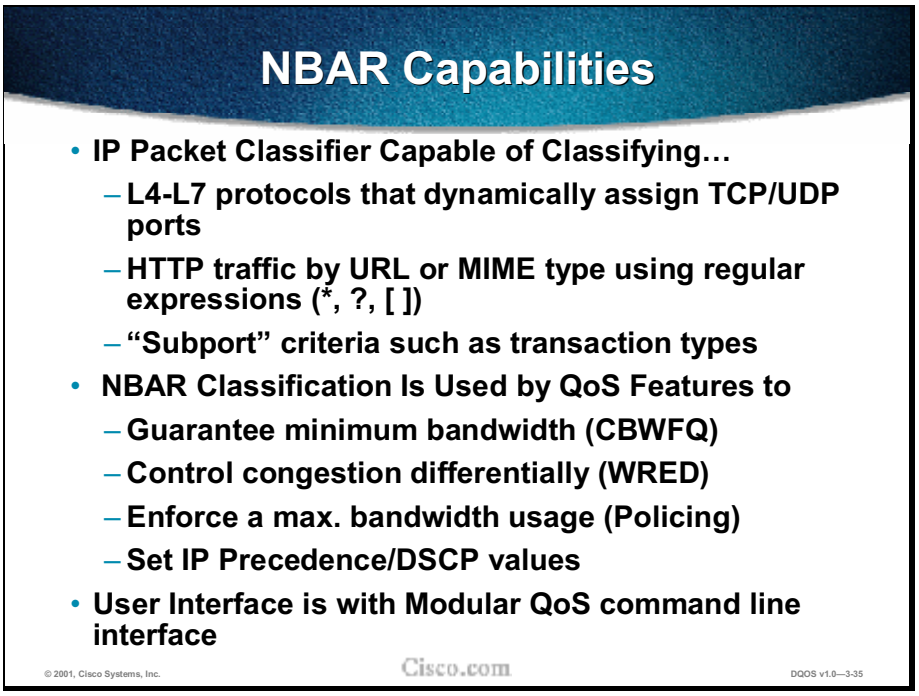


Network-based application recognition (NBAR) is an MQC-enabled classification and protocol-discovery feature. NBAR can determine the mix of traffic on the network, which is important in isolating congestion problems.

NBAR can classify application traffic by looking beyond the TCP/UDP port numbers of a packet. This is subport classification. NBAR looks into the TCP/UDP payload itself and classifies packets on content within the payload, such as transaction identifier, message type, or other similar data.

Classification of HTTP by URL or MIME type is an example of subport classification. NBAR classifies HTTP traffic by text within the URL, using regular expression matching. NBAR uses the UNIX filename specification as the basis for the URL specification format. The NBAR engine then converts the specification format into a regular expression.

NBAR can classify Citrix Independent Computing Architecture (ICA) traffic and perform subport classification of Citrix traffic, based on Citrix published applications. NBAR can monitor Citrix ICA client requests for a published application destined to a Citrix ICA Master browser. After the client requests to the published application, the Citrix ICA Master browser directs the client to the server with the most available memory. The Citrix ICA client then connects to this Citrix ICA server for the application.



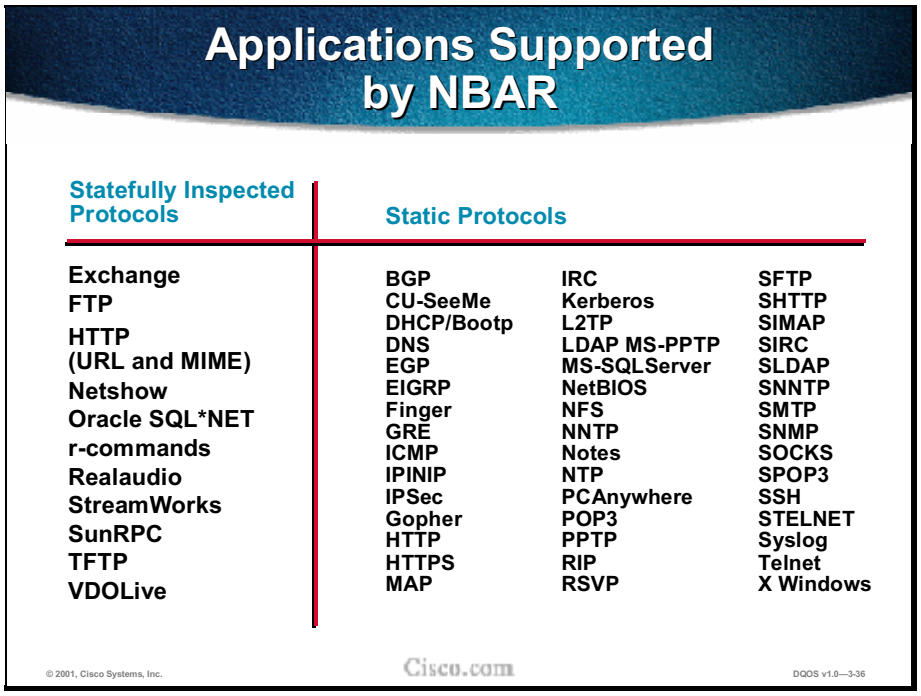
NBAR ensures that network bandwidth is used efficiently by working with QoS features to provide:

- Guaranteed bandwidth
- Bandwidth limits
- Traffic shaping
- Packet coloring

NBAR introduces several new classification features:

- Classification of applications that dynamically assign TCP/UDP port numbers
- Classification of HTTP traffic by URL, host, or MIME type
- Classification of Citrix ICA traffic by application name
- Classification of application traffic using subport information

NBAR can also classify static port protocols. Although access control lists (ACLs) can be used for this purpose, NBAR is easier to configure and can provide classification statistics that are not available when using ACLs. NBAR can classify application traffic by looking beyond the TCP/UDP port numbers of a packet. This is subport classification.



The NBAR protocol-discovery feature provides an easy way to discover application protocols that are transiting an interface. The protocol-discovery feature discovers any protocol traffic supported by NBAR. Protocol discovery can be applied to interfaces and can be used to monitor both input and output traffic. Protocol discovery maintains the following per-protocol statistics for enabled interfaces: total number of input and output packets and bytes, and input and output bit rates.

NBAR supports simpler configuration coupled with stateful recognition of flows. The simpler configuration means you don’t have to do a protocol analyzer capture to figure out ports and so on. Stateful recognition means smarter, deeper packet recognition.

Applications Supported by NBAR (cont.)

AppleTalk	DECnet_node	PPPoE
ARP	DECnet_router_L1	Printer
Bridge	DECnet_router_L2	QLLC
Bstun	DLSW	RCMD
CDP	IMAP	RSRB
Citrix	IPX	SQLServer
CLNS	IRC	STUN
CLNS_ES	LDAP	VINES
CLNS_IS	llc2	VOFR
CMNS	Novadigm	XNS
Compressedtcp	PAD	
DECNET	PCAnywhere	

Packet Description Language Module

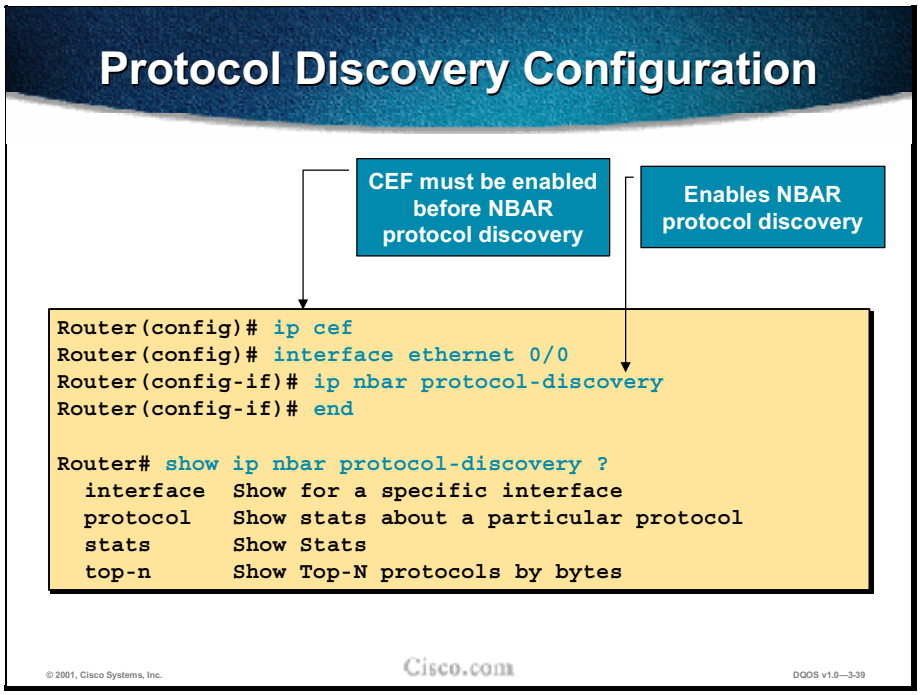
- PDLMs define applications recognizable by NBAR
- New applications easily supported by adding new PDLMs
- No Cisco IOS software upgrade or reboot required to add new PDLMs
- PDLMs must be produced by Cisco engineers

An external packet description language module (PDLM) can be loaded at run time to extend the NBAR list of recognized protocols. PDLMs can be used to enhance an existing protocol recognition capability. PDLMs allow NBAR to recognize new protocols without requiring a new Cisco IOS image or a router reload.

New PDLMs are released only by Cisco and are available from local Cisco representatives. They can be loaded from flash memory. Registered users can find them at: http://www.cisco.com/cgi-bin/tablebuild.pl/pdlm

To extend or enhance the list of protocols recognized by NBAR through a Cisco-provided PDLM, use the **ip nbar pdlm** configuration command. Use the **no** form of this command to unload a PDLM if it was previously loaded.

Use the **show ip nbar port-map** command to display the current protocol-to-port mappings in use by NBAR.

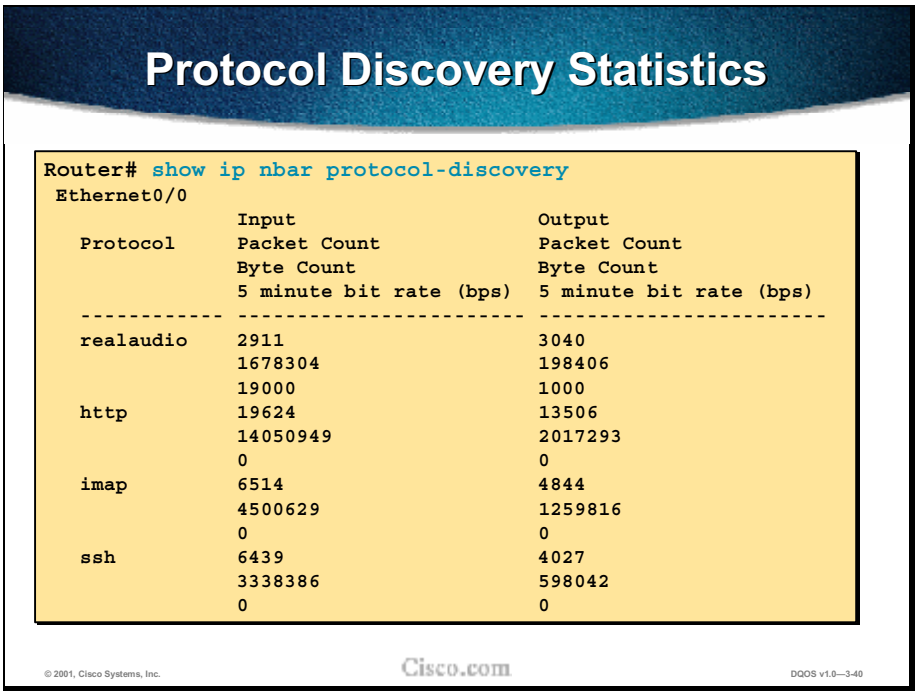


Use the **ip nbar protocol-discovery** command to configure NBAR to keep traffic statistics for all protocols known to NBAR. Protocol discovery provides an easy way to discover application protocols transiting an interface so that QoS policies can be developed and applied. The protocol-discovery feature discovers any protocol traffic supported by NBAR. Protocol discovery can be used to monitor both input and output traffic and can be applied with or without a service policy enabled.

Once the command is configured, use **show** commands to display the required information. The default output of this command includes input bit rate, input byte count, input packet count, and protocol name. Statistics are displayed for each enabled protocol for both incoming and outgoing packets.

Note in the example that you must enable Cisco Express Forwarding (CEF) before you configure NBAR. For more information on CEF, refer to *Cisco Express Forwarding Overview* at the following URL:

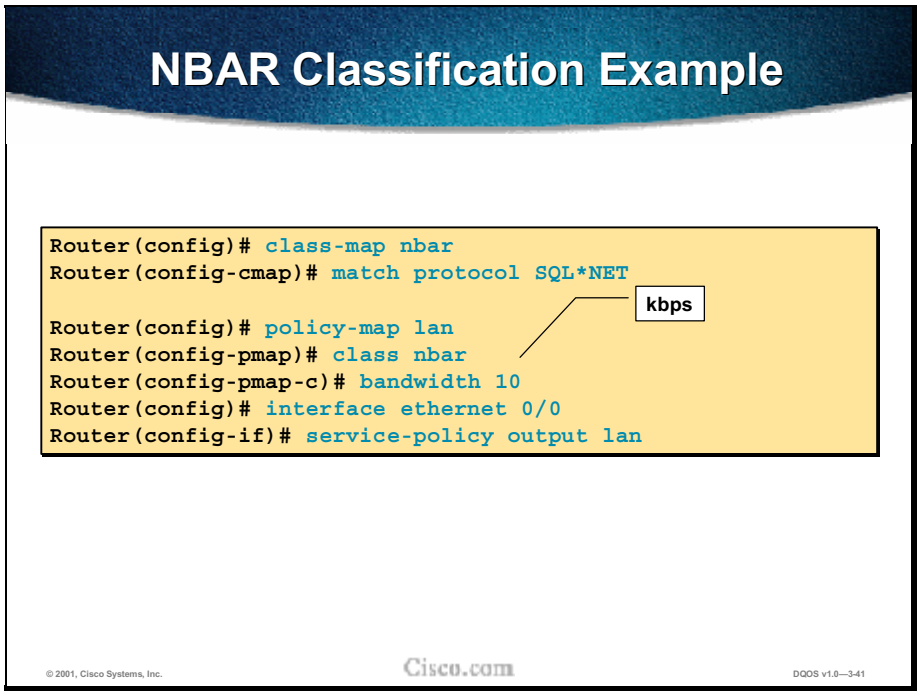
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/switch_c/xcprt2/xcdcef.htm.



Use the **show ip nbar protocol-discovery** command to display statistics gathered by the NBAR protocol-discovery feature. This command, by default, displays statistics for all interfaces on which protocol discovery is currently enabled. The default output of this command includes, in the following order, input bit rate (bps), input byte count, input packet count, and protocol name.

Protocol discovery can be used to monitor both input and output traffic and can be applied with or without a service policy enabled. NBAR protocol discovery gathers statistics for packets switched to output interfaces. These statistics are not necessarily for packets that exited the router on the output interfaces, because packets might have been dropped after switching for various reasons, including policing at the output interface, access lists, or queue drops.

The example displays partial output of the **show ip nbar protocol-discovery** command for an Ethernet interface.



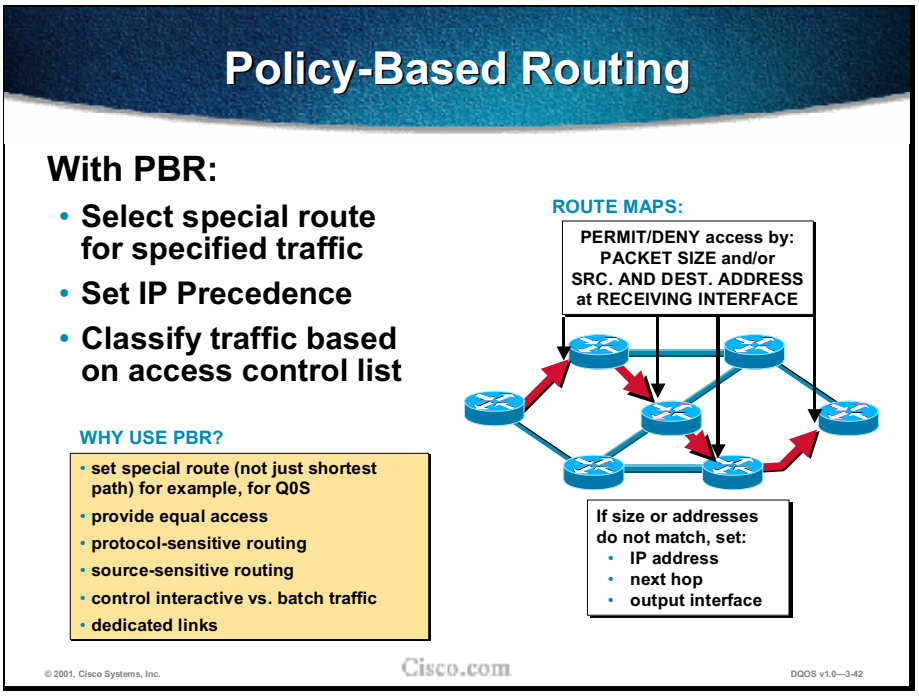
Once a protocol has been discovered and its statistics analyzed, the **match protocol** command can be used to apply a policy to that protocol.

In the example, the class map named *nbar* creates a traffic class consisting of the protocol SQL*NET. The policy map *lan* then specifies that traffic for this protocol is to receive 10 kbps. The service policy is attached to the Ethernet interface on which NBAR analyzed protocol traffic, which resulted in the decision to apply this policy.

In addition to the **match protocol** command, which can be used with a long list of protocols familiar to NBAR, there also are two special-case match commands. These are **match protocol http** and **match protocol citrix**.

For a description of all NBAR features and commands, see *Network-Based Application Recognition* at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/dtnbar.htm



Originally, policy-based routing (PBR) was a way to examine the source address or the interface a packet arrived on and determine the next routing hop or the outbound interface. Now PBR can be used with a route map on an interface to classify and mark packets. The **route-map** does the classification (matches patterns in packet addressing or other characteristics), then the **set** condition does the marking (alters the IP Precedence).

Policy routing provides the means to define customized routing paths for selected packets, based on criteria such as source address and application port, not normally considered by destination-based routing protocols. Thus, particular traffic types, such as voice traffic, can be sent over special routes that minimize hop counts and other delay characteristics to ensure high-quality service characteristics. Policy routing can be used on low-end and midrange routers to classify packets and mark the packets via the IP Precedence field, enabling backbone routers to give priority treatment to voice packets when congestion occurs.

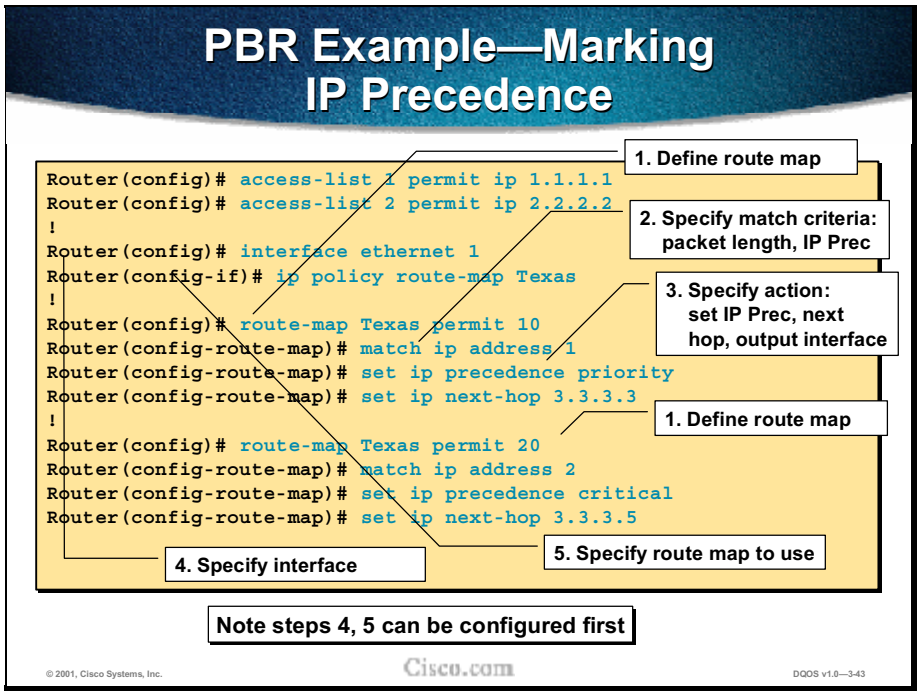
For a succinct description of PBR, see *Quality of Service Fact Sheet* at the following URL:

http://www.cisco.com/warp/public/cc/so/neso/vvda/avvid/eeqos_ds.htm

For additional information on PBR, see also *Quality of Service Overview* at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcdintro.htm

For more detailed information on PBR, see *Configuring Policy-Based Routing* at the following URL:http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcprt1/qcdpbr.htm



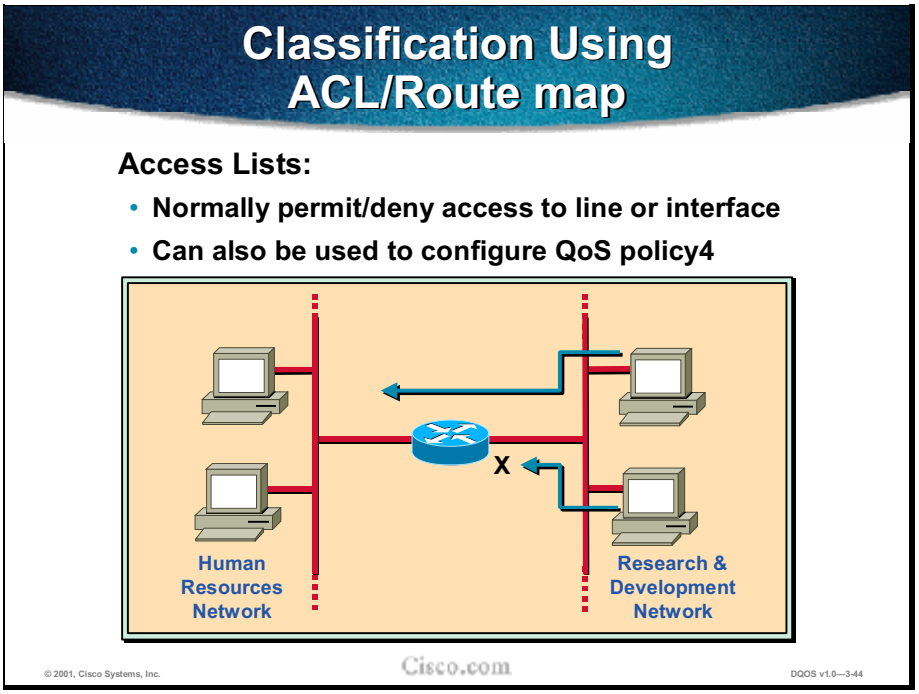
You specify PBR on the interface that receives the packet, not on the interface from which the packet is sent.

To enable PBR on an interface, beginning in global configuration mode:

- **Step 1**—Define a route map to control where packets are output. This command puts the router into route-map configuration mode.
- **Step 2**—Specify the match criteria. You can match the Level 3 length of the packet and/or the source and destination IP address. If you do not specify a **match** command, the route map applies to all packets.
- **Step 3**—Specify the action or actions to take on the packets that match the criteria. You can specify the IP Precedence, next hop, output interface, default next hop, or default output interface.
- **Step 4**—Specify the interface. This command puts the router into interface configuration mode.
- **Step 5**—Identify the route map to use for PBR. One interface can have only one route-map tag, but you can have multiple route-map entries with different sequence numbers. These entries are evaluated in sequence number order until the first match. If there is no match, packets are routed as usual.

The example illustrates how to route traffic from different sources to different places (next hops) and how to set the precedence bit in the IP header. Packets arriving from source 1.1.1.1 are sent to the next hop at 3.3.3.3 with the precedence bit set to priority; packets arriving from source 2.2.2.2 are sent to the next hop at 3.3.3.5 with the precedence bit set to critical.

Note Enabling PBR disables fast switching of all packets arriving on this interface.



An access control list (ACL) is commonly used to permit or deny access to a line or interface. A route map is commonly used to permit or deny access to a receiving interface. These tools can also be used together in implementing QoS policies and features.

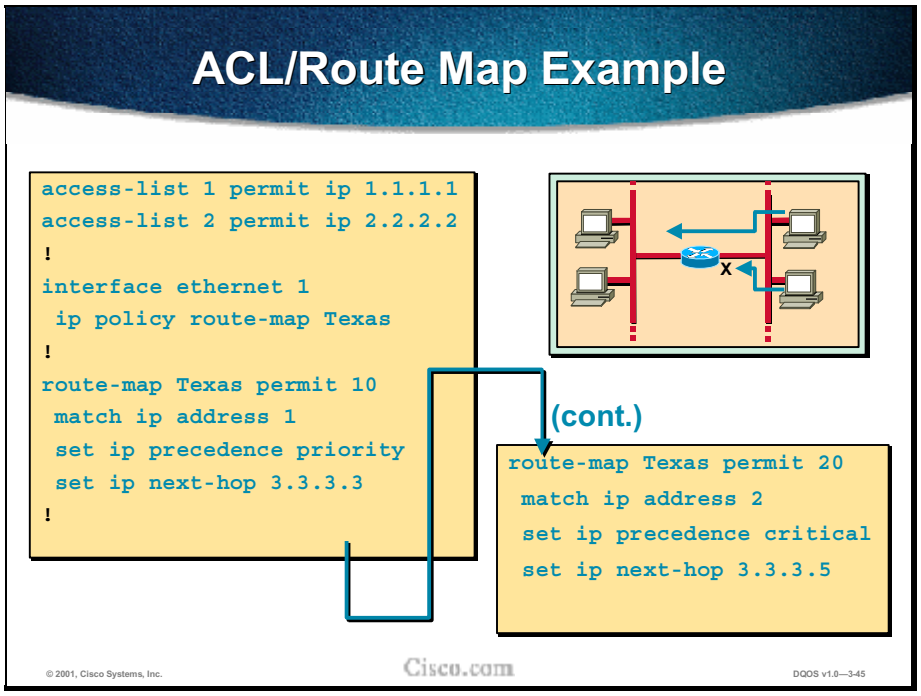
An ACL can be configured either by using the router CLI or by using a text editor and creating an ASCII text file that can be transferred from a TFTP server to the router. For the sake of simplicity, examples in this section use the CLI method.

For an overview of configuring ACLs for all protocols, see *Access Control Lists: Overview and Guidelines*, at the following URL:

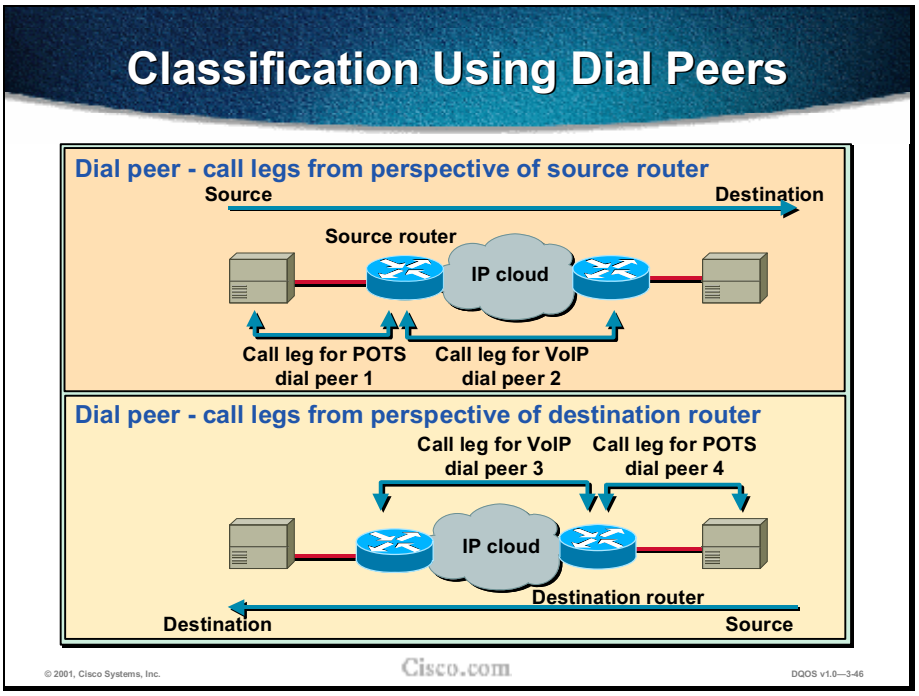
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_c/scprt3/scdacls.htm

For details on configuring ACLs for the IP protocol, see the “Configuring IP Services” chapter in the *Cisco IOS IP and IP Routing Configuration Guide*, at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/ip_c/ipcprt1/1cdip.htm



The example illustrates how to route traffic from different sources to different places (next hops) and how to set the precedence bit in the IP header. Packets arriving from source 1.1.1.1 are sent to the next hop at 3.3.3.3 with the precedence bit set to *priority*; packets arriving from source 2.2.2.2 are sent to the next hop at 3.3.3.5 with the precedence bit set to *critical*.



Dial peers are used to apply attributes to call legs and to identify call origin and destination. In other words, a dial peer defines a route for a phone number; it determines how you get to a particular phone number.

There are two different kinds of dial peers:

- POTS (for analog phone calls)
- VoIP (for IP phone calls)

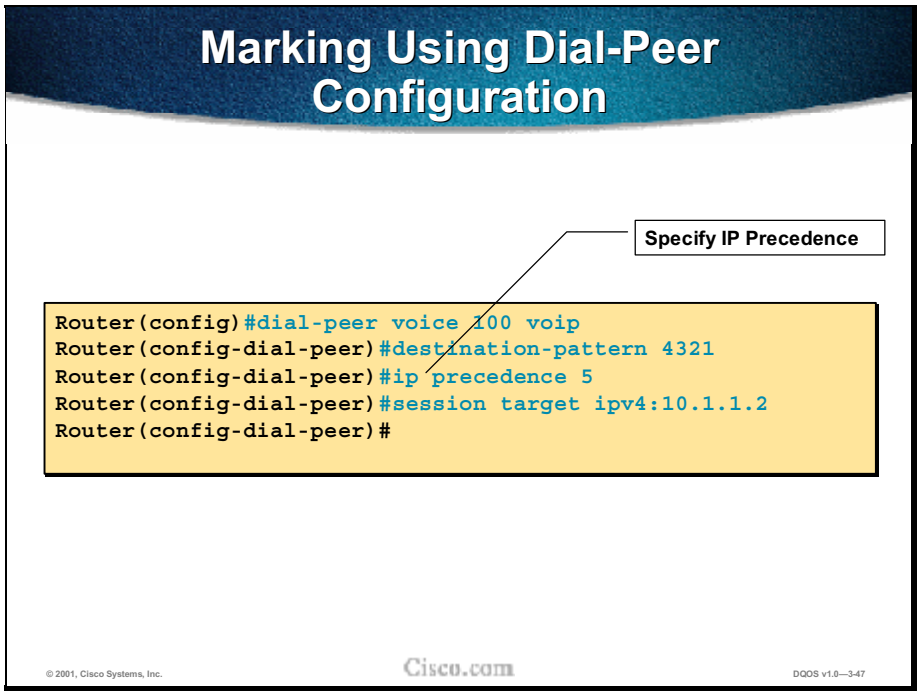
A call leg is a segment of a call connection between two points in that connection. All the call legs for a particular connection have the same connection ID.

An end-to-end call can have up to four call legs, two from the perspective of the source access server and two from the perspective of the destination access server. A dial peer is associated with each call leg. Attributes applied to a call leg include QoS, codec, VAD, and fax rate.

Dial peers are used for both inbound and outbound call legs.

For a detailed description of the role of dial peers, see *Configuring Voice over IP* at the following URL:

http://cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/multi_c/mcprt1/mcdvoip.htm#xtocid232308

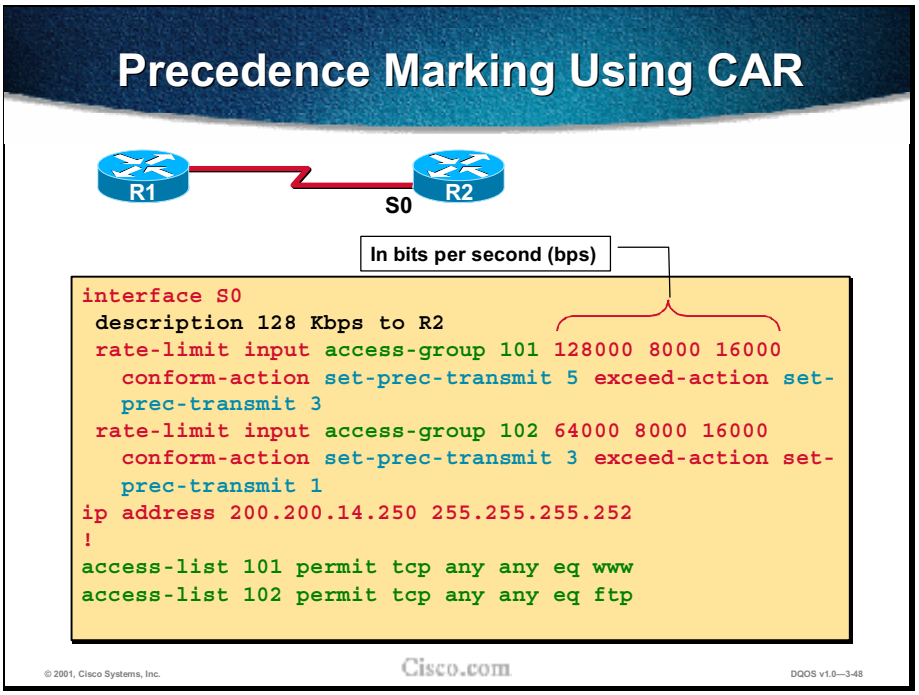


To give real-time voice traffic a higher priority than other network traffic, you can weight the voice data traffic associated with a particular VoIP dial peer by using IP Precedence.

The example ensures that voice traffic associated with VoIP dial peer 100 is given a higher priority than other IP network traffic.

For a detailed explanation of configuring dial peers for IP Precedence, see the section "Configuring IP Precedence for Dial Peers" in *Configuring Voice over IP* at the following URL:

http://cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/multi_c/mcprt1/mcdvoip.htm#xtocid2323015

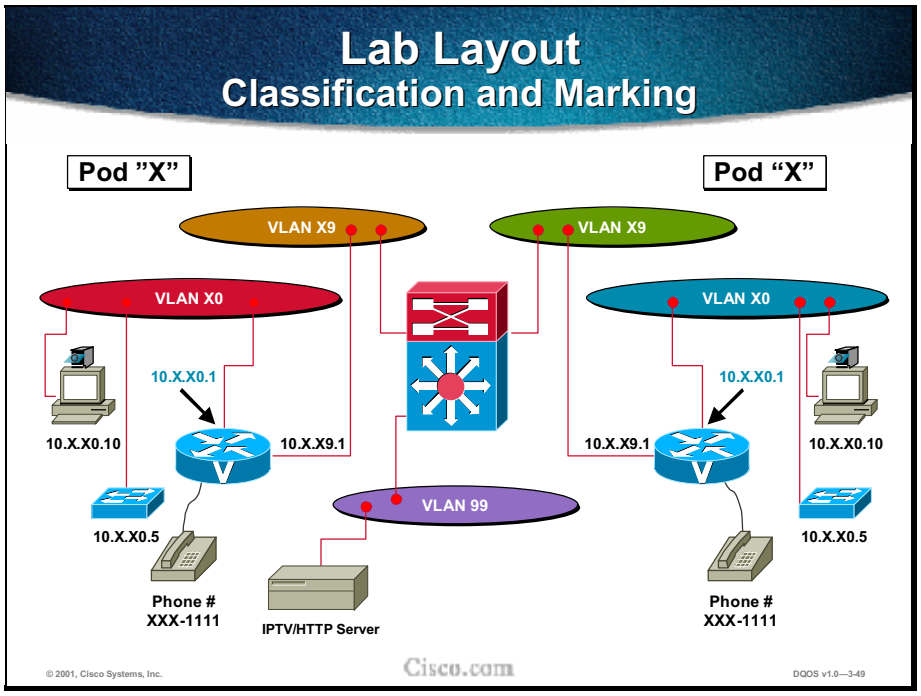


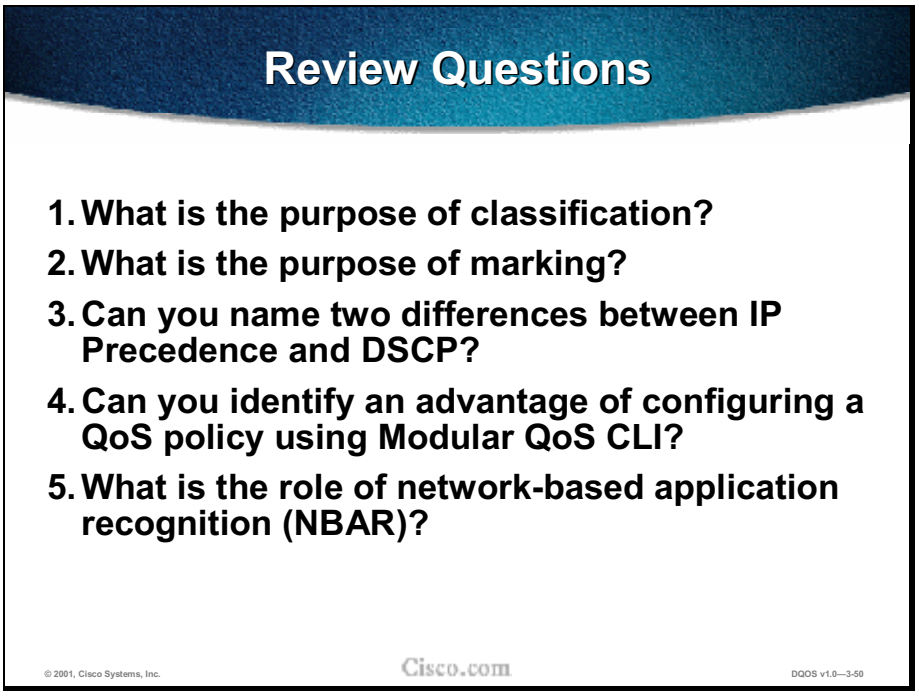
Committed Access Rate (CAR) is the predecessor to the MQC **police** command. Policing (controlling traffic flow) is covered in another chapter. The example in this chapter shows that CAR also can classify (specify which traffic is in which class) and mark (set the IP Precedence bits).

In the example, access list 101 specifies Web traffic, and 102 specifies FTP traffic. We rate-limit Web traffic to 128 kbps. If the traffic conforms to that limit, set the precedence to 5; otherwise set it to 3 (best effort). FTP is rate-limited to 64 kbps; if it conforms, the precedence is set to 3; if it exceeds, precedence is set to 1.

For a detailed discussion of both the policing and the classification and marking aspects of CAR, see *Configuring Committed Access Rate* at the following URL:

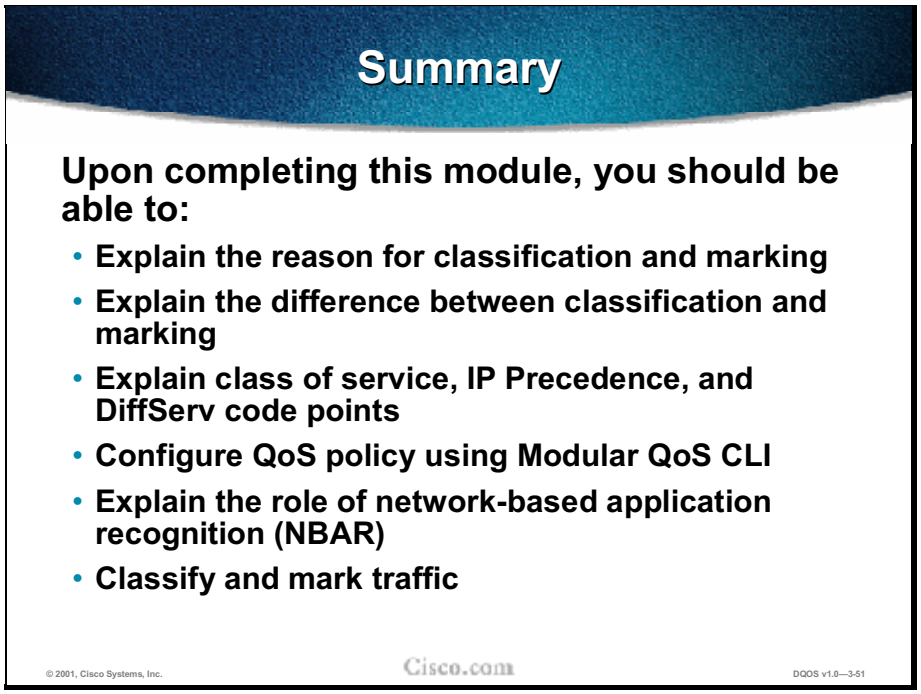
http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/qos_c/qcpart1/qccar.htm


Lab Layout
Classification and Marking
Pod "X"
Pod "X"
VLAN X9
VLAN X9
VLAN X0
VLAN X0
10.X.X0.1
10.X.X0.1
10.X.X0.10
10.X.X9.1
10.X.X9.1
10.X.X0.10
VLAN 99
10.X.X0.5
10.X.X0.5
Phone #
XXX-1111
IPTV/HTTP Server
Phone #
XXX-1111
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—3-49



1. What is the purpose of classification?
2. What is the purpose of marking?
3. Can you name two differences between IP Precedence and DSCP?
4. Can you identify an advantage of configuring a QoS policy using Modular QoS CLI?
5. What is the role of network-based application recognition (NBAR)?

Summary


Summary
Upon completing this module, you should be able to:
• Explain the reason for classification and marking
• Explain the difference between classification and marking
• Explain class of service, IP Precedence, and DiffServ code points
• Configure QoS policy using Modular QoS CLI
• Explain the role of network-based application recognition (NBAR)
• Classify and mark traffic
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—3-51

4

Congestion Management

Overview

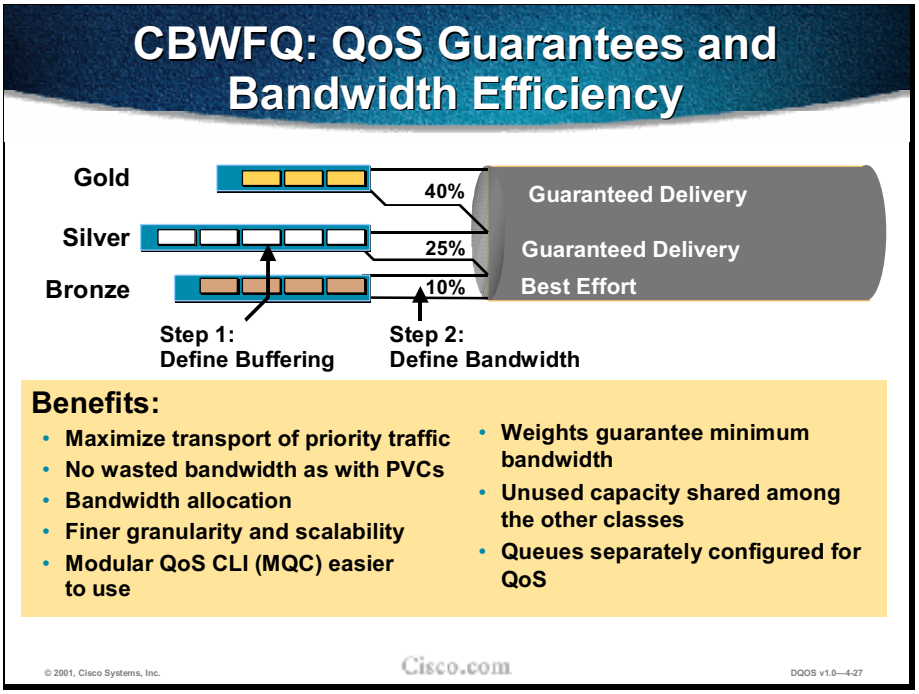
This chapter explains the different queuing techniques including class-based WFQ and low latency queuing. The chapter compares and contrasts the different queuing mechanisms and shows how these mechanisms can be configured.

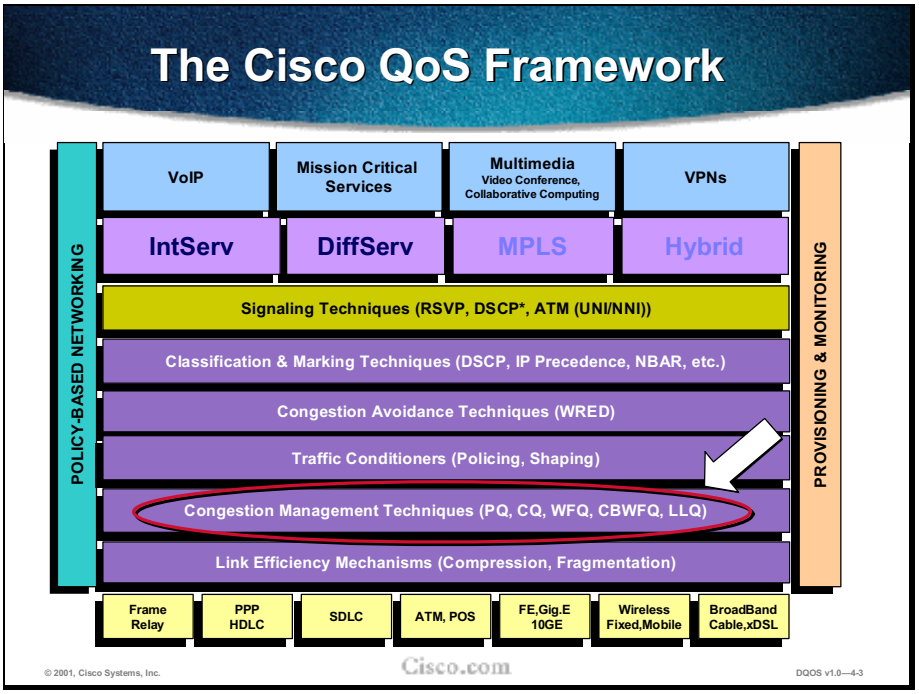
Objectives

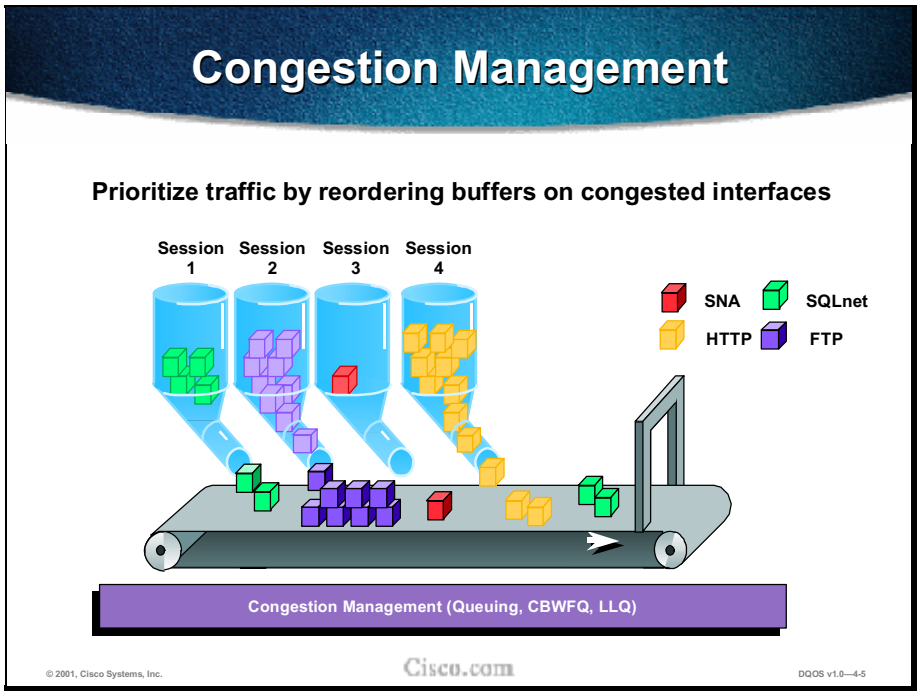
Upon completing this chapter, you will be able to:

- Identify and differentiate between the different IOS queuing techniques
- Correctly apply each queuing technique to the appropriate application
- Describe the difference between IP RTP Priority and LLQ
- Configure WFQ, CBWFQ, and LLQ

Outline







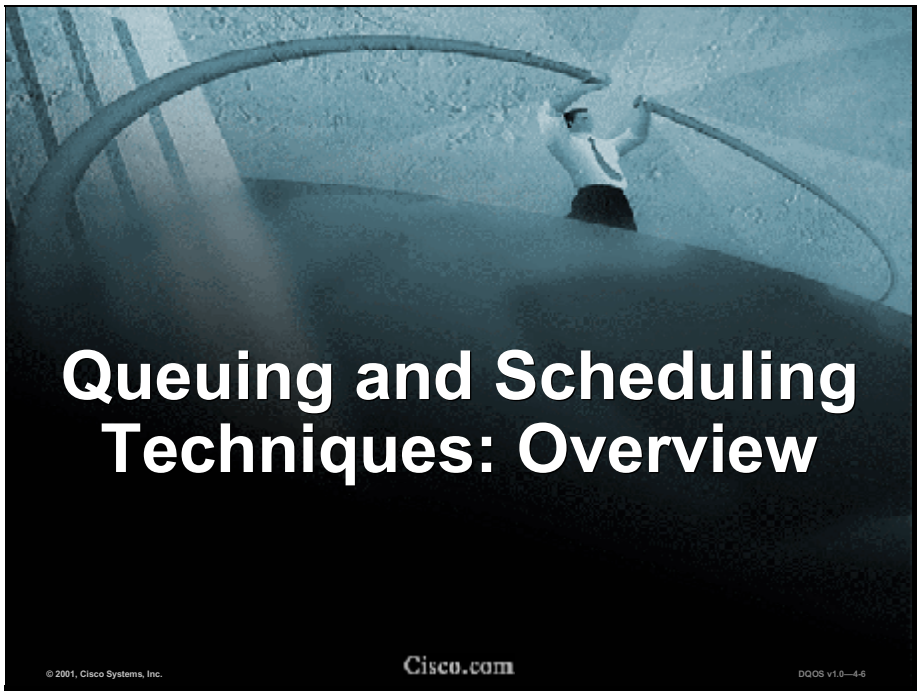
Congestion management features operate to control congestion once it occurs. One way that network elements handle an overflow of arriving traffic is to use a queuing algorithm to sort the traffic and determine some method of prioritizing it onto an output link. Each queuing algorithm was designed to solve a specific network traffic problem and has a particular effect on network performance. The Cisco IOS software features for congestion management, or queuing, include:

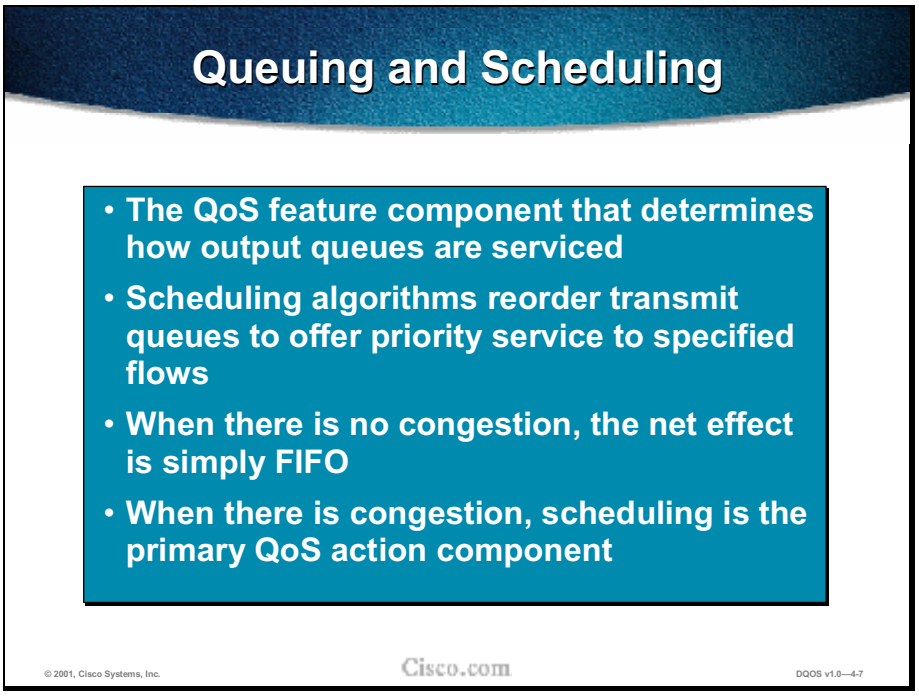
- FIFO—first-in, first-out
- PQ—priority queuing
- CQ—custom queuing
- WFQ—flow-based weighted fair queuing *
- CBWFQ—class-based WFQ *
- IP RTP Priority—also known as PQ/WFQ
- Frame Relay IP RTP Priority
- LLQ—low latency queuing

* You will also encounter flow-based DWFQ and class-based DWFQ. These are implementations of WFQ that run only in distributed mode on Versatile Interface Processors (VIPs) for the Cisco 7500 series platforms. Specific descriptive and configuration details are available in the references provided throughout this chapter. See especially *Congestion Management Overview* and *Configuring Weighted Fair Queuing* at the following URLs:

- http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcprt2/qcdconmg.htm

- http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcprt2/qcdwfq.htm


Queuing and Scheduling Techniques: Overview
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—4-6



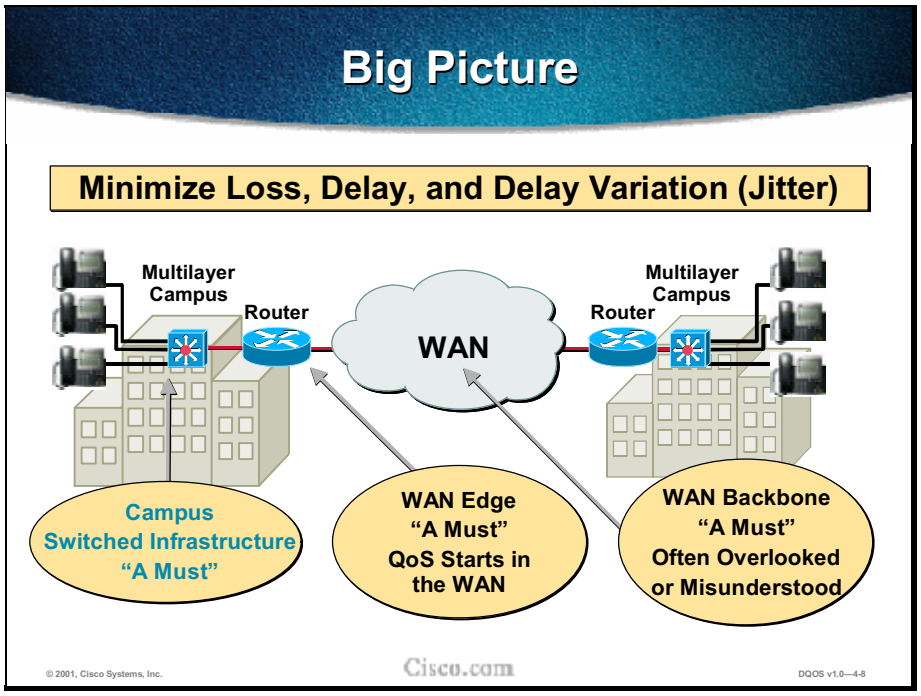
Heterogeneous networks include many different protocols used by applications, giving rise to the need to prioritize traffic in order to satisfy time-critical applications while still addressing the needs of less time-dependent applications, such as file transfer. Different types of traffic sharing a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

Congestion management features operate to control congestion once it occurs. One way that network elements handle an overflow of arriving traffic is to use a queuing algorithm to sort the traffic, then determine some method of prioritizing it onto an output link. Each queuing algorithm was designed to solve a specific network traffic problem and has a particular effect on network performance. Congestion management features allow control of congestion by determining the order in which packets are sent out of an interface, based on priorities assigned to those packets. Congestion management entails the creation of queues, assignment of packets to those queues based on the classification of the packet, and scheduling of the packets in a queue for transmission.

When there is no congestion, traffic is handled in a first-in, first-out (FIFO) manner. FIFO entails no concept of priority or classes of traffic. With FIFO, transmission of packets out the interface occurs in the order the packets arrive.

For a detailed discussion of congestion management, see *Congestion Management Overview* at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcprt2/qcdconmg.htm#xtocid182440



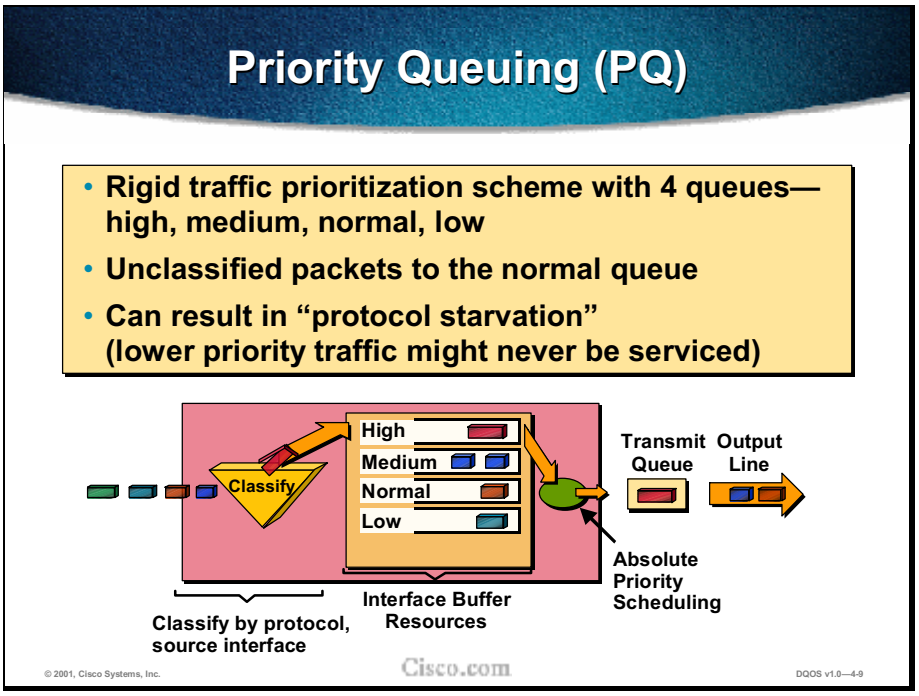
Everyone is challenged for bandwidth in the wide-area network. Most QoS starts in the WAN.

Within the campus, many problems can be fixed with additional bandwidth. Gigabit Ethernet links and 10 Gigabit Ethernet links abound.

Many campus networks today are over-provisioned and underutilized; therefore, congestion management tools are of little value in these networks. However, campus networks are subject to over-subscription (congestion) just as low-speed WAN links are. Eventually, some form of queuing will be needed in the campus as demand for bandwidth increases.

The 10/100 Ethernet to the desktop is switched. As new business client/server applications such as enterprise resource planning and supply-chain management are deployed on top of existing applications, network loads get heavier and heavier, forcing users to wait for files to download or screens to update. Users might mistakenly assume that a newly installed application is slow, when in reality a legacy network based on shared hubs is simply not up to the task of handling the extra load placed on it.

As shared devices, hubs share a fixed amount of bandwidth among connected users. Switches provide dedicated 10 Mbps or 100 Mbps bandwidth per port to individual users or servers. Less contention among users for bandwidth means fewer collisions, resulting in enhanced application performance, without costly wiring or network interface card changes.



With priority queuing (PQ), you configure four output queues: high, medium, normal, and low. When a packet is to be sent out an interface, the high priority queue is scanned first. When any packets from the high-priority queue have been sent, then the medium-priority queue is serviced. When all packets from the high or medium queue have been sent, the normal queue is serviced, and so on. The packet at the head of the highest queue is chosen for transmission. This procedure is repeated every time a packet is to be sent. Packets are classified based on user-specified criteria and placed into one of the four queues. Packets that are not classified by priority fall into the normal queue. When a queue is longer than the specified queue limit, all additional packets are dropped.

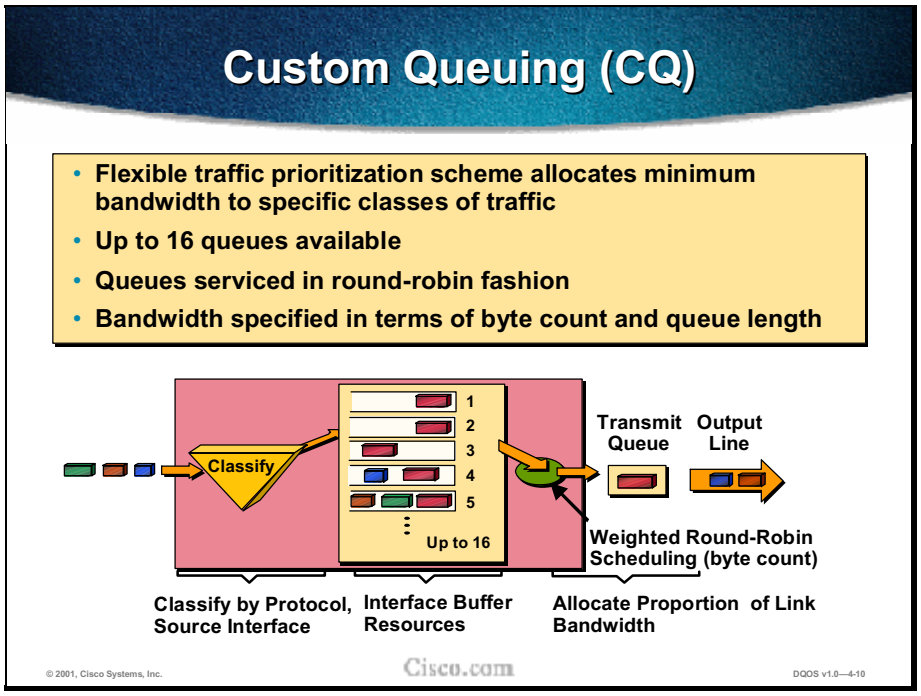
PQ provides absolute preferential treatment to high-priority traffic, ensuring that mission-critical traffic traversing various WAN links gets priority treatment. In addition, PQ provides a faster response time than other methods of queuing.

Although you can enable PQ for any interface, it is best used for low-bandwidth, serial interfaces that are subject to congestion. PQ introduces extra overhead that is acceptable for slow interfaces, but might not be acceptable for higher-speed interfaces such as Ethernet. With PQ enabled, the system takes longer to forward packets because the packets are classified by the system processor.

Because lower-priority traffic is often denied bandwidth in favor of higher-priority traffic, use of PQ could in the worst case result in lower-priority traffic never being sent. To avoid inflicting these conditions on lower-priority traffic, you can use traffic shaping, committed access rate (CAR), or class-based policing to rate-limit the higher-priority traffic.

To configure PQ, see *Configuring Priority Queuing* at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcprt2/qcdpq.htm



When CQ is enabled on an interface, you can specify, for up to 16 queues, the number of bytes the system should deliver before moving to the next queue. CQ cycles through the queues in round-robin fashion, sending the specified number of bytes for each queue before moving to the next queue. Packets are sent until the number of bytes sent exceeds the queue byte count or the queue is empty. If one queue is empty, the router sends packets from the next queue that has packets.

Bandwidth used by a particular queue is specified indirectly in terms of byte count and queue length. Simple formulas can be used to calculate byte counts so they give different bandwidth percentages to different packet streams; these are explained in the section “Determining Byte Count Values for Queues” in *Congestion Management Overview* at the following URL:

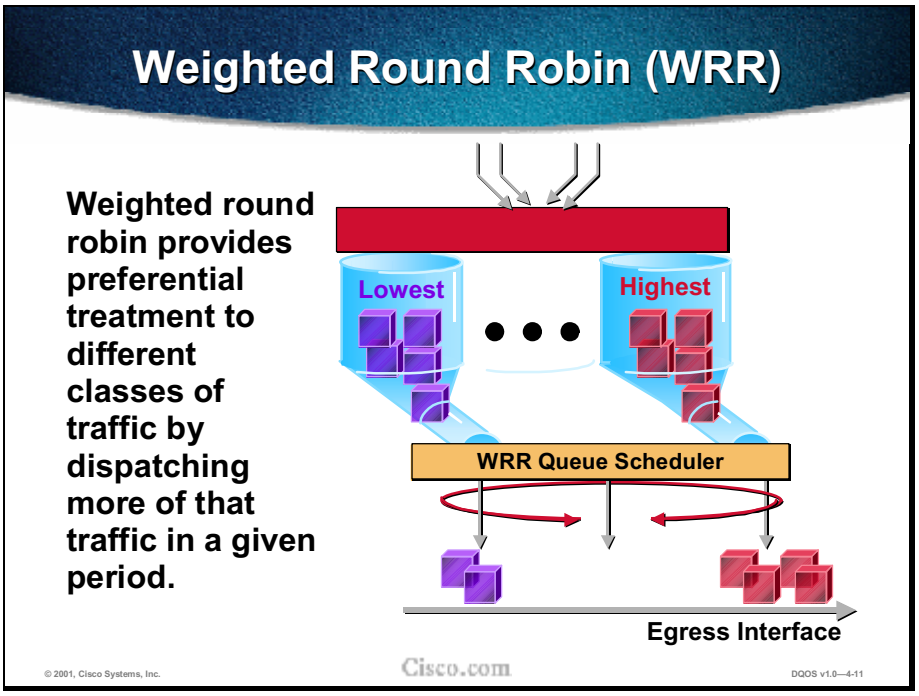
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcprt2/qcdconmg.htm#xtocid1824431

CQ ensures that no application or specified group of applications achieves more than a predetermined proportion of overall capacity when the line is under stress. Like PQ, CQ is statically configured and does not automatically adapt to changing network conditions. With CQ enabled, the system takes longer to forward packets than FIFO because the packets are classified by the system processor.

CQ can provide specific traffic with a fixed portion of available bandwidth at a potential congestion point, leaving the remaining bandwidth to other traffic. If a particular type of traffic is not using the bandwidth reserved for it, unused bandwidth can be dynamically allocated to other traffic types.

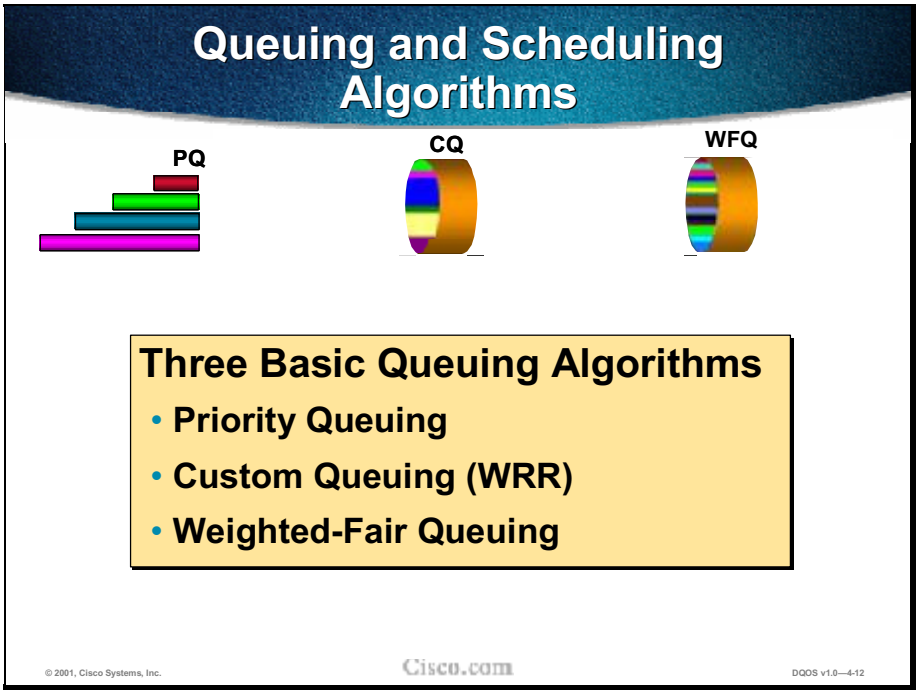
For specific instructions on configuring CQ, see *Configuring Custom Queuing* at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcprt2/qcdcq.htm



Weighted round robin is similar in operation to CQ, but is streamlined for operation in higher-end Layer 2 and Layer 3 switches. There are generally two or four queues, and classification is based on CoS or ToS/DSCP.

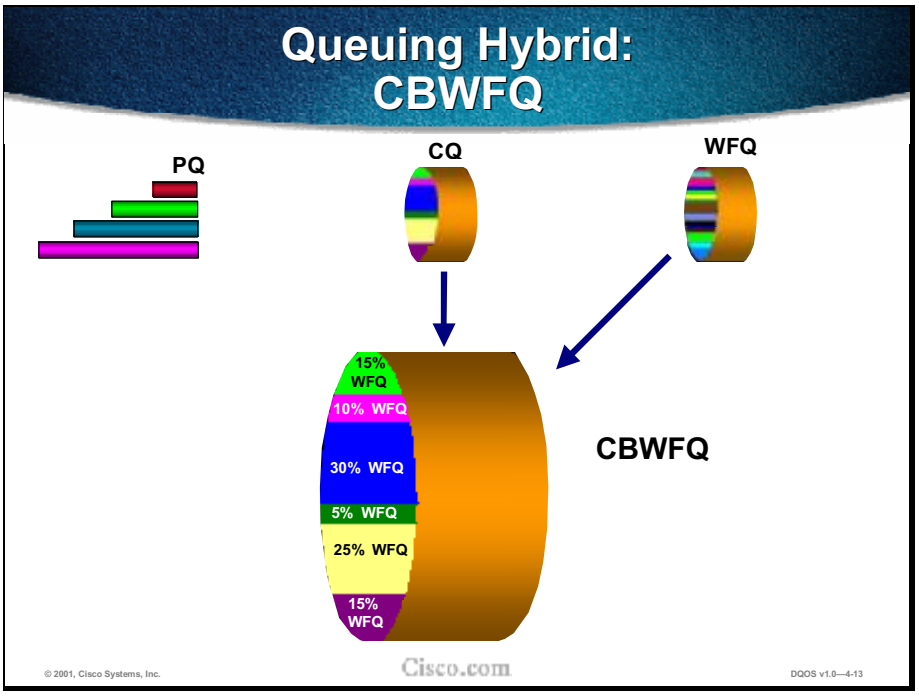
There are several variations on WRR. See the "QoS on Catalyst Switches" appendix, or refer to the latest platform-specific documentation on Cisco Connection Online (CCO) for more details.



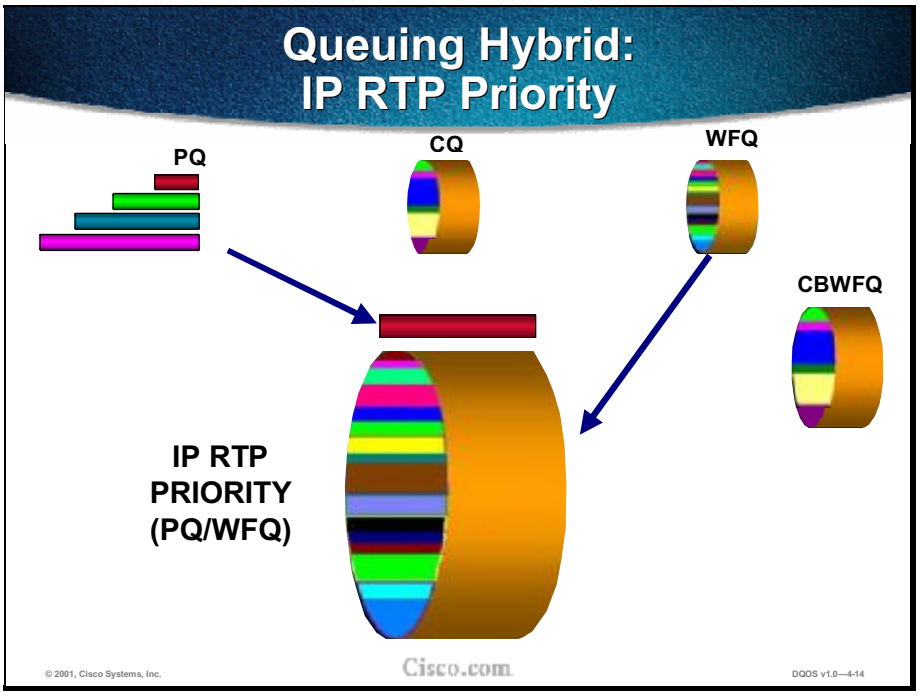
- Priority queuing (PQ)—This is the simplest queuing algorithm: Four (increasingly longer) queues are available for packet scheduling (high, medium, normal/default, and low). Only when *all* packets from the high queue have been serviced will the medium queue begin to be serviced. When the medium queue has been emptied, the high queue is checked for packets, and if there are packets in it, the process starts again from the top, working its way down to the low queue. PQ is an excellent mechanism for protecting important traffic at the expense of anything else; the major drawback of PQ is the very real probability of starving out traffic assigned to the lower queues.

- Custom queuing (CQ)—Up to 16 queues can be user-defined and have applications assigned to them; these queues are serviced in a round-robin fashion (preventing starvation of any given application/queue). Bandwidth can be guaranteed by setting the limits (byte-counts) for each queue. FIFO services each application/queue.

- Weighted fair queuing (WFQ)—This default algorithm for slow-speed links dynamically divides available bandwidth by a calculation based on the total number of flows and the weight (or ToS value) of each given flow. Bandwidth cannot be guaranteed, as the number of flows are constantly changing and thus so is the allocated bandwidth to each flow.

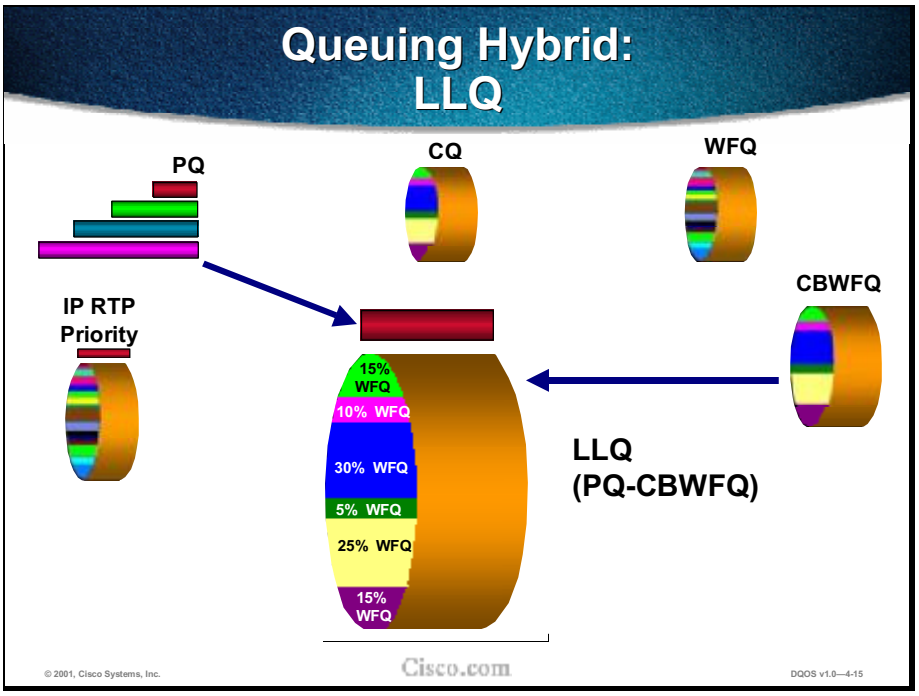


- Class-based WFQ (CBWFQ) extends the standard WFQ functionality to provide support for user-defined traffic classes. It allows you to specify the exact amount of bandwidth to be allocated for a specific class of traffic. Taking into account available bandwidth on the interface, you can configure up to 64 classes and control distribution among them.



- IP RTP Priority (also known as priority queuing/weighted fair queuing, or PQ/WFQ) provides a strict priority queuing scheme that allows delay-sensitive data such as voice to be dequeued and sent first—that is, before packets in other queues are dequeued. This feature can be used on serial interfaces and Frame Relay PVCs in conjunction with either WFQ or CBWFQ on the same outgoing interface. In either case, traffic matching the range of User Datagram Protocol (UDP) ports specified for the priority queue is guaranteed strict priority over other CBWFQ classes or WFQ flows; packets in the priority queue are always serviced first.

- Frame Relay IP RTP Priority—The Frame Relay IP RTP Priority feature provides a strict priority queuing scheme on a Frame Relay permanent virtual circuit (PVC) for delay-sensitive data such as voice. Voice traffic can be identified by its Real-Time Transport Protocol (RTP) port numbers and classified into a priority queue configured by the **frame-relay ip rtp priority** command. The result of using this feature is that voice is serviced as strict priority in preference to other nonvoice traffic.

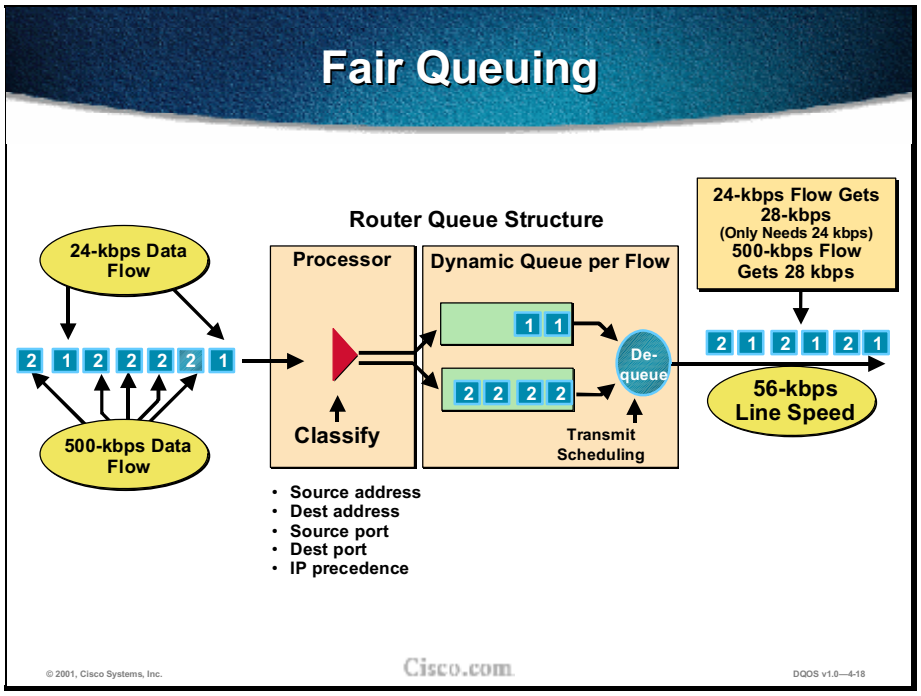


- LLQ (low latency queuing) provides strict priority queuing on ATM VCs and serial interfaces. This feature allows you to configure the priority status for a class within CBWFQ and is not limited to UDP port numbers, as is IP RTP Priority. LLQ and IP RTP Priority can be configured at the same time, but IP RTP Priority takes precedence.

Queuing Summary

	PQ	CQ	WRR	WFQ	CBWFQ	IP RTP Priority (PQ-WFQ)	LLQ (PQ-CBWFQ)
Classification	Protocol, interface	Protocol, interface	CoS, ToS/DSCP	IP Prec, RSVP, protocol, port	Mod CLI	VoFR and IP RTP Priority	VoFR and Mod CLI
# Queues	4	16	2 or 4	Per flow	64 classes	1 PQ + WFQ	1 PQ + CBWFQ
Scheduling	Strict priority	Round-robin	Round-robin	Fair: weight, arrival time	Fair: weight and BW	PQ: Strict WFQ: Fair	PQ: Strict CBWFQ: Fair/BW
Delay Guarantee	Yes	No	No	No	No	Yes	Yes
BW Guarantee	No	No	No	No	Yes	PQ: Yes WFQ: No	Yes
Used for Voice	No	No	Yes, Campus	Last resort	No	Yes	Yes

This table summarizes some key characteristics of available queuing techniques. WFQ, CBWFQ, PQ/WFQ, and LLQ are discussed in greater detail later in this chapter.



Weighted fair queuing (WFQ) classifies traffic into different flows, based on packet header information. WFQ uses a traffic data stream discrimination registry service to determine which traffic stream a message belongs to. For IP, the attributes of a message that are used to classify traffic into data streams are type of service (ToS), IP protocol, source IP address (if message is not fragmented), destination IP address (if message is not fragmented), source TCP/UDP port, and destination TCP/UDP port. For Frame Relay, the attribute used is data link connection identifier (DLCI). Attributes used for other protocols include, for example, protocol, source and destination MAC address, and session source and destination port and socket numbers.

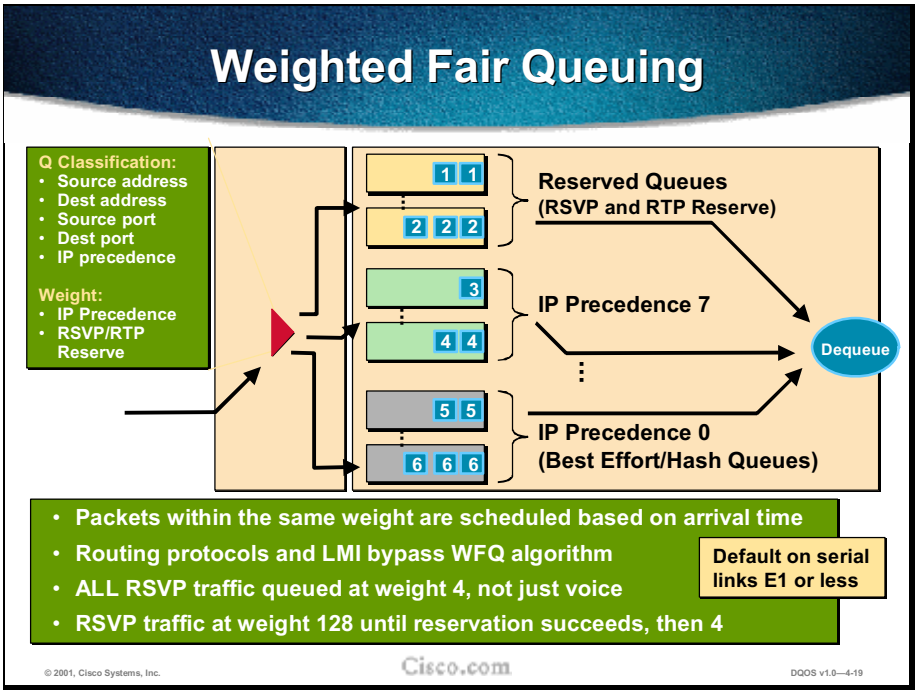
There are two flow categories: high-bandwidth sessions and low-bandwidth sessions. Low-bandwidth traffic has effective priority over high-bandwidth traffic, and high-bandwidth traffic shares the transmission service proportionally according to assigned weights. Low-bandwidth traffic streams, which make up the majority of traffic, receive preferential service, allowing their entire offered loads to be sent in a timely fashion. High-volume traffic streams share the remaining capacity proportionally among themselves.

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

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

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

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



WFQ is a flow-based queuing algorithm that simultaneously schedules interactive traffic to the front of the queue to reduce response time and fairly shares the remaining bandwidth between high-bandwidth flows. If you look at how WFQ works, you can see a problem with using it for voice. Packets are first classified by source and destination address, source and destination ports, and IP Precedence setting. They are “weighted” by the IP Precedence setting or by the RSVP/RTP Reserve flag.

The queues are served in the order of reserved queues first, then by IP Precedence field. As the precedence value increases, the algorithm allocates more bandwidth to that conversation to make sure that it gets served more quickly when congestion occurs. WFQ assigns a weight to each flow, which determines the transmit order for queued packets. In this scheme, lower weights are served first. IP Precedence serves as a divisor to this weighting factor. For instance, traffic with an IP Precedence value of 7 gets a lower weight than traffic with an IP Precedence field value of 3, and thus has priority in the transmit order.

For example, if you have one flow at each precedence level on an interface, each flow will get precedence+1 parts of the link, as follows:

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

The flows will get 8/36, 7/36, 6/36, and 5/36 of the link, and so on. However, if you have 18 precedence-1 flows and one of each of the others, the formula looks like this:

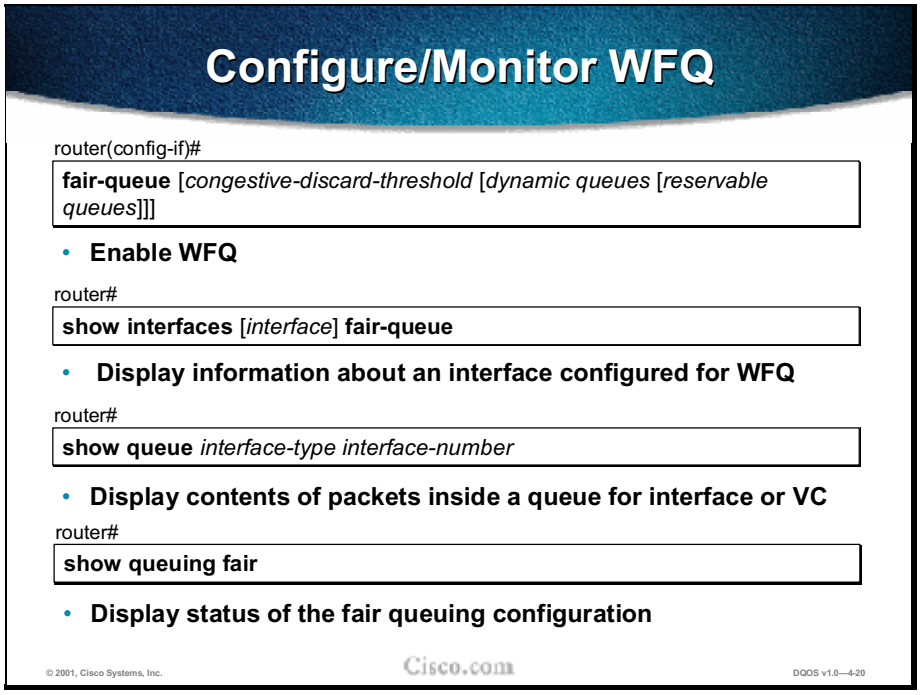
1+18x2+3+4+5+6+7+8 = 36-2+18x2 = 70

The flows will get 8/70, 7/70, 6/70, 5/70, 4/70, 3/70, 2/70, and 1/70 of the link, and 18 of the flows will get approximately 2/70 of the link.

The catch is that all flows are served, and under conditions where the number of queues becomes large, even the higher-priority traffic begins to experience unacceptable latency.

For a broader discussion of WFQ, see the section "Weighted Fair Queuing" in *Congestion Management Overview* at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcprt2/qcdconmg.htm#xtocid182444



The **fair-queue** interface command enables WFQ.

Three optional values can be configured. For each, if the value is not configured, the default value is used. The values are:

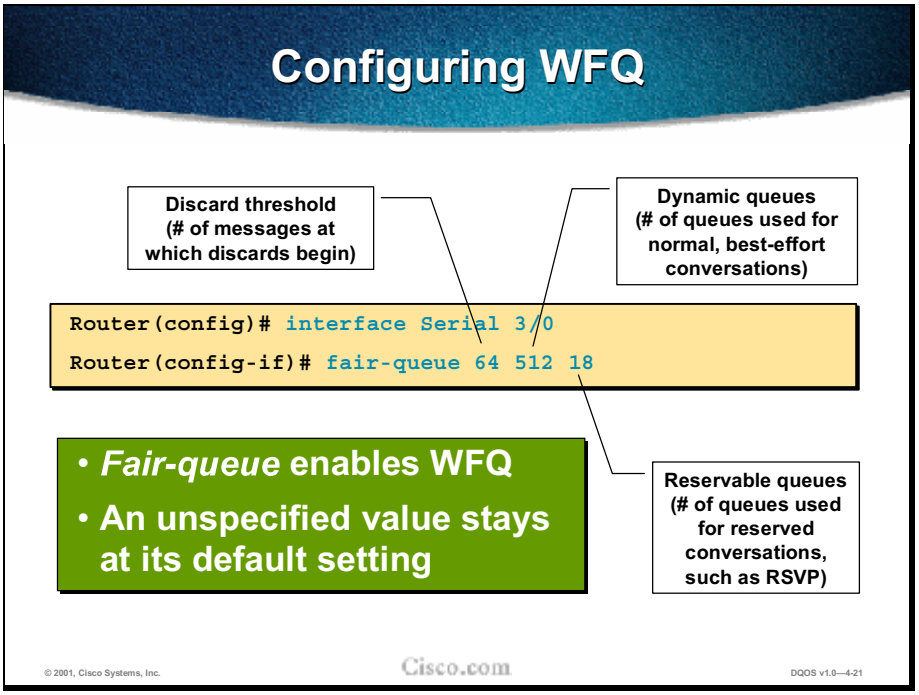
- *congestive-discard-threshold*: Number of messages allowed in each queue. The default is 64 messages, and a new threshold must be a power of 2 in the range 16 to 4096. When a conversation reaches this threshold, new message packets are discarded.

- *dynamic-queues*: Number of dynamic queues used for best-effort conversations (that is, a normal conversation not requiring any special network services). Values are 16, 32, 64, 128, 256, 512, 1024, 2048, and 4096. The default number of dynamic queues is determined

by bandwidth and ranges from 64 (up to and including 64 kbps) to 256 (greater than 512 kbps). (ATM defaults: 16 to 25, for 16 to 8000+ kbps, respectively.)

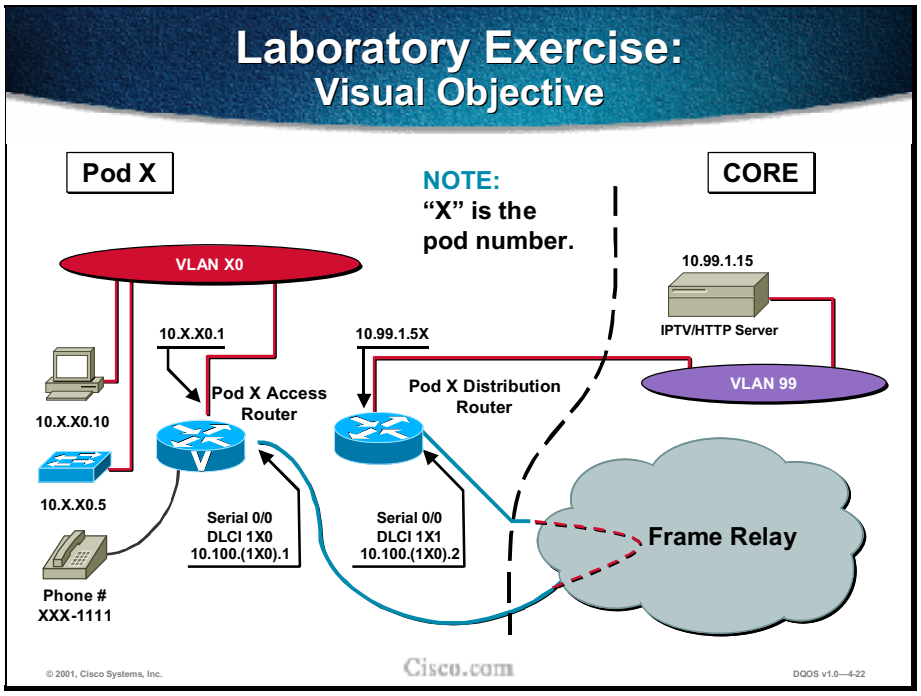
- *reservable-queues*: Number of reservable queues used for reserved conversations in the range 0 to 1000. The default is 0. Reservable queues are used for interfaces configured for features such as Resource Reservation Protocol (RSVP).
  - When RSVP is configured on an interface that supports fair queuing or on an interface that is configured for fair queuing with the reservable queues set to 0 (the default), the reservable queue size is automatically configured as interface bandwidth divided by 32 kbps. You can override this default by specifying a reservable queue other than 0.

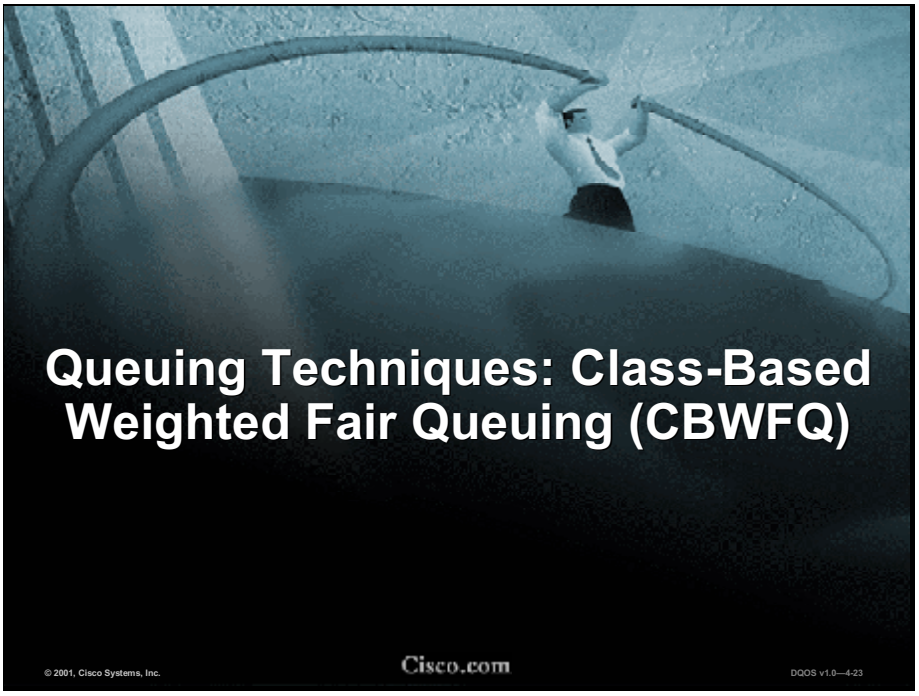
As seen in the figure, several **show** commands are available for monitoring WFQ.

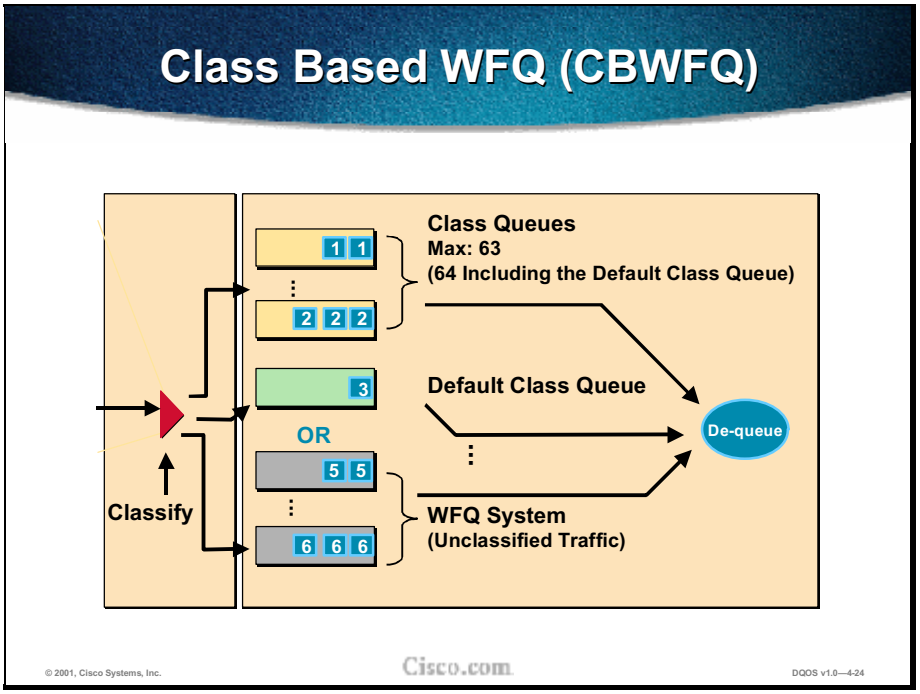


The example requests a fair queue with a congestive discard threshold of 64 messages, 512 dynamic queues, and 18 RSVP queues.

Unspecified parameters take the default values.


Laboratory Exercise:
Visual Objective
Pod X
CORE
NOTE:
"X" is the
pod number.
VLAN X0
10.99.1.15
IPTV/HTTP Server
10.X.X0.1
10.99.1.5X
Pod X Access
Router
Pod X Distribution
Router
VLAN 99
10.X.X0.10
10.X.X0.5
Serial 0/0
DLCI 1X0
10.100.(1X0).1
Serial 0/0
DLCI 1X1
10.100.(1X0).2
Frame Relay
Phone #
XXX-1111
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—4-22





With WFQ, all flows are served, and under conditions where the number of queues becomes large, even the higher-priority traffic begins to experience unacceptable latency. CBWFQ solves this problem by fixing bandwidth for voice queues and exhaustively queuing them ahead of remaining queues.

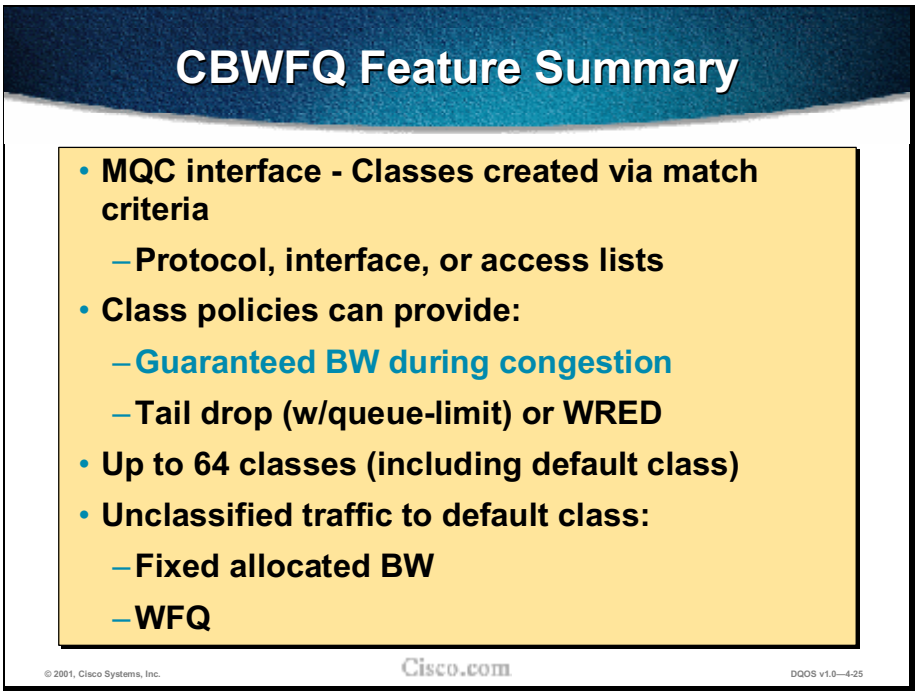
CBWFQ extends the standard WFQ functionality to provide support for user-defined traffic classes. For CBWFQ, you define traffic classes based on match criteria, including protocols, access control lists (ACLs), and input interfaces. Packets satisfying the match criteria for a class constitute the traffic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

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

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

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

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



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

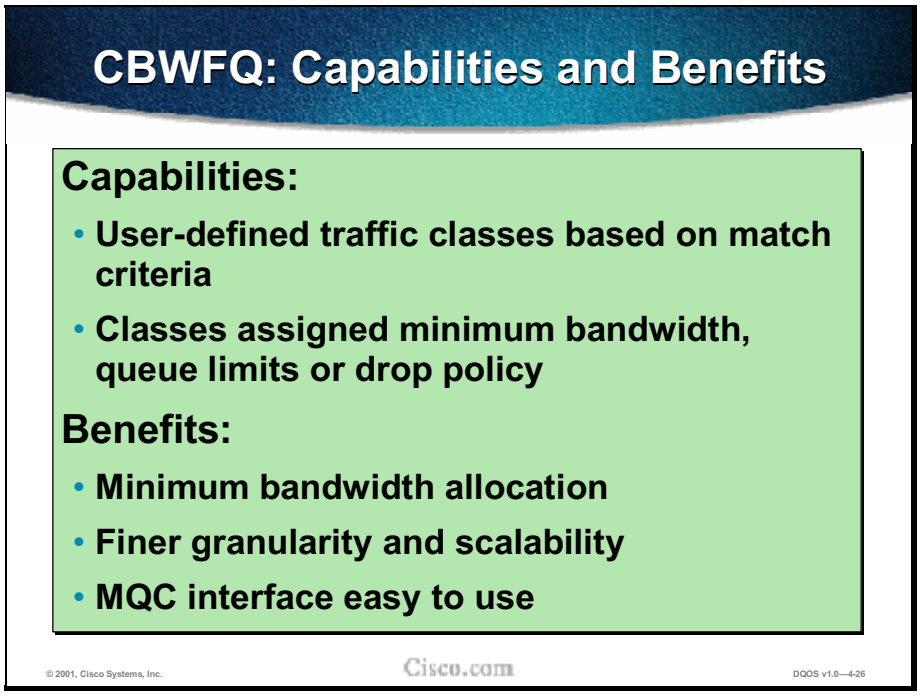
Flow classification is standard WFQ treatment. That is, packets with the same source IP address, destination IP address, source TCP or UDP port, or destination TCP or UDP port are classified as belonging to the same flow. WFQ allocates an equal share of bandwidth to each flow. Flow-based WFQ is also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

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

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

For more details refer to:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120t/120t5/cbwfq.htm



CBWFQ allows the user to define traffic classes based on custom-defined match criteria such as access control lists (ACLs), input interfaces, protocol, and QoS label. For example, a class might consist of a team working on a certain project, or a class can be created for the important mission-critical applications such as enterprise resource planning (ERP). When the traffic classes have been defined, they can be assigned a bandwidth, queue limit, or drop policy such as weighted random early detection (WRED).

- Bandwidth allocation—CBWFQ allows you to specify the exact amount of bandwidth to be allocated for a specific class of traffic. Accounting for available bandwidth on the interface, you can configure up to 64 classes.

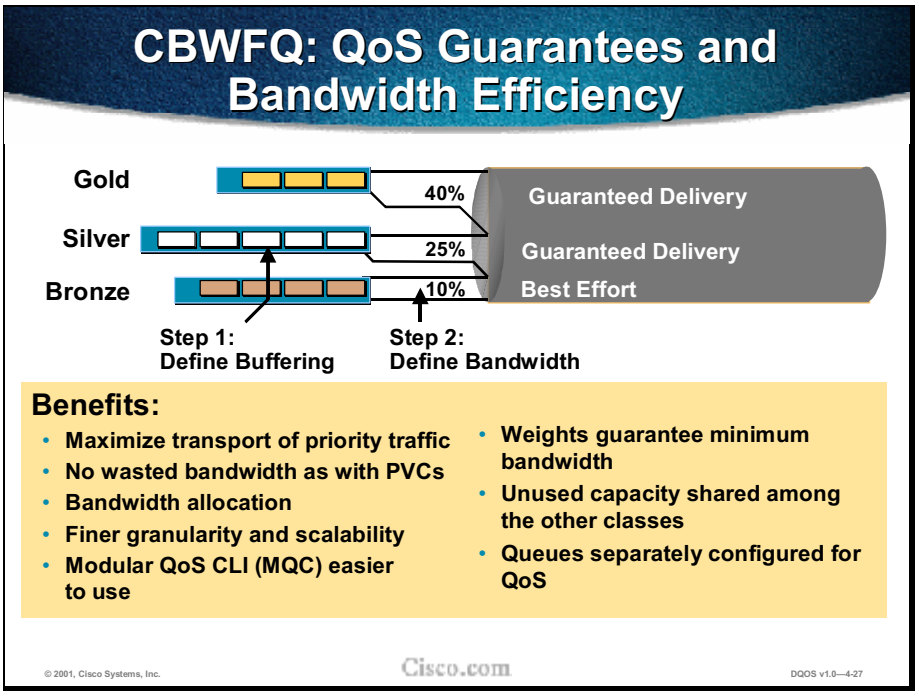
- Finer granularity and scalability—CBWFQ allows you total flexibility to define a class, based on ACLs and protocols or input interfaces, thereby providing finer granularity.

- Support in Modular QoS command-line interface (MQC).

- WRED supported as a drop policy—CBWFQ supports WRED as a drop policy per class, allowing you to provide differentiated service within a class.

This feature is supported on all platforms that WFQ is supported on; in other words, the Cisco 7200, 4700, 4500, 3600, and 2600 series, and so on.

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcprt2/qcdwfq.htm



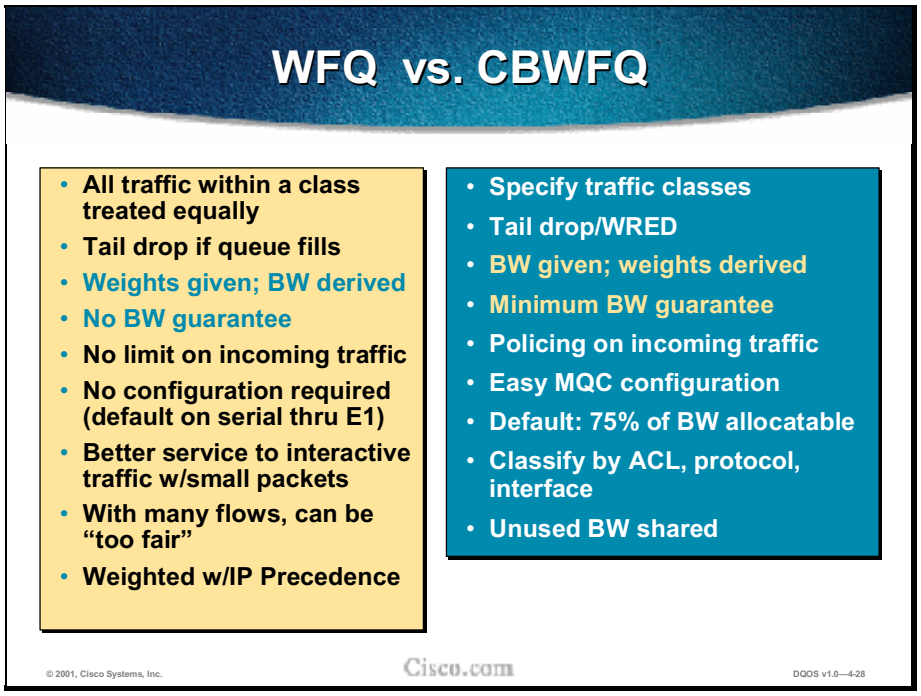
Let's take a closer look at CBWFQ and see how it can be used to guarantee service levels and maximize bandwidth utilization.

In this example, we've defined three service classes:

- Gold—with guaranteed delivery
- Silver—with guaranteed delivery
- Bronze—a best-effort service

Bandwidth is configured per class, not per connection.

By separately allocating bandwidth and buffering space, we can tailor each class to the specific service needs. For example, the gold class could be used for voice traffic: A large bandwidth allocation ensures that sufficient bandwidth is available for all the cells in the voice queue, while a moderately sized buffer limits the potential cell delay. Since these shares are relative weights, allocating a large share to gold means that a minimum is guaranteed; if the gold class is underutilized, the bandwidth will be shared by the remaining classes in proportion to their weights. This ensures maximum efficiency and ensures that priority customer traffic will be sent if bandwidth is available.

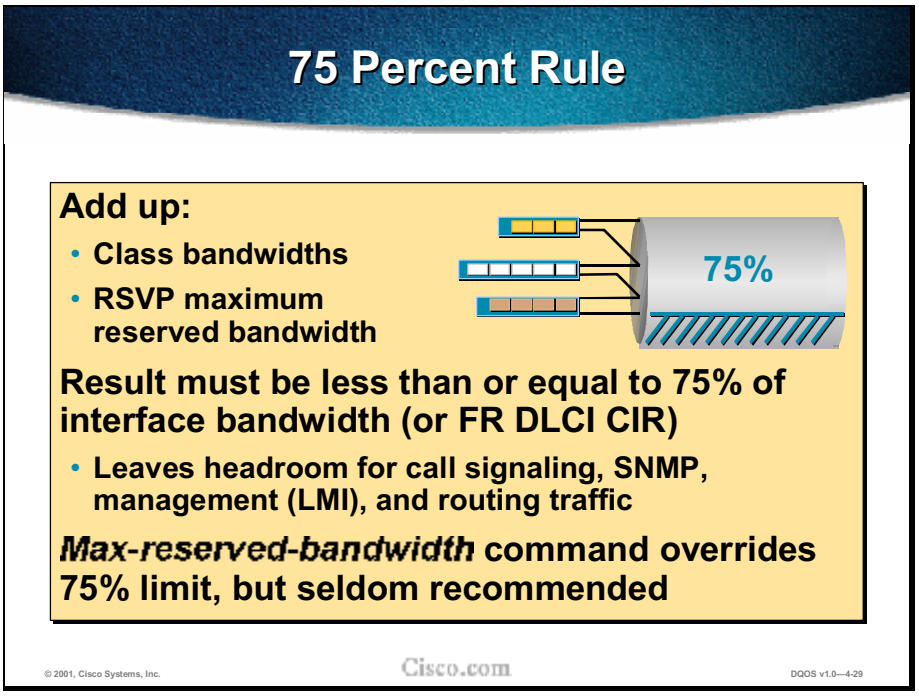


Here are some general factors to consider in determining whether you need to configure CBWFQ:

- Bandwidth allocation, CBWFQ—CBWFQ allows you to specify the exact amount of bandwidth to be allocated for a specific class of traffic. Taking into account available bandwidth on the interface, you can configure up to 64 classes and control distribution among them, which is not the case with flow-based WFQ.

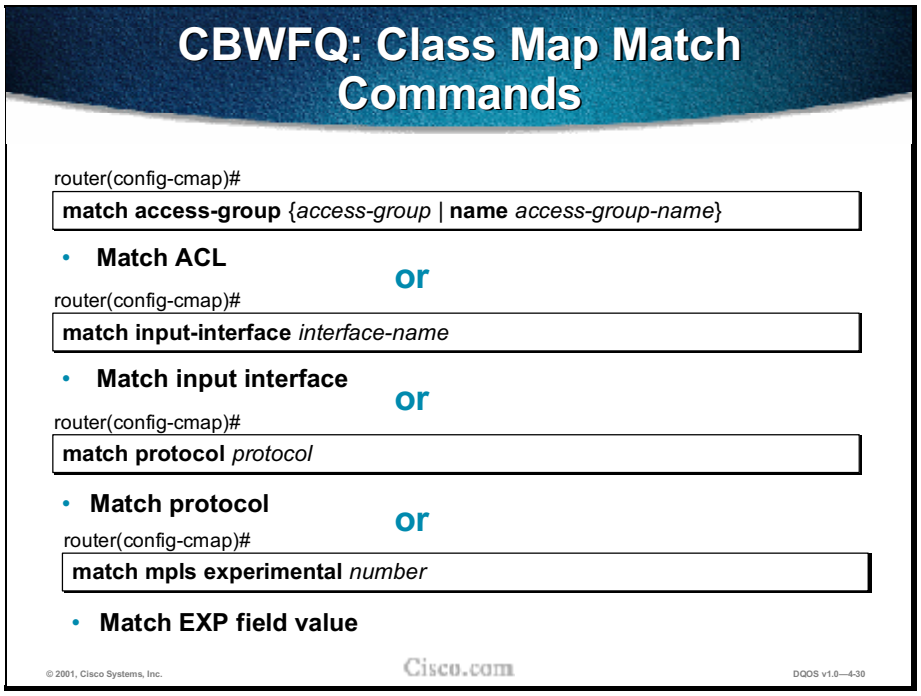
- Bandwidth allocation, WFQ—Flow-based WFQ applies weights to traffic to classify it into conversations and determine how much bandwidth each conversation is allowed relative to other conversations. Packets with the same source IP address, destination IP address, source TCP or User Datagram Protocol (UDP) port, destination TCP or UDP port, protocol, and ToS field belong to the same flow. (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. During periods of congestion, WFQ allocates an equal share of the bandwidth to each active queue. Flow-based WFQ is also called fair queuing because all flows are equally weighted and are allocated equal bandwidth.

- CBWFQ coarser granularity and scalability—CBWFQ allows you to define what constitutes a class based on criteria that exceed the confines of flow. CBWFQ allows you to use access control lists and protocols or input interface names to define how traffic will be classified, thereby providing coarser granularity. You need not maintain traffic classification on a flow basis. Moreover, you can configure up to 64 discrete classes in a service policy.



Properly provisioning the network bandwidth is a major component of successful network design. You can calculate the required bandwidth by adding the bandwidth requirements for each major application (for example, voice, video, and data). This sum then represents the minimum bandwidth requirement for any given link, and it should not exceed approximately 75 percent of the total available bandwidth for the link. This 75 percent rule assumes that some bandwidth is required for overhead traffic, such as routing and Layer 2 keepalives, as well as for additional applications such as e-mail and Hypertext Transfer Protocol (HTTP) traffic.

Thus, the total amount of bandwidth allocated for all classes included in a policy map should not exceed 75 percent of the available bandwidth on the interface. The **max-reserved bandwidth** command overrides the 75 percent limitation, but overriding is recommended only for the most knowledgeable network administrators who have access to precise figures for available, used, and required bandwidth. If not all the bandwidth is allocated, the remaining bandwidth is proportionally allocated among the classes, based on their configured bandwidth.



CBWFQ is configured using the Modular QoS command-line interface (MQC). The figure shows match commands used to create a class map in an enterprise environment. A class is created by specifying an ACL, an input interface, the protocol, or MPLS EXP value.

To configure CBWFQ, the following standard steps for creating class and policy maps and attaching policy maps to an interface are required:

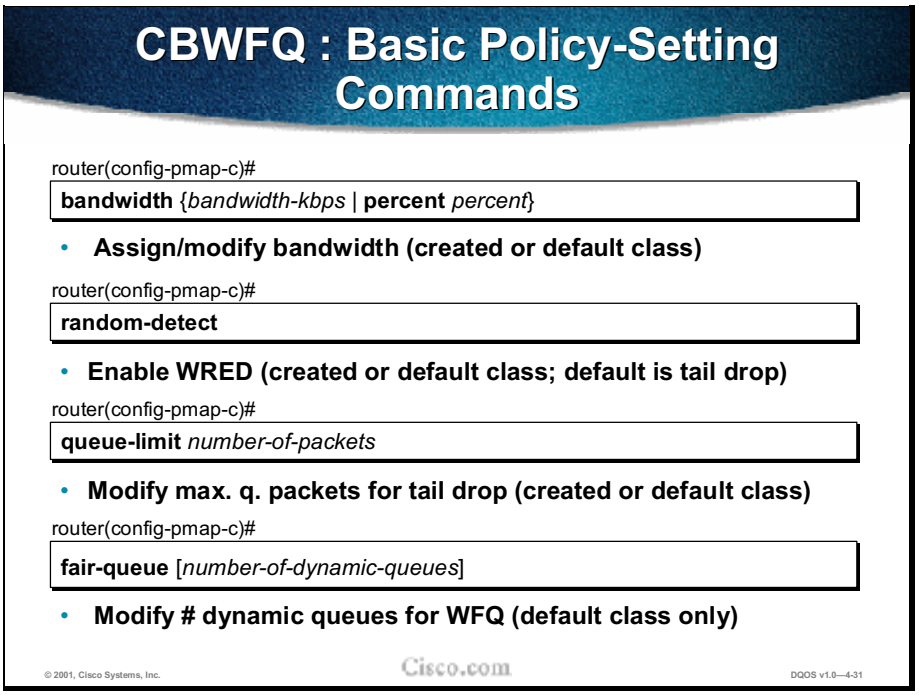
- Defining class maps
- Configuring class policy in the policy map
- Attaching the service policy and enabling CBWFQ

Commands are also available for the following optional steps:

- Modifying the bandwidth for an existing policy map class
- Modifying the queue limit for an existing policy map class
- Configuring the bandwidth limiting factor
- Deleting classes
- Deleting policy maps
- Verifying configuration of policy maps and their classes

For complete instructions on all available commands, see the section “Class-Based Weighted Fair Queuing Configuration Task List” in *Configuring Weighted Fair Queuing* at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcprt2/qcdwfq.htm#xtocid243909



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

- **class**: Name of a class to include in the service policy.
- **bandwidth**: (Created class or default class) Bandwidth, in kbps or percentage of available bandwidth, to be assigned to the class. Amount should accommodate Layer 2 overhead.

- **fair-queue**: (Default class only) Modifies number of default class dynamic (best-effort) queues to be reserved for use by flow-based WFQ. Values are 16, 32, 64, 128, 256, 512, 1024, 2048, and 4096. The default number of dynamic queues is determined by bandwidth and ranges from 64 (up to and including 64 kbps) to 256 (greater than 512 kbps). (ATM defaults are 16 to 25, for 16 to 8000+ kbps, respectively.)

- **queue-limit**: (Created class or default class) Modifies maximum number of packets that can be enqueued when the class is using tail drop. Default is 64.

or

- **random-detect**: (Created class or default class) Enables WRED. If enabled, the following are also available:

  - **random-detect exponential-weighting-constant** *exponent*—configures the exponential weight factor used in calculating the average queue length.

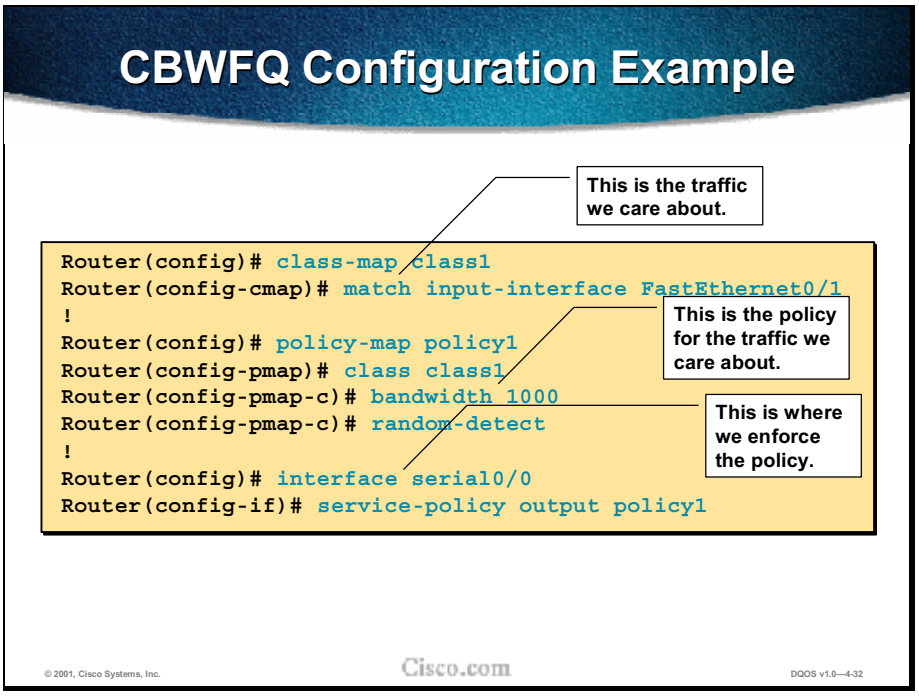
and/or

  - **random-detect precedence** *precedence min-threshold max-threshold mark-prob-denominator*—Configures WRED parameters for packets with a specific IP Precedence. Repeat this command for each precedence.

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

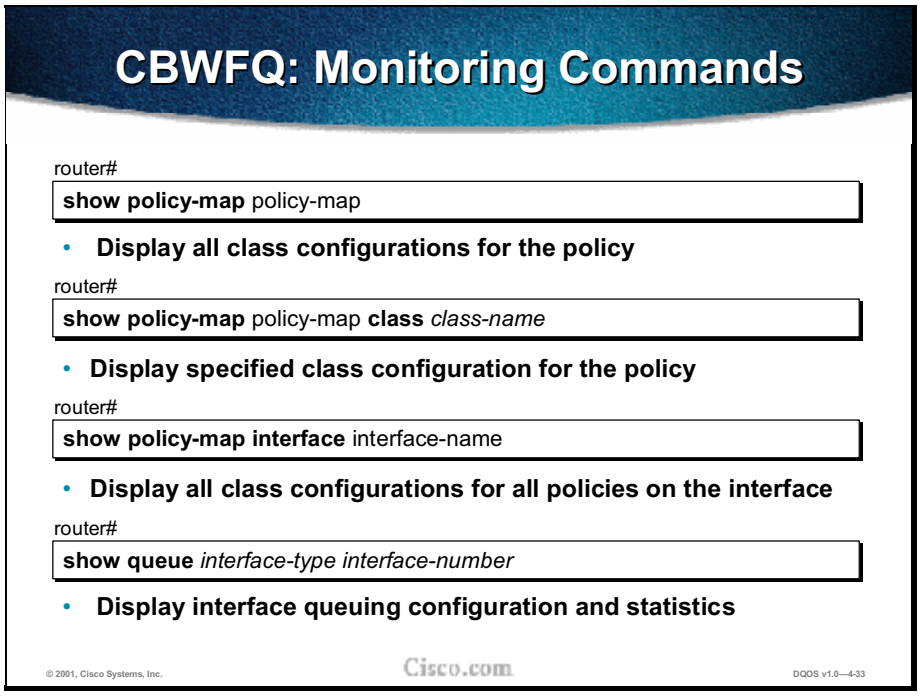
Traffic that does not satisfy the match criteria of other classes is directed to the default class.

You can define up 64 class policies, including the default class. However, as discussed earlier, total bandwidth for all classes should not exceed 75 percent of available bandwidth on the interface.



In the example, the class map *class1* is created and defined to use the input interface FastEthernet0/1 as a match criterion to determine if packets belong to the class. Next, the policy map *policy1* is defined to contain policy specification for *class1*. In the policy map, the **bandwidth** command specifies the bandwidth for traffic in that class, and the **random-detect** command specifies WRED packet drop.

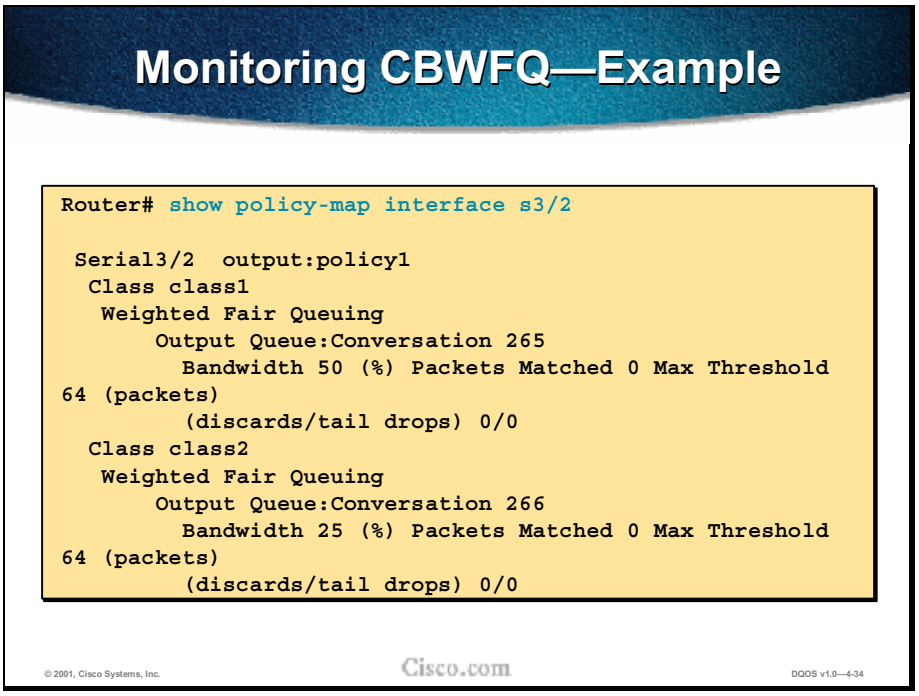
Note that CBWFQ is enabled only with the **service-policy output** command (not **service-policy input**).



As shown in the figure, several standard **show** commands are available in global configuration mode for verifying CBWFQ configurations.

The counters displayed after issuing the **show policy-map interface** command are updated only if congestion is present on the interface.

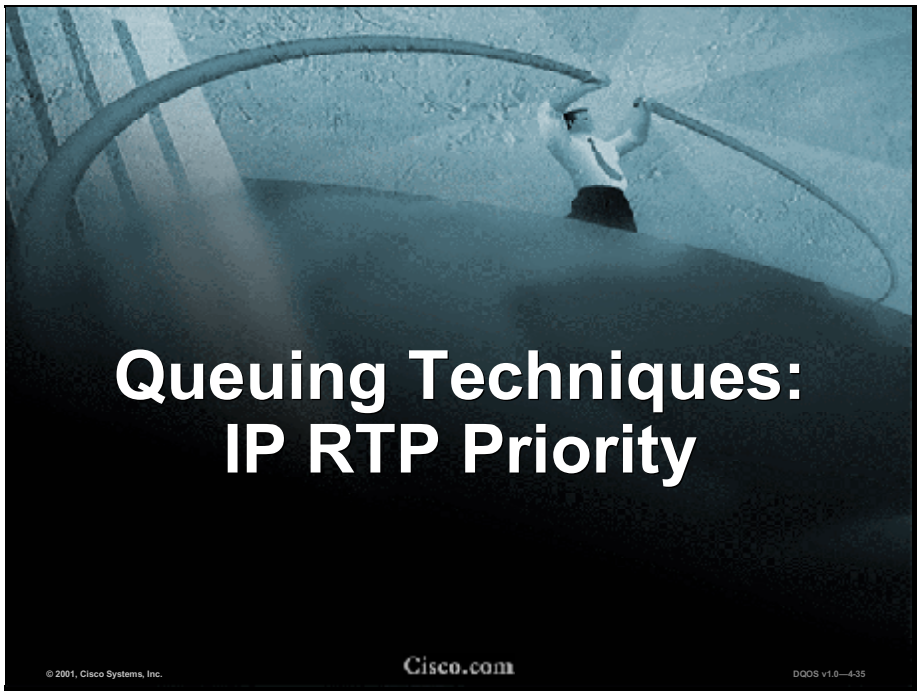
Any conversation ID that does not show up in the **show policy-map** output is a flow-based queue; **show policy-map** shows the class-based queues.

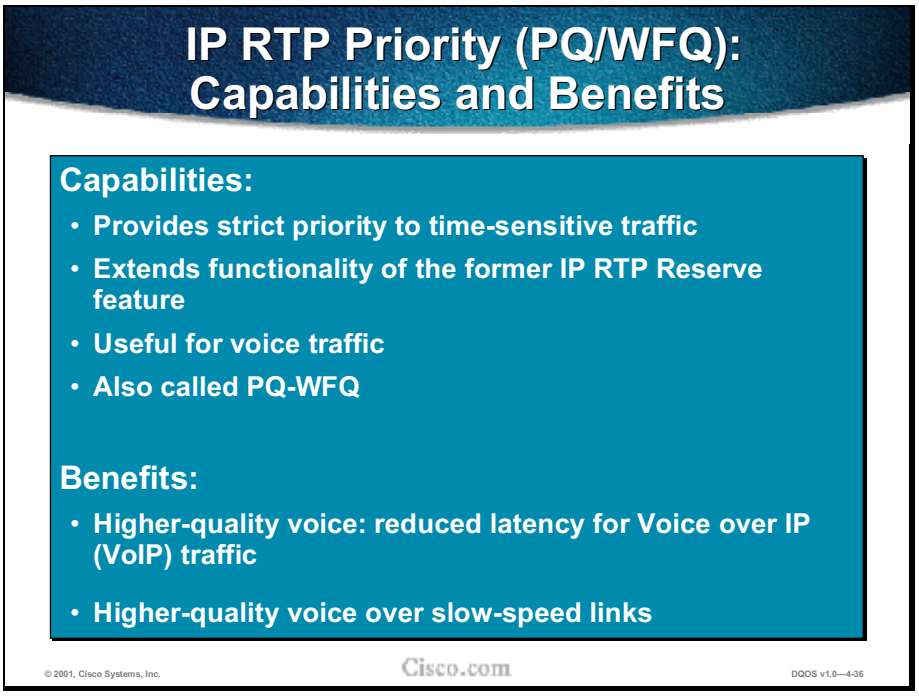


Bandwidth can be configured as a percentage rather than as a value in kbps. In the figure, the example output from the **show policy-map interface** command shows that 50 percent of the interface bandwidth is guaranteed for class1 and 25 percent is guaranteed for class2.

In this example, the entire interface bandwidth is available for CBWFQ because RSVP, IP RTP Priority, and LLQ are not enabled. If this policy map is attached to a physical interface, the available bandwidth is equal to the link bandwidth. During periods of congestion, 50 percent of the link bandwidth is guaranteed to class1 and 25 percent of the link bandwidth is guaranteed to class2. For example, if this policy map was attached to a 1 Mbps link, class1 would be guaranteed 500 kbps and class2 would be guaranteed 250 kbps during periods of congestion.

The counters displayed after issuing the **show policy-map interface** command are updated only if congestion is present on the interface.


Queuing Techniques:
IP RTP Priority
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—4-35



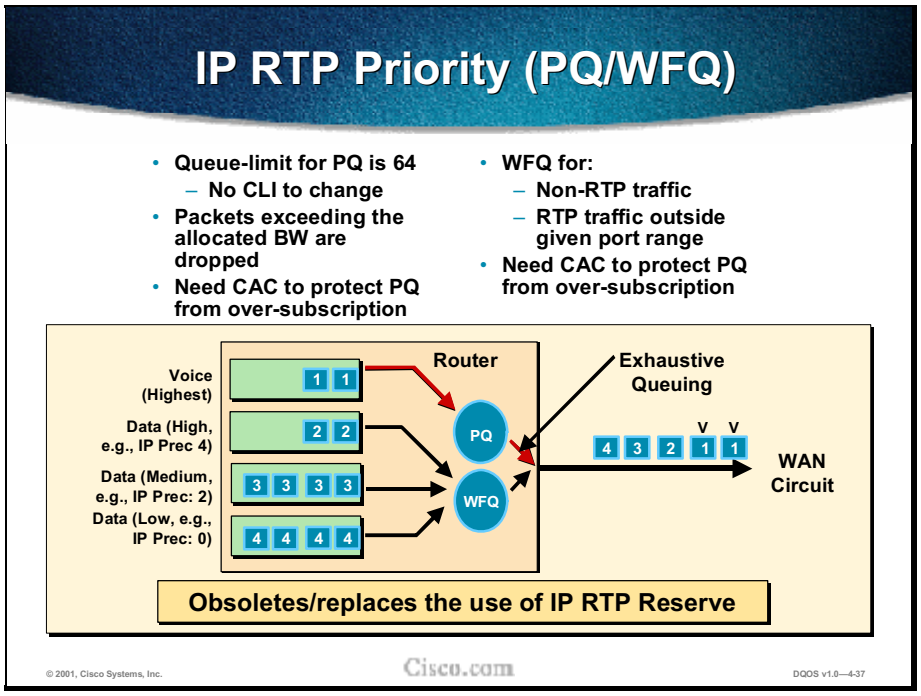
The IP RTP Priority (also know as priority queue/weighted fair queuing, or PQ/WFQ) feature provides a strict priority queuing scheme for delay-sensitive data such as voice. It is configured with the **ip rtp priority** command. The result is that voice or other high-priority traffic is serviced as strict priority in preference to other traffic.

IP RTP Priority replaces and should be used instead of the **ip rtp reserve** command. IP RTP Priority lets you specify a range of UDP/RTP ports whose traffic is guaranteed strict-priority service over any other queues or classes using the same output interface. Strict priority means that if packets exist in the priority queue, they are dequeued and sent first—that is, before packets in other queues are dequeued.

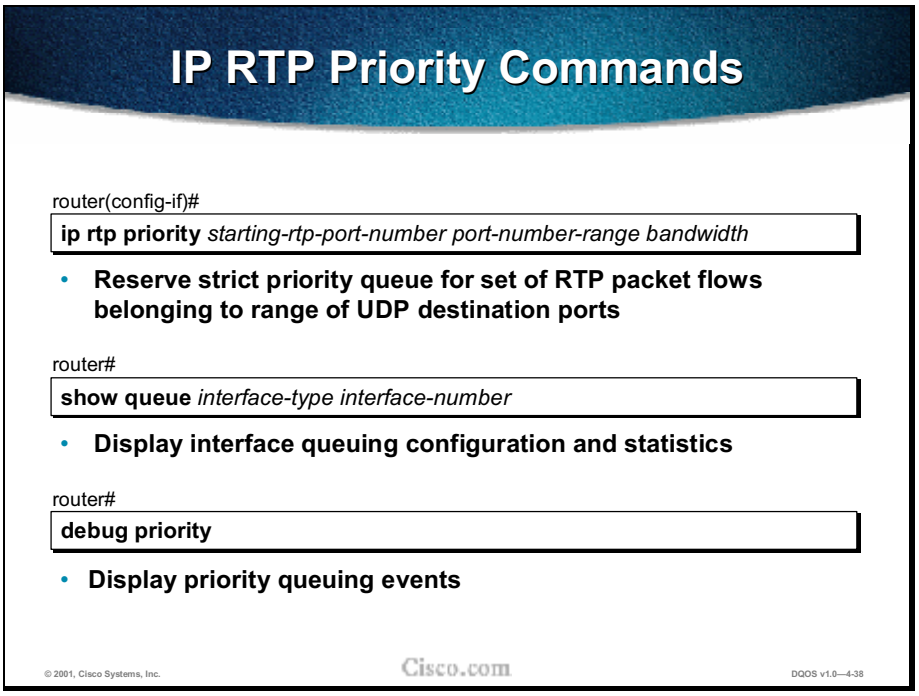
This feature can be used on the same outgoing interface with WFQ or CBWFQ. In either case, traffic matching the range of ports specified for the priority queue is guaranteed strict priority over other CBWFQ classes or WFQ flows; packets in the priority queue are always serviced first.

Because voice packets are small in size and the interface also can have large packets going out, the link fragmentation and interleaving (LFI) feature should be configured on lower-speed interfaces. When you enable LFI, the large data packets are broken up so that the small voice packets can be interleaved between the data fragments that make up a large data packet. LFI prevents a voice packet from needing to wait until a large packet is sent. Instead, the voice packet can be sent in a shorter amount of time.

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcprt2/qcdconmg.htm



IP RTP Priority creates a strict-priority queue for a set of RTP packet flows belonging to a range of UDP destination ports. While the actual ports used are dynamically negotiated between end devices or gateways, all Cisco VoIP products utilize the same UDP port range (16384 to 32767). Once the router recognizes the high-priority traffic, it places it into the strict-priority queue. When the priority queue is empty, the other queues are processed according to standard WFQ. IP RTP Priority does not become active until there is congestion on the interface.



Use the **ip rtp priority** interface command to reserve a strict-priority queue for a set of RTP packet flows belonging to a range of UDP destination ports. Configurable values are:

- *starting-rtp-port-number:* The starting RTP port number; the lowest port number to which the packets are sent
- *port-number-range*: The range of UDP destination ports; the number, which added to the starting-rtp-port-number argument, yields the highest UDP port number
- *bandwidth*: Maximum allowed bandwidth (in kbps)

Because the **ip rtp priority** command gives absolute priority over other traffic, it should be used with care. In the event of congestion, if the traffic exceeds the configured bandwidth, all excess traffic is dropped.

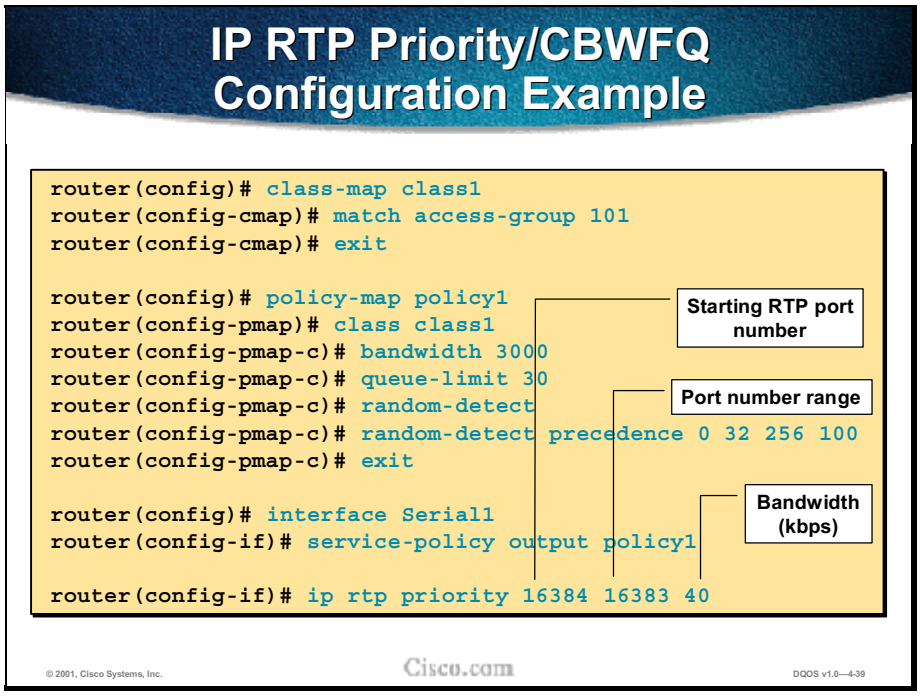
The **ip rtp reserve** and **ip rtp priority** commands cannot be configured on the same interface.

Use the **show queue** EXEC command to see the contents of the priority queue (such as queue depth and the first packet queued).

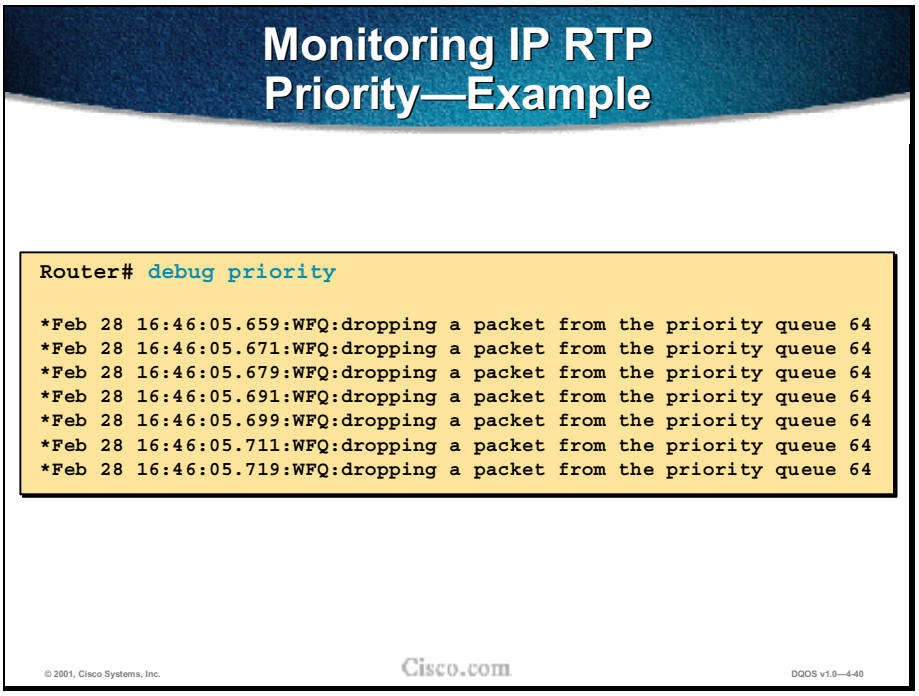
Use the **debug priority** and/or the **show queue** EXEC command to tune your RTP bandwidth or decrease RTP traffic if the priority queue is experiencing drops.

For additional details, see the section “IP RTP Priority Configuration Task List” in *Configuring Weighted Fair Queuing* at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcprt2/qcdwfq.htm#xtocid2439022



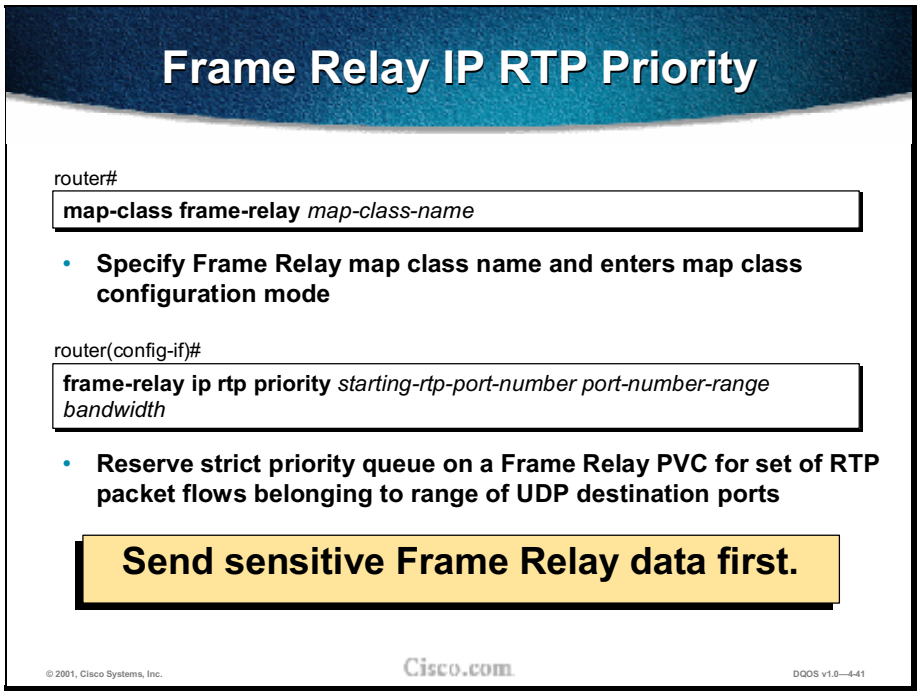
The example first defines a CBWFQ configuration, then reserves a strict-priority queue with the following values: a starting RTP port number of 16384, a range of 16383 UDP ports, and a maximum bandwidth of 40 kbps.



The example shows a sample output from the **debug priority** command.

Use the **debug priority** and/or the **show queue** EXEC command to tune your RTP bandwidth or decrease RTP traffic if the priority queue is experiencing drops.

Use the **show queue** EXEC command to see the contents of the priority queue (such as queue depth and the first packet queued).



The Frame Relay IP RTP Priority feature provides a strict-priority queuing scheme on a Frame Relay permanent virtual circuit (PVC) for delay-sensitive data such as voice. Voice traffic can be identified by its Real-Time Transport Protocol (RTP) port numbers and classified into a priority queue configured by the **frame-relay ip rtp priority** command. The result of using this feature is that voice is serviced as strict priority in preference to other nonvoice traffic.

This feature extends the functionality offered by the **ip rtp priority** command by supporting Frame Relay PVCs. This feature allows you to specify a range of User Datagram Protocol (UDP) ports whose voice traffic is guaranteed strict-priority service over any other queues or classes using the same output interface. Strict priority means that if packets exist in the priority queue, they are dequeued and sent first—that is, before packets in other queues are dequeued.

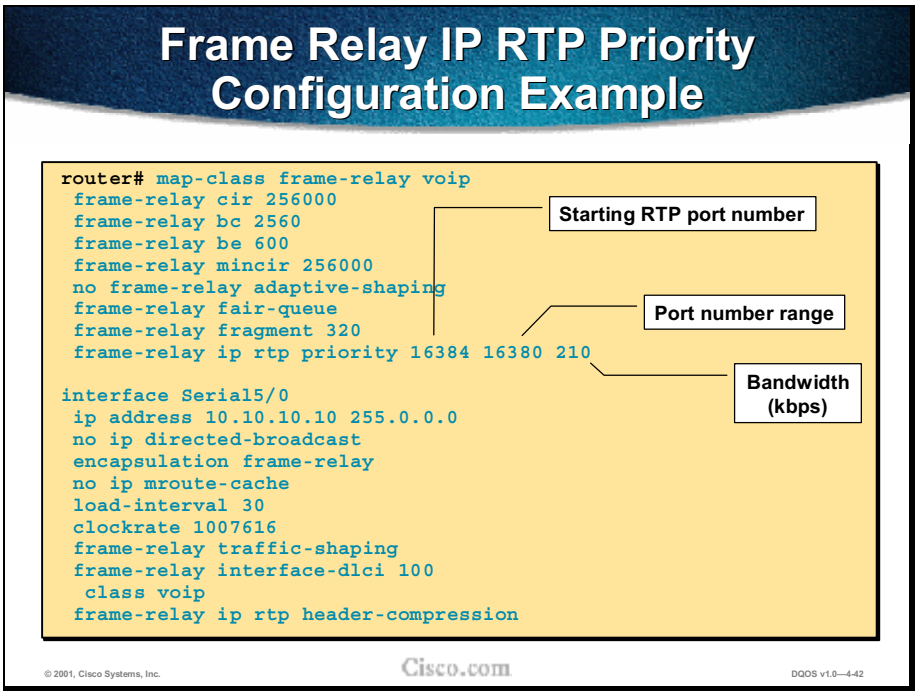
The strict-priority queuing scheme allows delay-sensitive data such as voice to be dequeued and sent first. Delay-sensitive data is given preferential treatment over other traffic. This process is performed on a per-PVC basis, rather than at the interface level.

Because the **frame-relay ip rtp priority** command gives absolute priority over other traffic, it should be used with care. In the event of congestion, if the traffic exceeds the configured bandwidth, all excess traffic is dropped. Therefore, some form of call admission control (CAC) must be used to prevent over-subscription of the PQ.

Notice in the figure that configuring QoS policies for Frame Relay differs from configuring QoS policies using MQC.

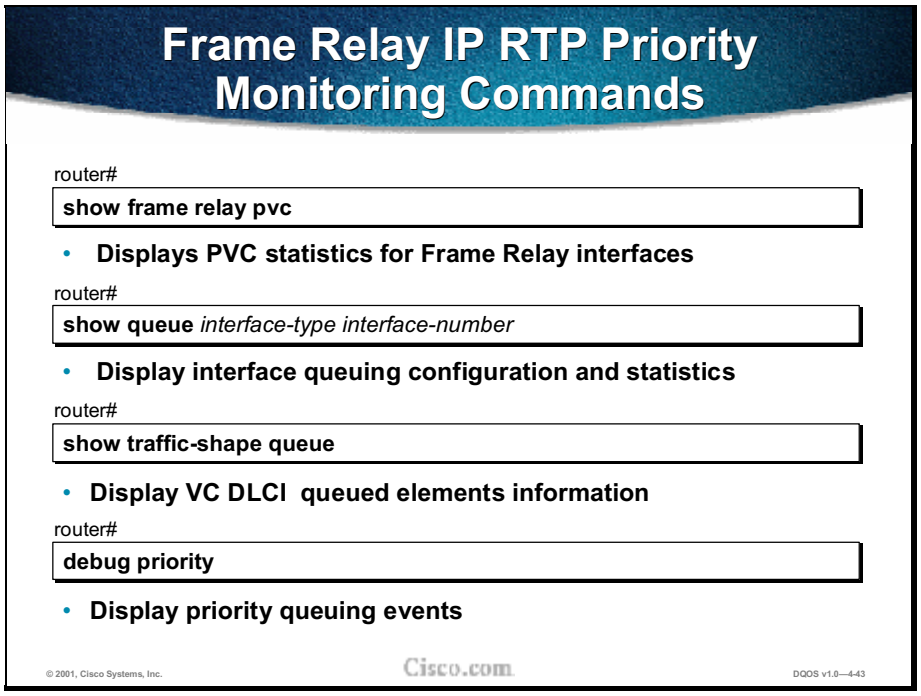
For additional details on configuring Frame Relay and Frame Relay map classes, see *Configuring Frame Relay* at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/wan_c/wcdfrely.htm

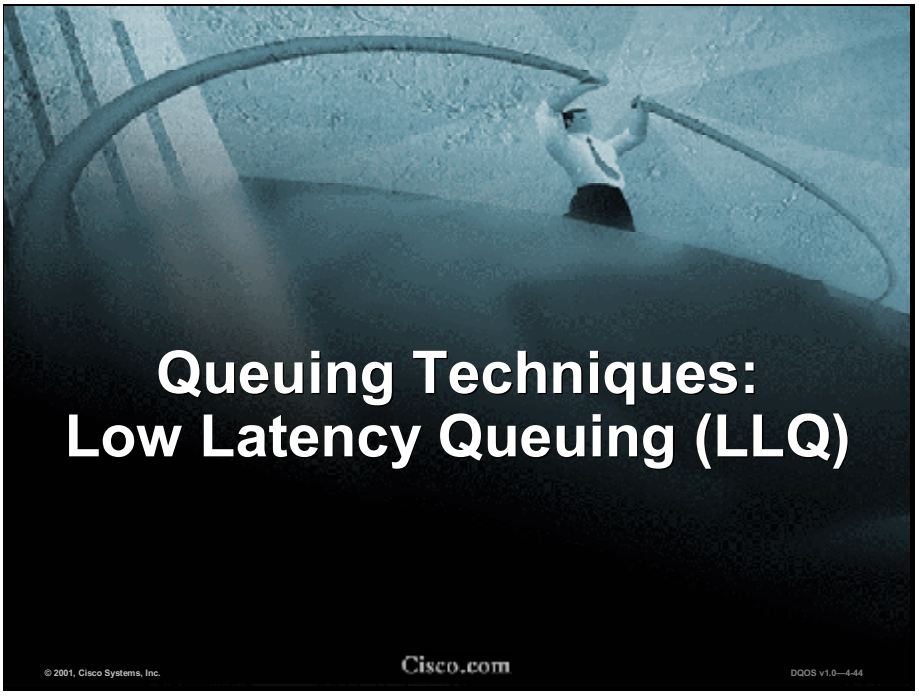


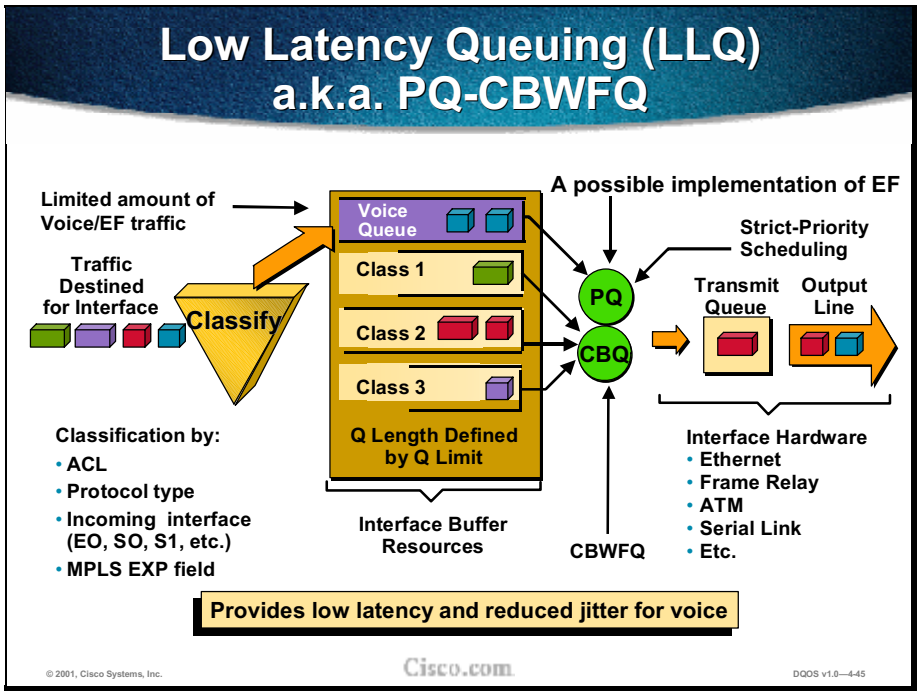
The example first configures the Frame Relay map class called *voip,* then applies the map class to PVC 100 to provide strict-priority service to matching RTP packets.

In the example, RTP packets on PVC 100 with UDP ports in the range 16384 to 32764 will be matched and given strict-priority service.



The figure shows the commands available for verifying and monitoring the Frame Relay IP RTP configuration.





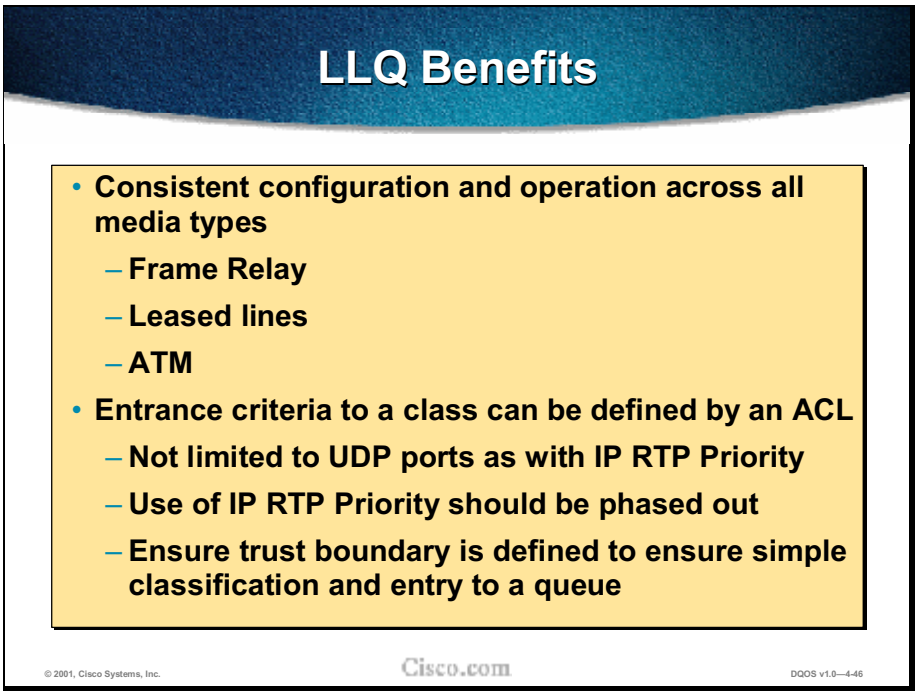
Without LLQ, CBWFQ provides WFQ based on defined classes, with no priority queue available for real-time traffic. CBWFQ allows you to define traffic classes and assign characteristics to that class. For example, the minimum bandwidth delivered to the class during congestion can be designated.

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

DiffServ expedited forwarding (EF) can be implemented using priority queuing (PQ), along with rate-limiting on the class (or BA). When implemented in a DiffServ network, EF PHB provides a virtual leased line, or premium service.

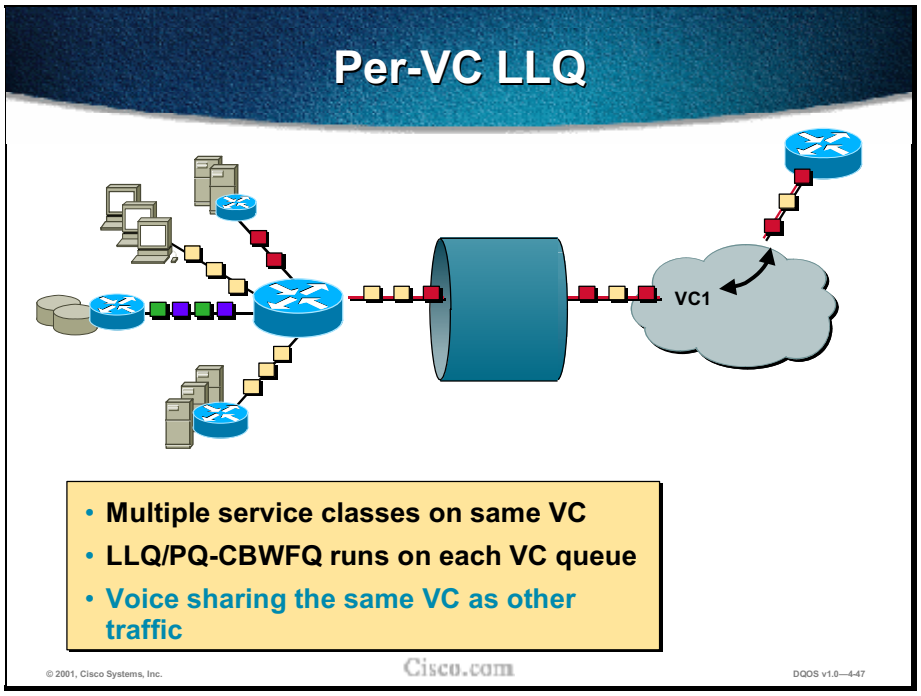
For optimal efficiency, however, EF PHB should be used only for the most critical applications because, in instances of traffic congestion, it is not feasible to treat all or most traffic as high priority.

EF PHB is ideally suited for applications such as VoIP that require low bandwidth, guaranteed bandwidth, low delay, and low jitter.



One benefit is having a consistent configuration across all media types, irrespective of the media used. Previously you had to configure MLPPP in one case, FRF.12 in another. Now LLQ provides a consistent queuing mechanism.

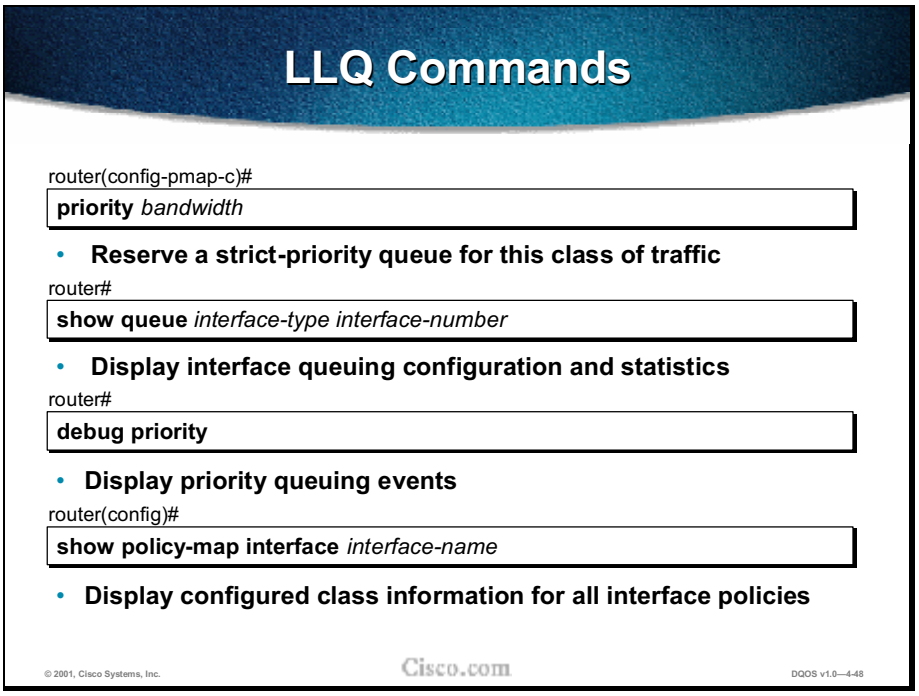
Also, with LLQ the entrance criterion to a class can be as granular as you like, because you define it by an ACL. You're not limited, as with IP RTP Priority, to a simple UDP port range. If the port range feature didn't get changed, it is probable that future voice applications would take advantage of it, knowing that when they hit the Cisco infrastructure, they would get preferential treatment.



LLQ can be used with ATM. The figure illustrates a single VC running voice and data, multiplexing both IP flows onto a single ATM VC.

At present, the bandwidth configured via CBWFQ is theoretical, meaning that it does not take the ATM cell tax into account.

On an ATM network, you can use a VBR VC, which gives a form of strict priority, or you can use a strict-priority class for voice traffic. For strict-priority classes, you do not assign bandwidth to the class, as you would otherwise. To characterize a class, you specify the queue limit for that class, which is the maximum number of packets allowed to accumulate in its queue. Packets belonging to a class are subject to the queue limits that characterize the class and to the bandwidth assigned when you enable strict priority for the class.



Use the **priority** command in policy-map class configuration mode to give priority to a class within a policy map. This reserves a strict-priority queue for this class of traffic. The bandwidth specified is the guaranteed allowed bandwidth (in kbps) for the priority traffic. Beyond the guaranteed bandwidth, the priority traffic will be dropped in the event of congestion, to ensure that nonpriority traffic is not starved.

The maximum reserved allocated bandwidth (default 75%) can be changed with the **max-reserved-bandwidth** command, but changing the default setting is recommended only for the most knowledgeable network administrators who have access to precise figures for available, used, and required bandwidth.

The following commands are available for verifying, monitoring, and maintaining the LLQ configuration:

To tune your RTP bandwidth or decrease RTP traffic if the priority queue is experiencing drops, use one or more of the following commands in EXEC mode:

- **show queue** *interface-type interface-number*: Display queuing configuration and statistics for a particular interface (EXEC mode command).
- **debug priority**: Display priority queuing events (EXEC mode command).
- **show queue** *interface-type interface-number*: Display queuing configuration and statistics for a particular interface (EXEC mode command).
- **show policy-map interface** *interface-name*: Display the configuration of all classes configured for all service policies on the specified interface. Display if packets and bytes were discarded or dropped for the priority class in the service policy attached to the interface (configuration mode command).

Remember the following guidelines when using the **priority** command:

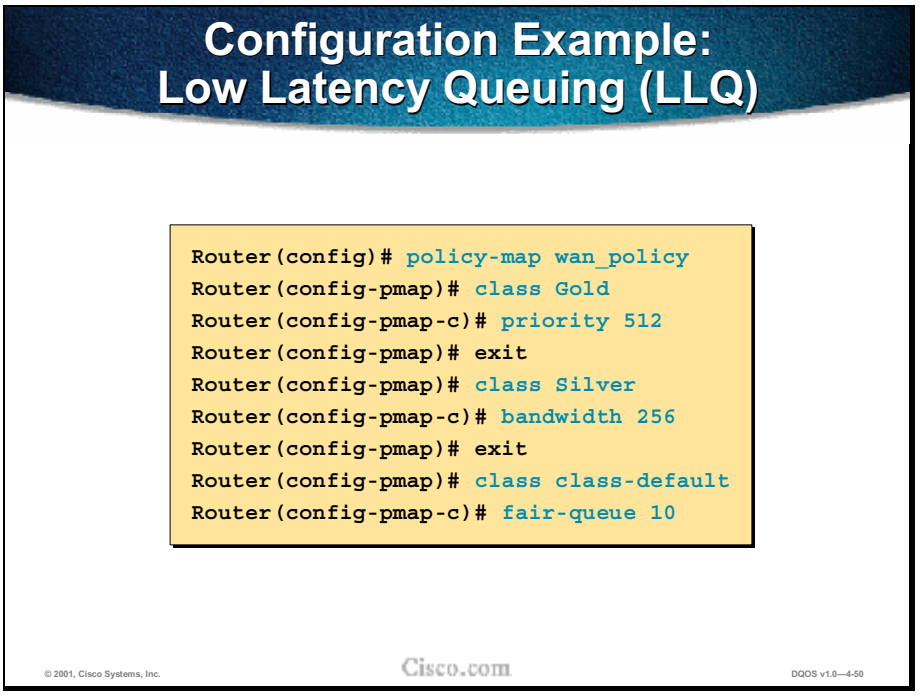
- Layer 2 encapsulations are accounted for in the amount of bandwidth specified with the **priority** command. However, the amount of bandwidth does not include other headers such as ATM cell tax overheads. You must allow bandwidth for possible jitter introduced by the routers in the voice path.

- The **priority** command can be used for Voice over IP (VoIP) on serial links and ATM PVCs. The **priority** command does not support VoIP over Frame Relay links.

- The **random-detect**, **queue-limit**, and **bandwidth** commands cannot be used while the **priority** command is configured.

- The **priority** command can be configured in multiple classes, but it should be used only for voice-like, constant bit rate (CBR) traffic.

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_r/qrdcmd3.htm



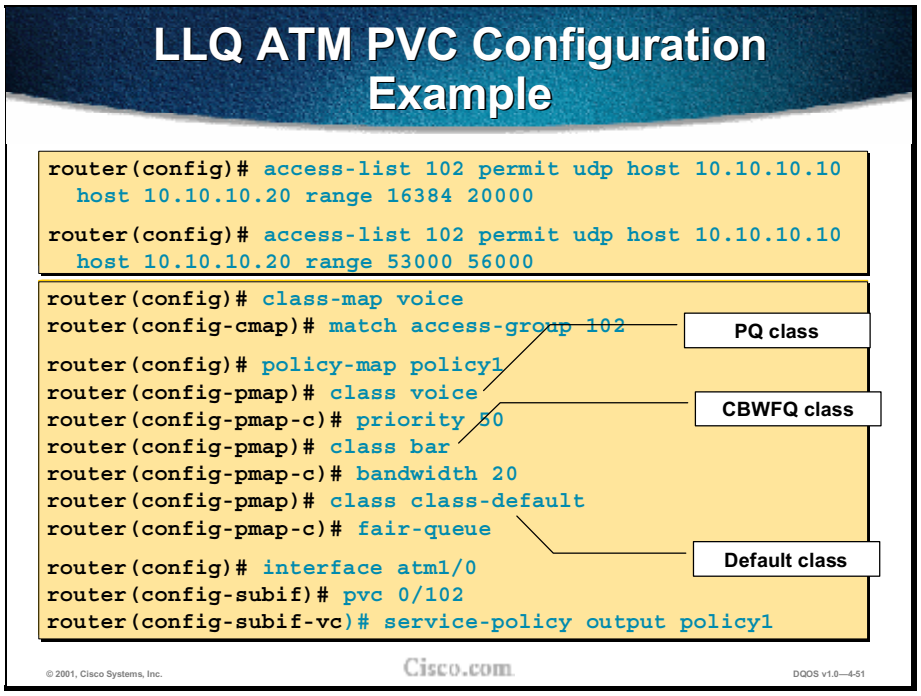
This example creates a policy map for two existing traffic classes, gold and silver.

Gold traffic is configured for LLQ, with a single priority queue and guaranteed bandwidth of 512 kbps.

Silver traffic is configured for CBWFQ (by use of the **bandwidth** command in the policy map configuration) with guaranteed bandwidth of 256 kbps, since neither tail drop nor WRED congestion avoidance apply.

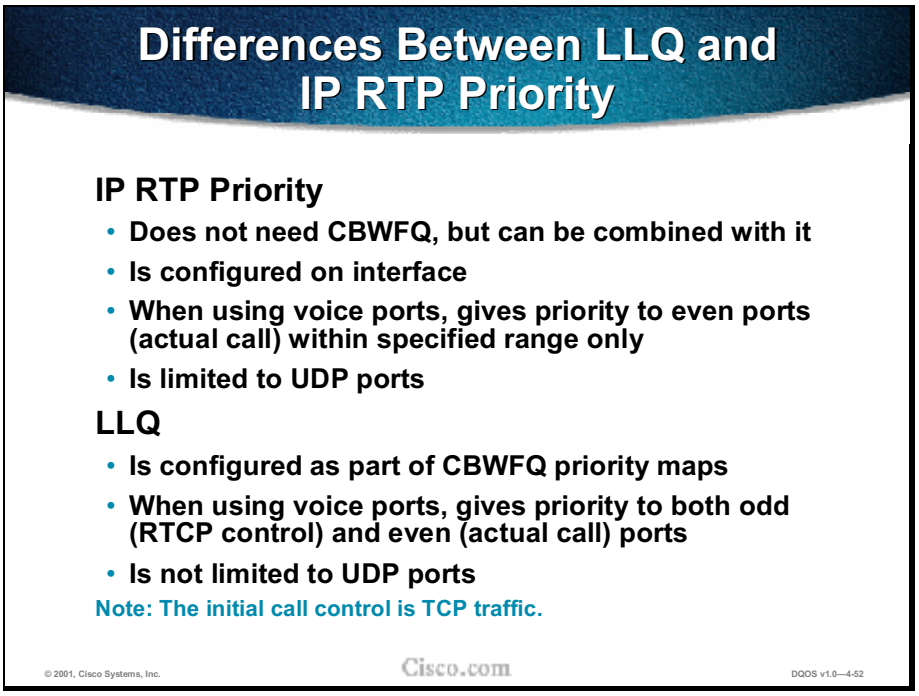
In the default class, the **fair-queue** command enables WRED congestion avoidance and configures ten dynamic queues for the default class (the class packets are assigned to if they are neither gold nor silver).

In the event of congestion, if gold traffic exceeds 512 kbps, gold traffic packets will be dropped.



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

In the second (lower box) access list 102 is configured to match the desired voice traffic. Then the class map *voice* is defined, and the policy map *policy1* is created; a strict-priority queue for the class *voice* is reserved, a bandwidth of 20 kbps is configured for the class *bar*, and the default class is configured for WFQ. The **service-policy** command then attaches the policy map to PVC 0/102 on subinterface atm1/0.



Some general factors must be considered in determining whether to configure LLQ:

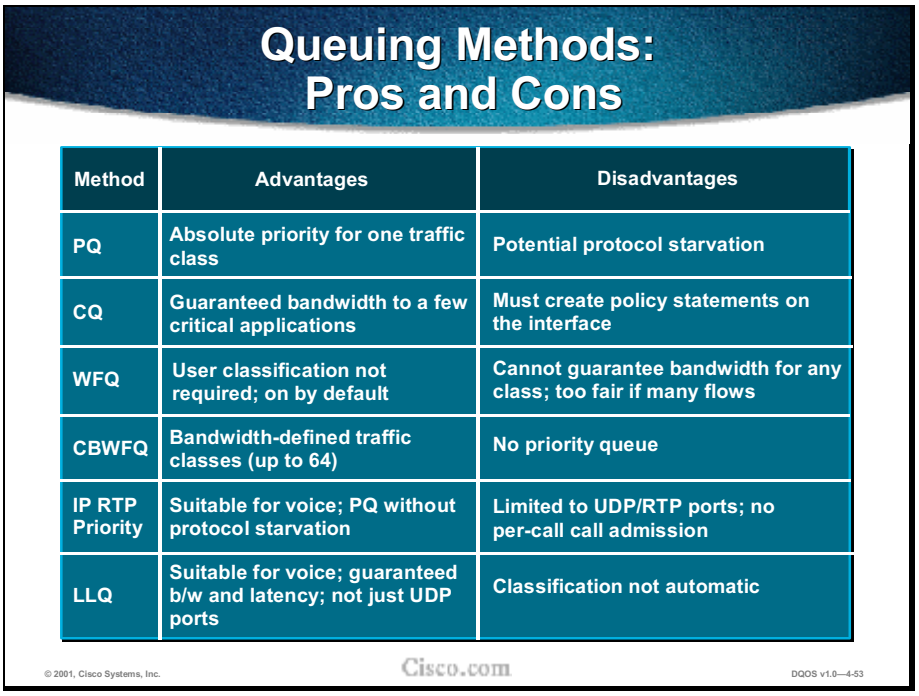
- LLQ provides strict-priority service on ATM VCs and serial interfaces. (The IP RTP Priority feature allows priority queuing only on interfaces.)

- LLQ is not limited to UDP port numbers. Because you can configure the priority status for a class within CBWFQ, you are no longer limited to UDP port numbers to stipulate priority flows. Instead, all the valid match criteria used to specify traffic for a class now apply to priority traffic.

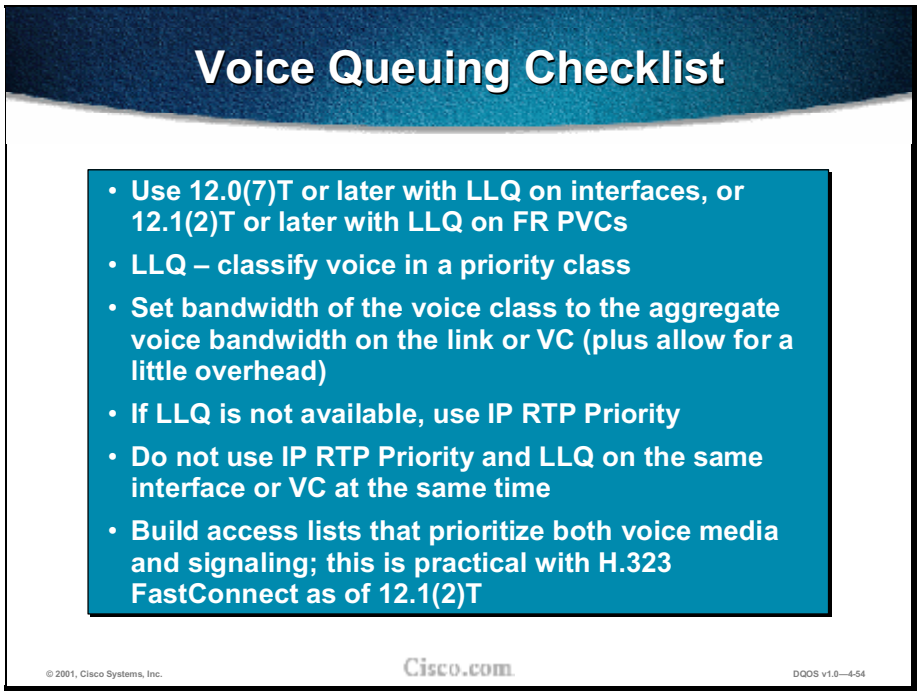
LLQ and IP RTP Priority can be configured at the same time (it is not recommended), but IP RTP Priority takes precedence. To demonstrate how they work together, consider the following configuration:

```
policy-map llqpolicy

class voice

priority 50

ip rtp priority 16384 16383 40

service-policy output llqpolicy
```

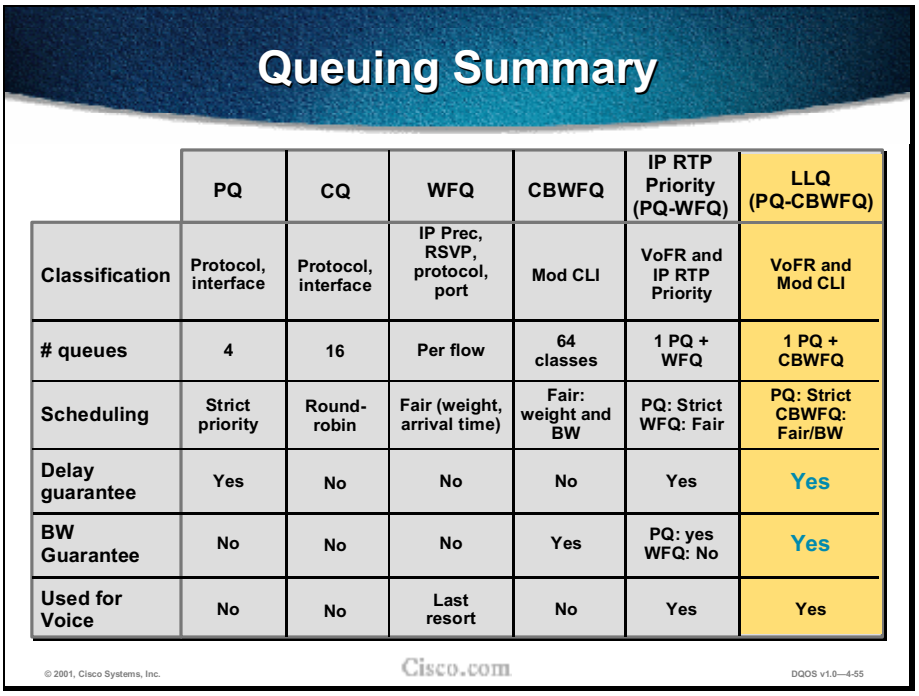

In this example, packets that match the specified port range will be given priority with 40 kbps bandwidth (the two port numbers are added, so the range is 16384 to 32767, the entire range of UPD ports used by all Cisco VoIP products); packets that match the *voice* class will be given priority with 50 kbps bandwidth. In the event of congestion, packets that match the specified port range will receive no more than 40 kbps of bandwidth, and packets that match the *voice* class will receive no more than 50 kbps of bandwidth. If packets match both criteria (ports 16384 to 32767 and class *voice*), IP RTP Priority takes precedence. In this example, the packets will be considered to match the 16384 to 32767 port range and will be accounted for in the 40 kbps bandwidth.



This figure shows some advantages and disadvantages of different queuing methods.



This figure provides a checklist to use when implementing queuing.



Consider these differences in deciding whether to use CQ or PQ:

- CQ guarantees some level of service to all traffic, because you can allocate bandwidth to all classes of traffic. You can define the size of the queue by determining its configured packet-count capacity, thereby controlling bandwidth access.

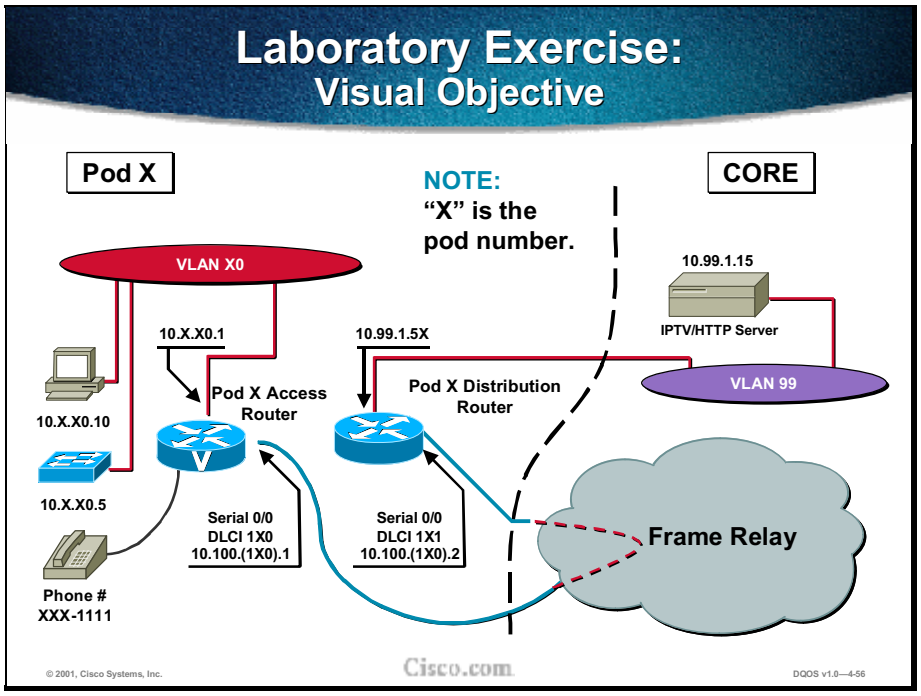
- PQ guarantees strict priority in that it ensures that one type of traffic will be sent, possibly at the expense of all others. For PQ, a low-priority queue can be detrimentally affected and, in the worst case, never allowed to send its packets if a limited amount of bandwidth is available or if the transmission rate of critical traffic is high.

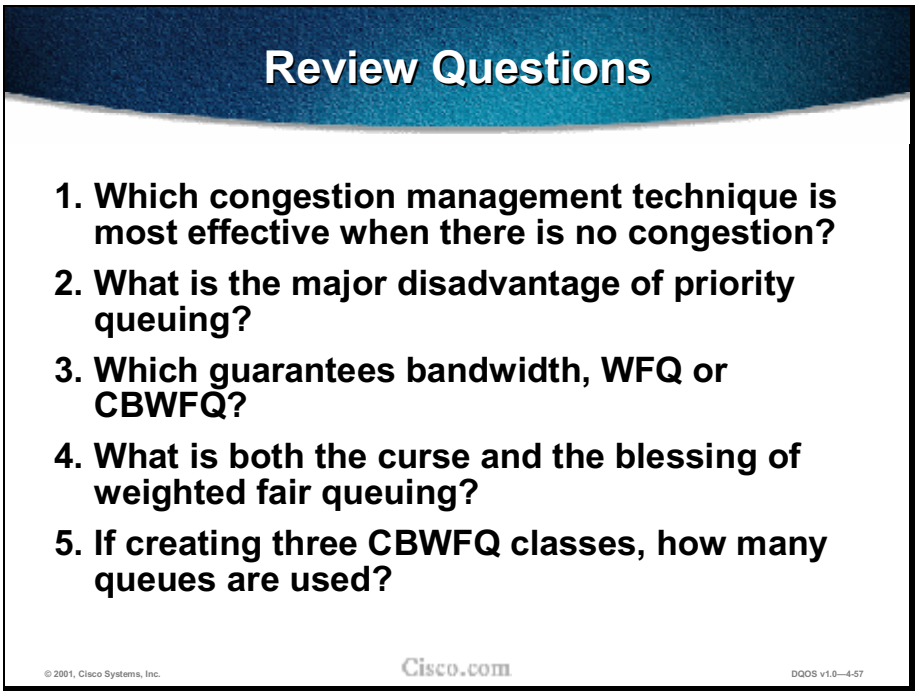
Consider in deciding whether to use WFQ, or to use either PQ or CQ:

- WFQ does not require configuration of access lists to determine the preferred traffic on a serial interface. Rather, the fair queue algorithm dynamically sorts traffic into messages that are part of a conversation.

- Low-volume interactive traffic gets fair allocation of bandwidth with WFQ, as does high-volume traffic such as file transfers.

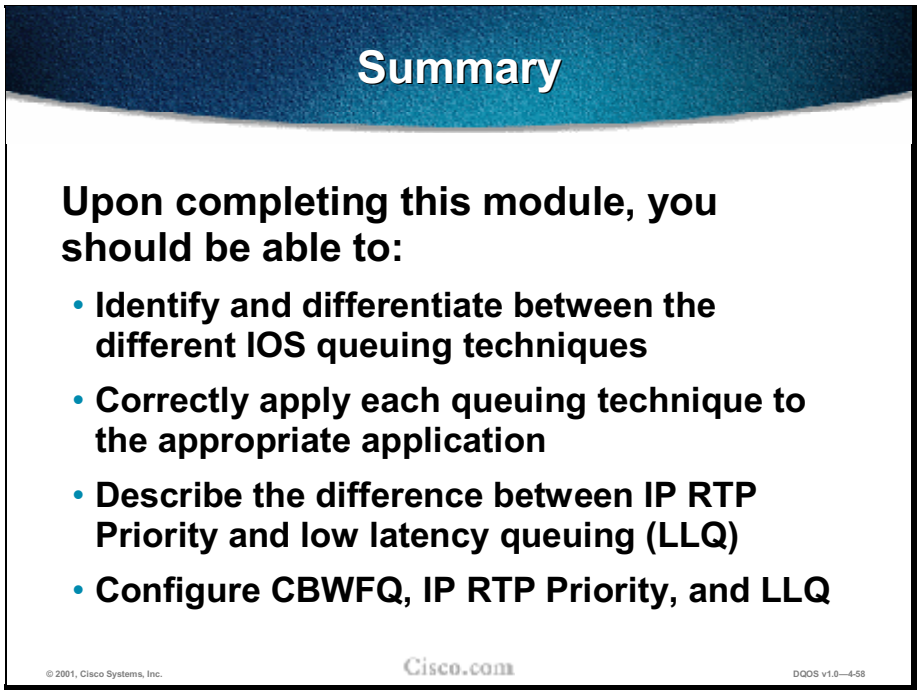
- Strict-priority queuing (using a separate priority queue for specified traffic) can be accomplished with WFQ by using IP RTP Priority (PQ/WFQ) or low latency queuing (LLQ). Strict-priority queuing allows delay-sensitive data such as voice to be dequeued and sent first (before packets in other queues are dequeued), giving delay-sensitive data preferential treatment over other traffic.


Laboratory Exercise:
Visual Objective
Pod X
CORE
NOTE:
"X" is the
pod number.
VLAN X0
10.99.1.15
IPTV/HTTP Server
10.X.X0.1
10.99.1.5X
Pod X Access
Router
Pod X Distribution
Router
VLAN 99
10.X.X0.10
10.X.X0.5
Serial 0/0
DLCI 1X0
10.100.(1X0).1
Serial 0/0
DLCI 1X1
10.100.(1X0).2
Frame Relay
Phone #
XXX-1111
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—4-56



1. Which congestion management technique is most effective when there is no congestion?
2. What is the major disadvantage of priority queuing?
3. Which guarantees bandwidth, WFQ or CBWFQ?
4. What is both the curse and the blessing of weighted fair queuing?
5. If creating three CBWFQ classes, how many queues are used?

Summary


Summary
Upon completing this module, you should be able to:
• Identify and differentiate between the different IOS queuing techniques
• Correctly apply each queuing technique to the appropriate application
• Describe the difference between IP RTP Priority and low latency queuing (LLQ)
• Configure CBWFQ, IP RTP Priority, and LLQ
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—4-58

5

Congestion Avoidance

Overview

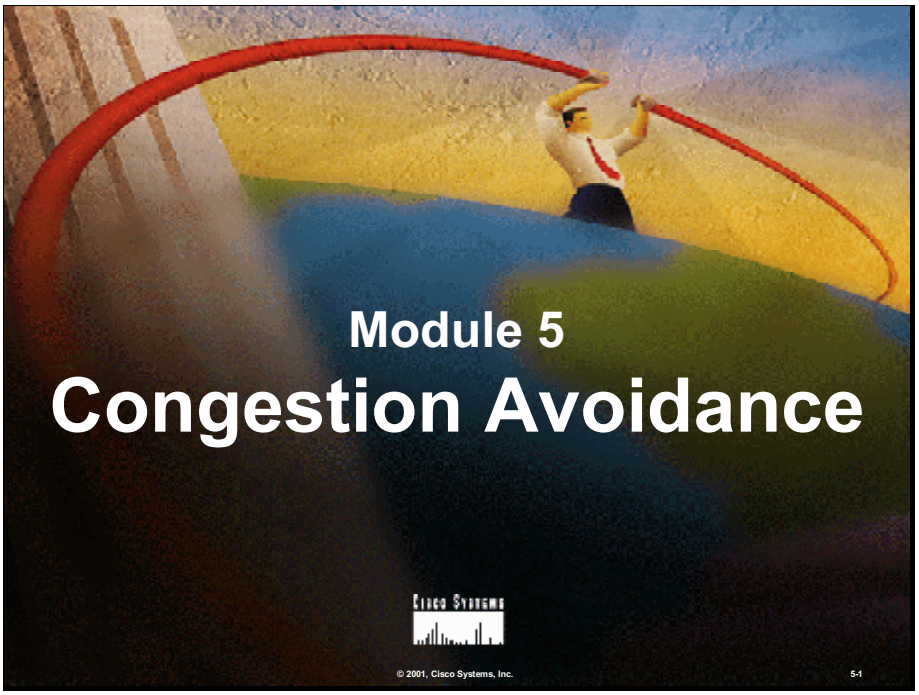
This chapter explains the behavior of the TCP traffic during congestion. Congestion avoidance techniques such as RED, WRED, and its different variants are covered and include configuration examples.

Objectives

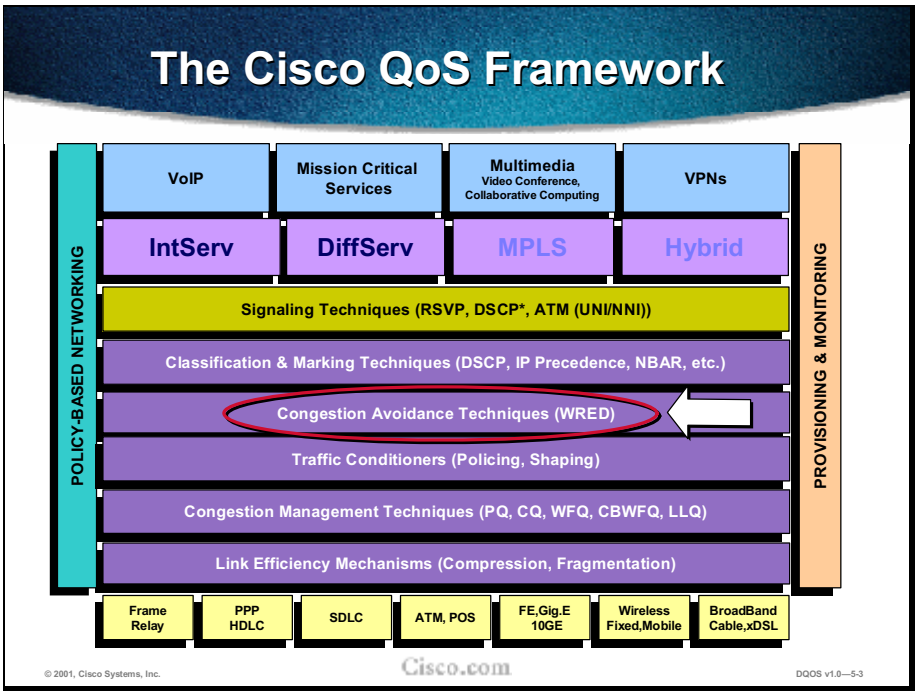
Upon completing this chapter, you will be able to:

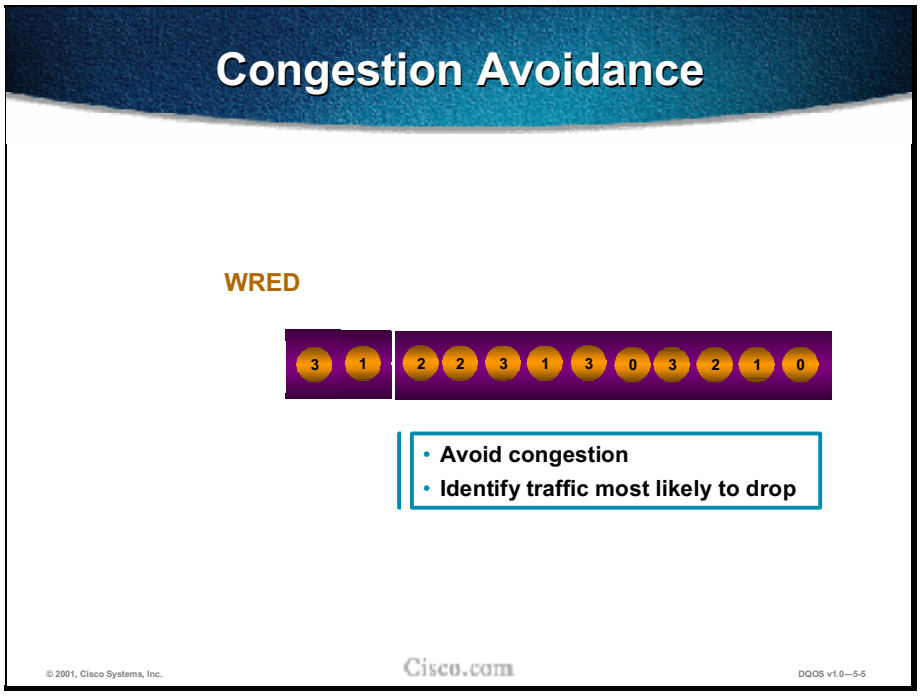
- Explain how TCP responds to congestion
- Understand tail drop and global synchronization
- Identify and differentiate between the following IOS congestion avoidance tools: RED, WRED, and flow-based WRED
- Configure IOS congestion avoidance features

Outline



The figure shows the plan for the week.





Congestion avoidance techniques monitor network traffic loads in an effort to anticipate and avoid congestion at common network bottlenecks. Congestion avoidance is achieved through packet dropping.

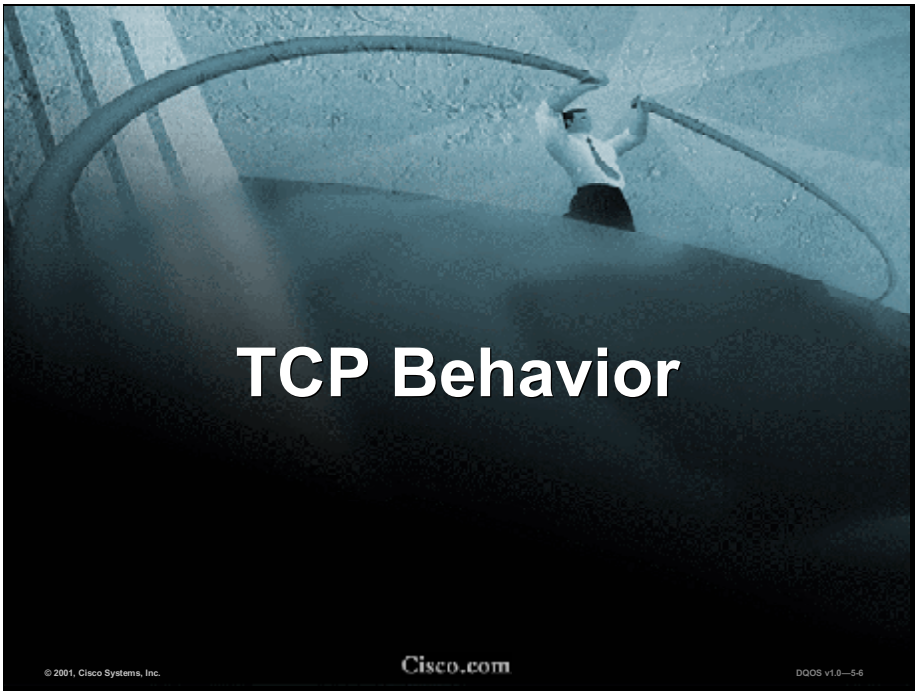
The nonanimated “hard copy” of this slide shows a full queue, which is an indication that the network is becoming, or has become, congested. The animated slide compares two ways the network can respond to congestion:

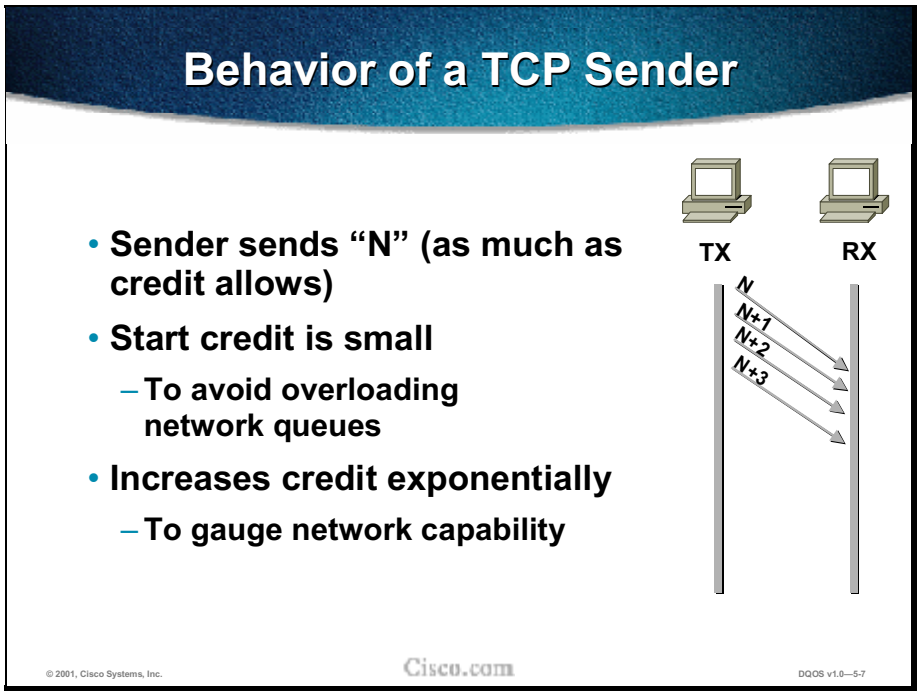
- Tail drop is the default queuing response to congestion. When the output queue is full and tail drop is in effect, all packets trying to enter (at the tail of) the queue are dropped until the congestion is eliminated and the queue is no longer full.

- Weighted random early detection (WRED) increases the probability that congestion is avoided by dropping low-priority packets rather than high-priority packets and by dropping packets randomly so there is no “global synchronization” reaction. When Transmission Control Protocol (TCP) stations drop packets, it causes transmitting stations to throttle back their transmission rates in order to avoid further congestion.

Note that WRED is NOT recommended for any voice queues that carry voice traffic (there is no IP Precedence set to 5 in the example). The network should not be designed to drop voice packets since lost voice packets result in reduced voice quality. Furthermore, voice payload is carried in the User Datagram Protocol (UDP), so dropping voice packets would not ease congestion much anyway. WRED controls congestion by impacting other TCP-based traffic, and avoiding congestion helps to ensure voice quality.

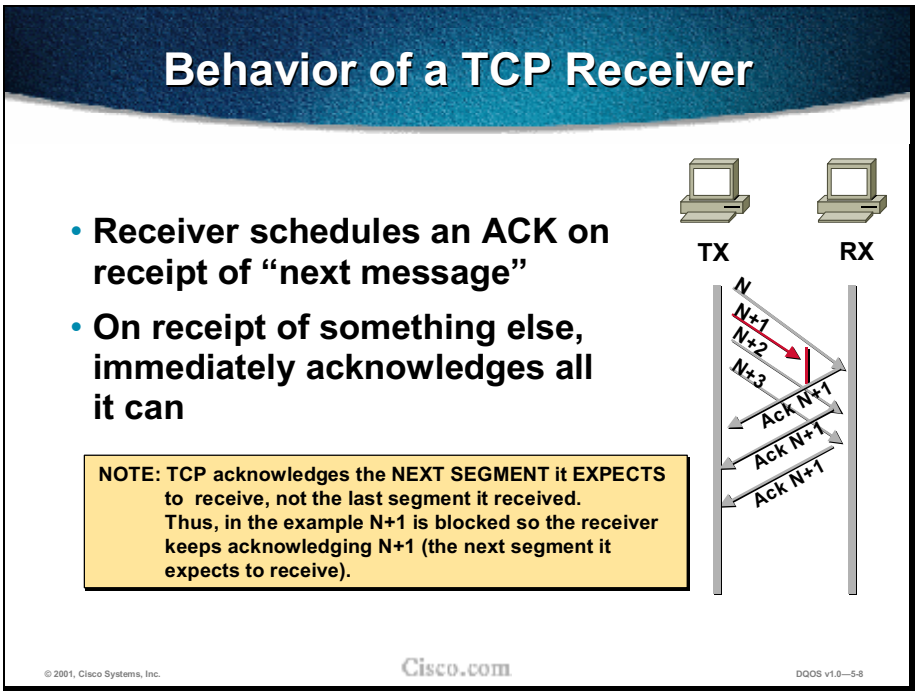
Tail drop and WRED are discussed in this chapter in more detail, but first it is helpful to understand how TCP recognizes and responds to congestion.





Before any data is transmitted using TCP, a connection must first be established between the transmitting and receiving hosts. When the connection is initially established, the two hosts must agree on certain parameters to use for the communication session. One of the parameters that must be decided is called the window size, or how many TCP segments to transmit at a time.

As the slide shows, TCP initially sends a small number of segments, then exponentially increases the number sent.

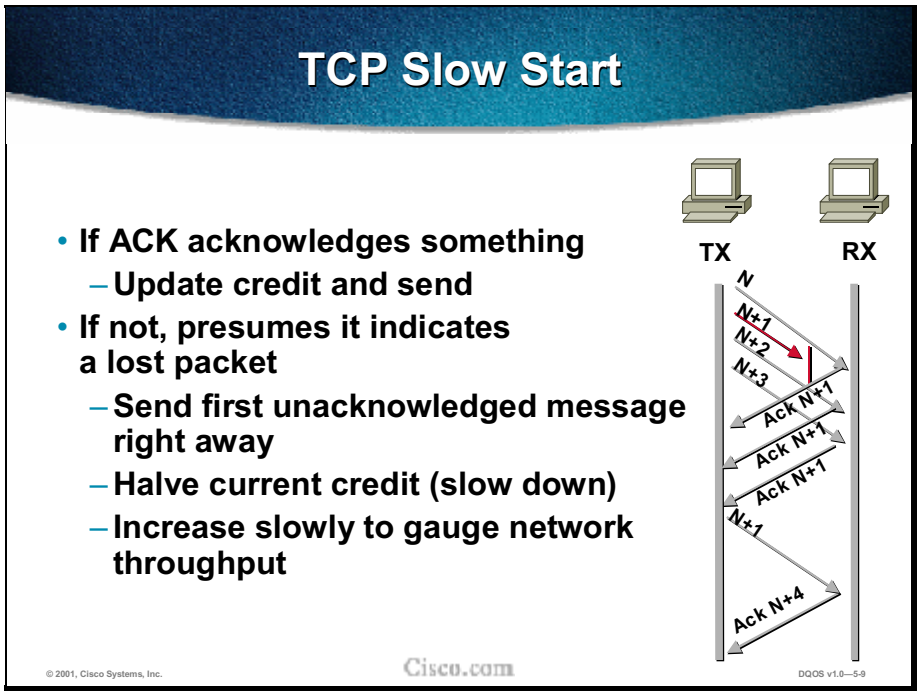


When the receiver receives a data segment, it checks that data segment's sequence number. If the sequence number matches the number the receiver expected, it indicates that the data segment was received in order.

If the numbers match, the receiver:

- Delivers all the data that it holds to the target application
- Then it updates the sequence number to reflect the next number in order
- Finally it either:
  - Immediately transmits an acknowledgment (ACK) to the sender, or
  - Schedules an ACK to be transmitted to the sender after a short delay

The ACK notifies the sender that the receiver received all data segments up to but not including the one marked with the new sequence number. Receivers usually try to send an ACK in response to alternating data segments they receive. They send the ACK because for many applications, if the receiver waits out a small delay, it can efficiently piggyback its reply acknowledgment on a normal response to the sender. However, when the receiver receives a data segment out of order, it immediately responds with an ACK to direct the sender to retransmit the lost data segment.

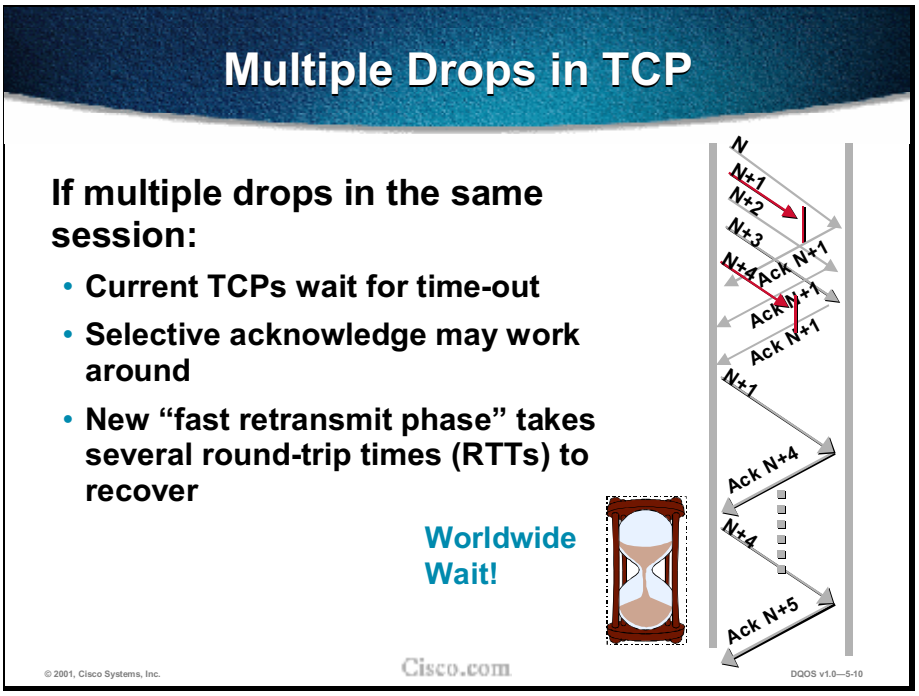


When the sender receives an ACK, it determines if any data is outstanding:

- If not, the sender determines that the ACK is a keepalive, meant to keep the line active, and it does nothing.
- If data is outstanding, the sender determines whether the ACK indicates that the receiver has received some or none of the data.
  - If the ACK acknowledges receipt of some data sent, the sender determines if new credit has been granted to allow it to send more data.
  - When the ACK acknowledges receipt of none of the data sent and there is outstanding data, the sender interprets the ACK to be a repeatedly sent ACK. This condition indicates that some data was received out of order, forcing the receiver to remit the first ACK, and that a second data segment was received out of order, forcing the receiver to remit the second ACK. In most cases, the receiver would receive two segments out of order because one of the data segments had been dropped.

When a TCP sender detects a dropped data segment, it retransmits the segment. Then it slows its transmission rate so that the rate is half of what it was before the drop was detected. This is known as the TCP slow-start mechanism.

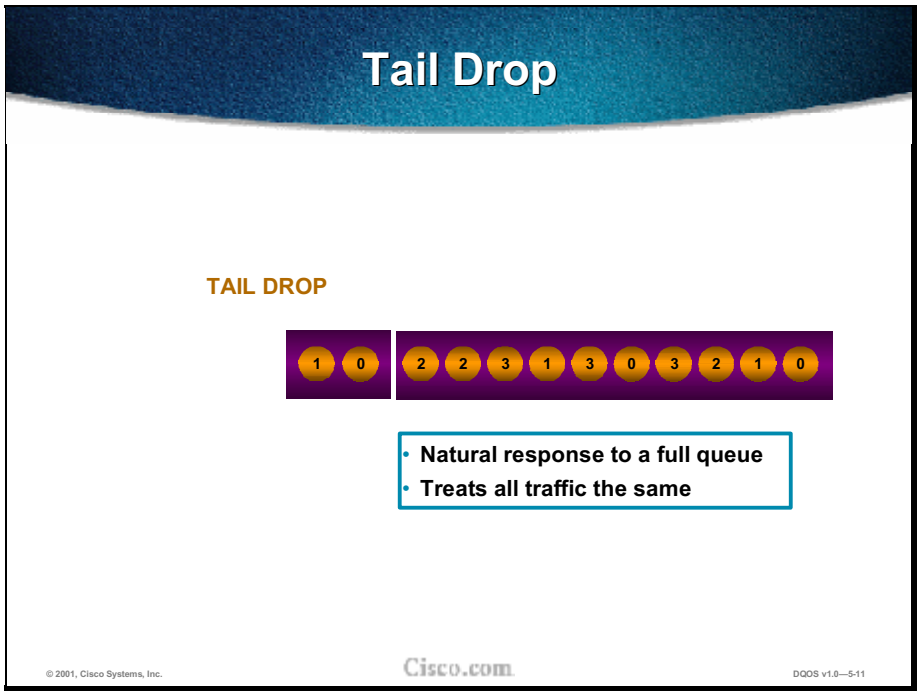
Further details on TCP slow start can be found in RFC 2001.



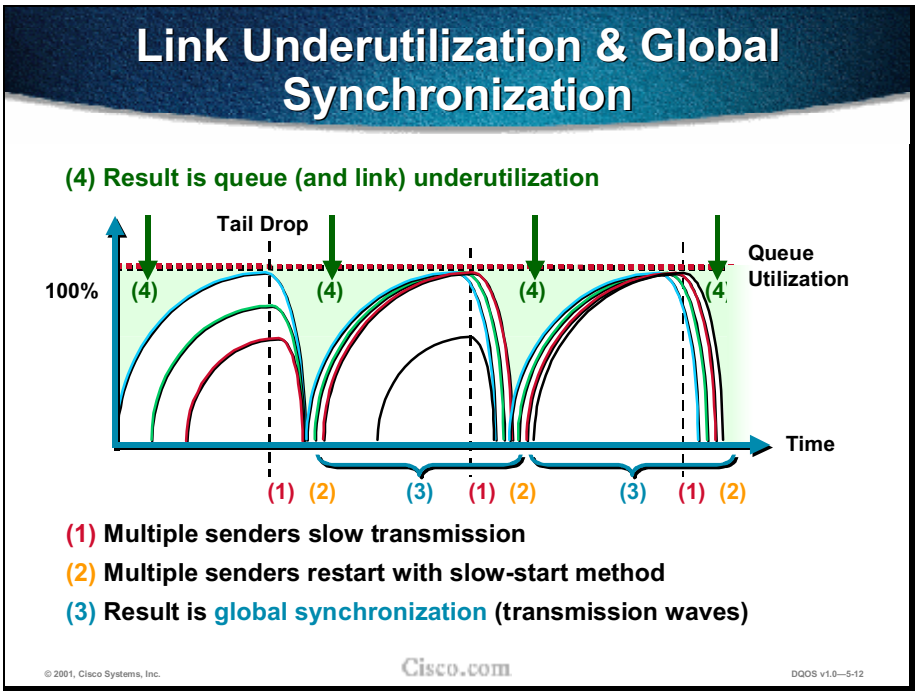
Although the TCP slow-start behavior is appropriately responsive to congestion, problems can arise when multiple TCP sessions are carried on concurrently with the same router and all TCP senders slow down transmission of packets at the same time.

For a detailed discussion of TCP congestion behavior see Geoff Huston, Telstra, "The Future for TCP," *Internet Protocol Journal,* Vol. 3, No. 2, June 2000, at the following URL:

http://www.cisco.com/warp/public/759/ipj_3-3/ipj_3-3_futureTCP.html



Tail drop is the default queuing response to congestion. Tail drop treats all traffic equally and does not differentiate between classes of service. Queues fill during periods of congestion. When the output queue is full and tail drop is in effect, packets are dropped until the congestion is eliminated and the queue is no longer full.



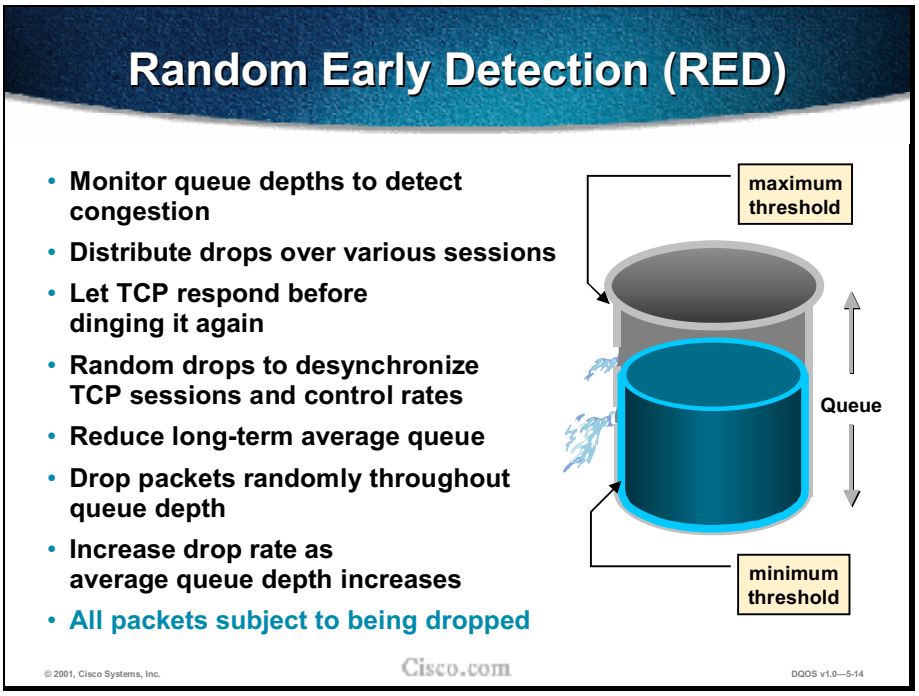
A router can handle multiple concurrent TCP sessions. There is a high probability that when traffic exceeds the queue limit at all, it vastly exceeds the limit. However, there is also a high probability that the excessive traffic depth is temporary and that traffic does not stay excessively deep except at points where traffic flows merge or at edge routers.

If the receiving router drops all traffic that exceeds the queue limit, as is done by default (with tail drop), many TCP sessions then simultaneously go into slow start. Consequently, traffic temporarily slows down to the extreme and then all flows slow-start again. This activity creates a condition of global synchronization.

Global synchronization occurs as waves of congestion crest, only to be followed by troughs during which the transmission link is not fully utilized. Global synchronization of Transmission Control Protocol (TCP) hosts, for example, can occur because packets are dropped all at once. Global synchronization manifests when multiple TCP hosts reduce their transmission rates in response to packet dropping, then increase their transmission rates once again when the congestion is reduced.

The most important point to understand from this slide is that the waves of transmission known as global synchronization result in significant link underutilization.


Avoidance Techniques
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—5-13



Before periods of high congestion, random early detection (RED) randomly drops packets. This tells the randomly selected packet source to decrease its transmission rate. If the packet source is using TCP, it decreases its transmission rate until all the packets reach their destination, indicating that the congestion is cleared. Directing one TCP session at a time to slow down allows for full utilization of the bandwidth, rather than utilization manifesting as crests and troughs of traffic.

The probability of a packet being dropped is based on three configurable parameters:

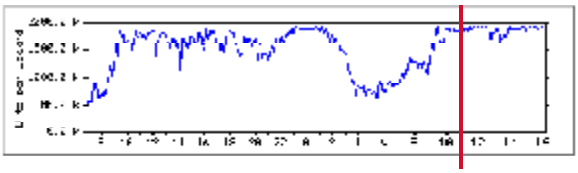
- Minimum threshold—When the average queue depth is above the minimum threshold, RED starts dropping packets. The rate of packet drop increases linearly as the average queue size increases, until the average queue size reaches the maximum threshold.

- Maximum threshold—When the average queue size is above the maximum threshold, all packets are dropped. If the difference between the maximum threshold and the minimum threshold is too small, many packets might be dropped at once, resulting in global synchronization.

- Mark probability denominator—This is the fraction of packets dropped when the average queue depth is at the maximum threshold. For example, if the denominator is 512, one out of every 512 packets is dropped when the average queue is at the maximum threshold.

RED distributes losses in time and maintains normally low queue depth while absorbing spikes.

Effect of RED



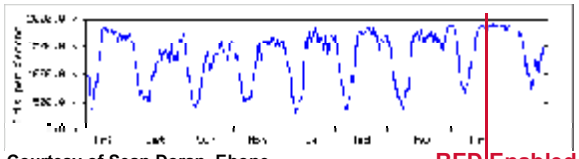
Courtesy of Sean Doran, Ebone

RED Enabled

- One day, below 100% throughput
  - Simple FIFO with tail drop
- Starting 10:00 second day, 100% throughput
  - Random early detection enabled

This graphic, a sample captured from a live network, illustrates the effects of using RED/WRED. The graph to the left of the vertical bar shows that the link was not fully utilized during peak periods before RED was enabled. To the right of the vertical represents the period after RED was enabled on the link. It is clear that link utilization improved after RED was enabled on the link.

Was That a Fluke?



Courtesy of Sean Doran, Ebone

No, here's what happened that week…

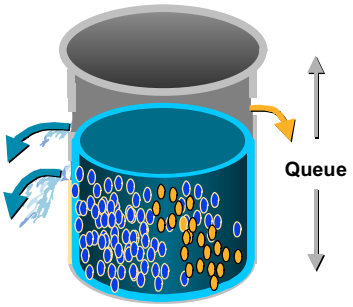
- **A similar graph over a long duration shows that RED clearly improved link utilization.**

Weighted RED

Weighted Random Early Detection (WRED):

- Cisco's implementation of RED
- If packets need to be dropped, lower priority traffic dropped first

DiffServe compliant 12.1(5)T



BLUE packets start dropping at a 50% average queue depth. Drop rate increases as average queue depth increases.

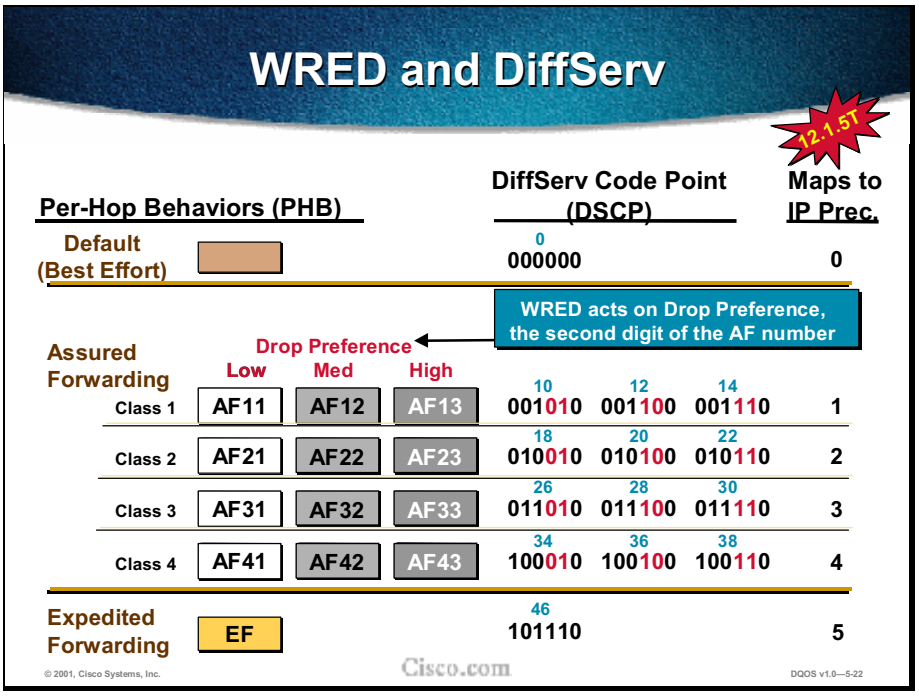
How WRED Works

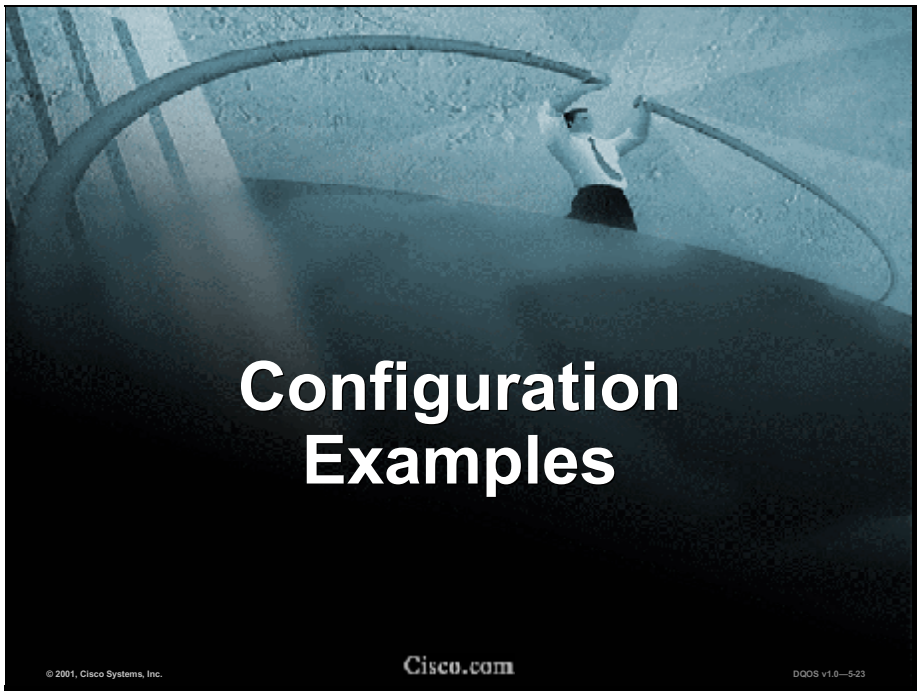
WRED

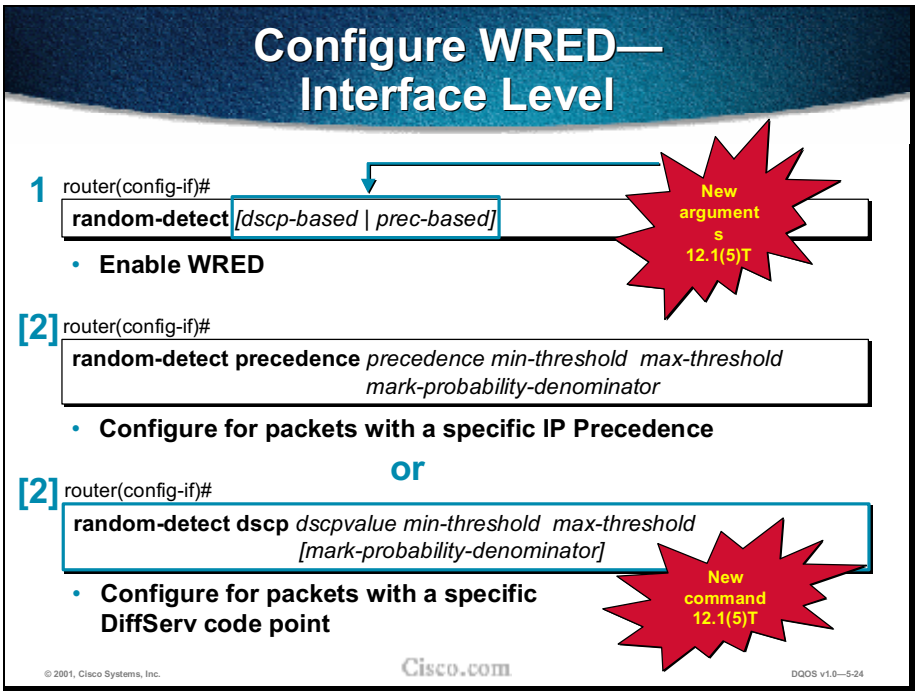
- Avoid congestion
- Identify traffic most likely to drop
- Not used for queues that carry voice

Flow-Based WRED (FRED)

- Extension of WRED
  - Classifies packets by flow (for example, source address, destination address, port)
  - Tracks flow of each packet in output queue
- Penalizes flows that do not respond to drops (e.g. UDP)
- No single flow hogs all the buffer resources
- Adaptive flows get fair share of resources




Configuration Examples
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—5-23



For complete details on configuring WRED and flow-based WRED at the interface level, see *Configuring Weighted Random Early Detection* at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcprt3/qcdwred.htm

IOS 12.1(5)T added support for differentiated services code point (DSCP) to the WRED feature. The DSCP value is the first six bits of the IP type of service (ToS) byte.

The following interface-level command is added:

random-detect dscp—Change the minimum and maximum packet thresholds

Also, two new arguments are added to the existing **random-detect** command:

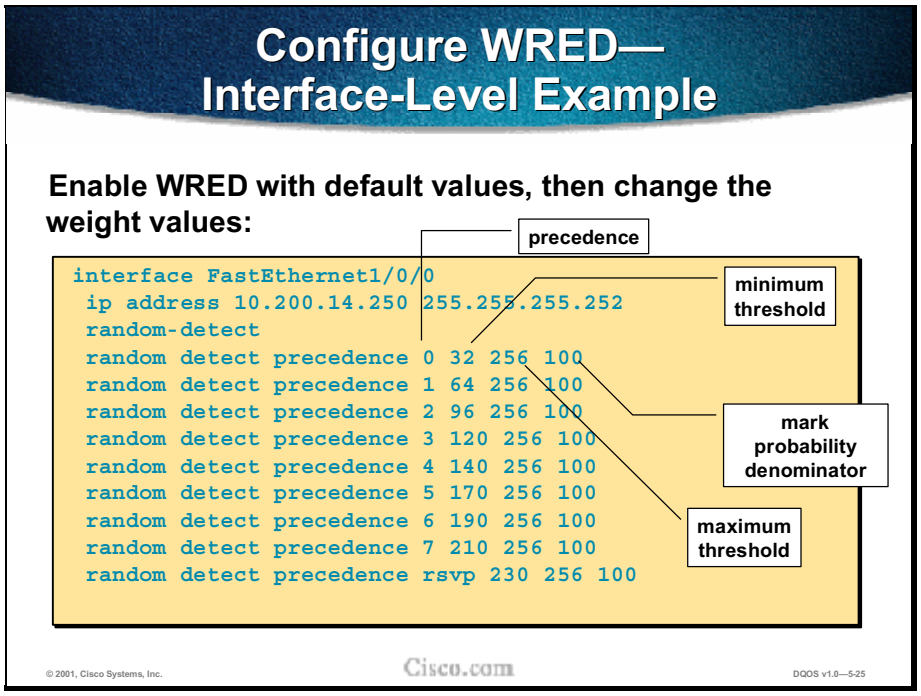
dscp-based—Use the DSCP value of a packet to calculate drop probability for the packet

prec-based—Use the IP Precedence value of a packet to calculate drop probability for the packet (default)

These arguments are optional and mutually exclusive.

For additional details on DiffServ-compliant WRED, see the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/dtdswred.htm#xtocid1550421

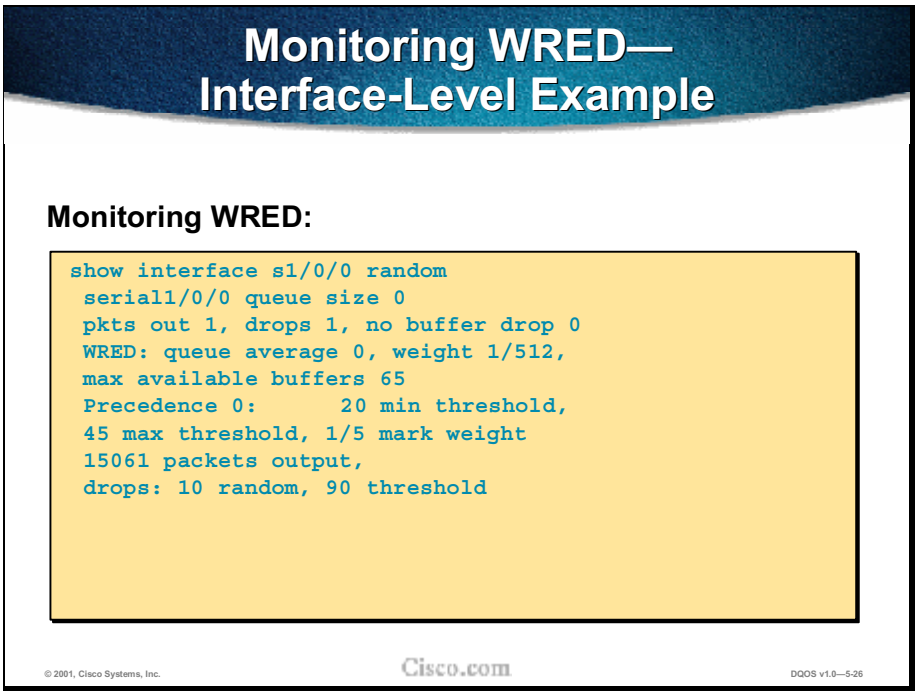


In this example, WRED is enabled with default values, then the values are changed for each IP Precedence level. The configured values, which are described above under "Random Early Detection," are repeated here for convenience:

- Minimum threshold—When the average queue depth is above the minimum threshold, RED starts dropping packets. The rate of packet drop increases linearly as the average queue size increases, until the average queue size reaches the maximum threshold.

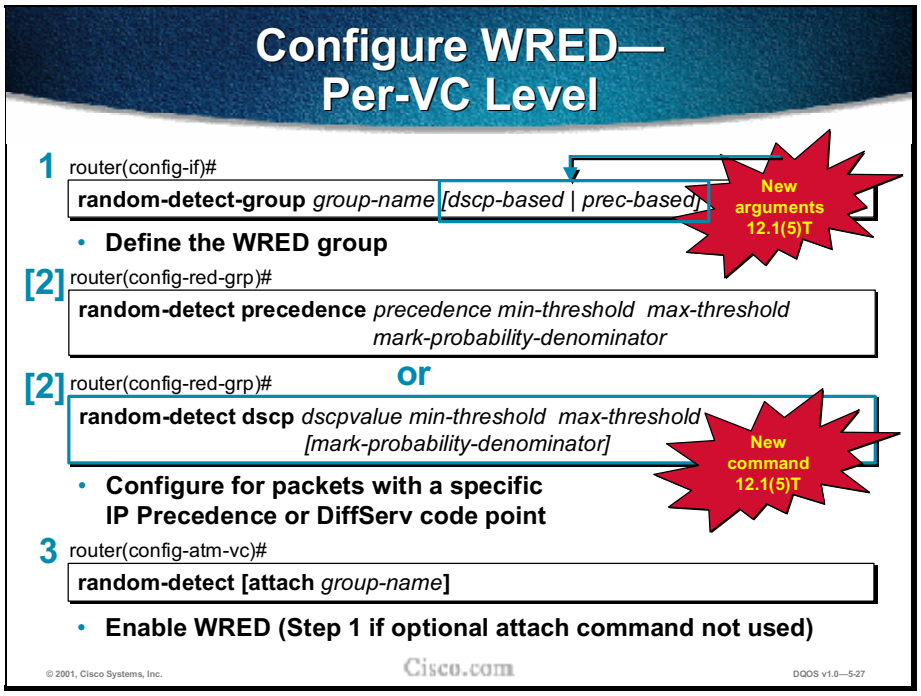
- Maximum threshold—When the average queue size is above the maximum threshold, all packets are dropped. If the difference between the maximum threshold and the minimum threshold is too small, many packets might be dropped at once, resulting in global synchronization.

- Mark probability denominator—This is the fraction of packets dropped when the average queue depth is at the maximum threshold. For example, if the denominator is 512, one out of every 512 packets is dropped when the average queue is at the maximum threshold.



Use the **show interfaces** command to show the WRED configuration on an interface. The command syntax is:

show interfaces *[type slot | port-adapter | port]*



IOS 12.1(5)T added support for differentiated services code point (DSCP) to the WRED feature. The following new per-VC-level command is added:

Dscp—Change the minimum and maximum packet thresholds for per-VC configuration

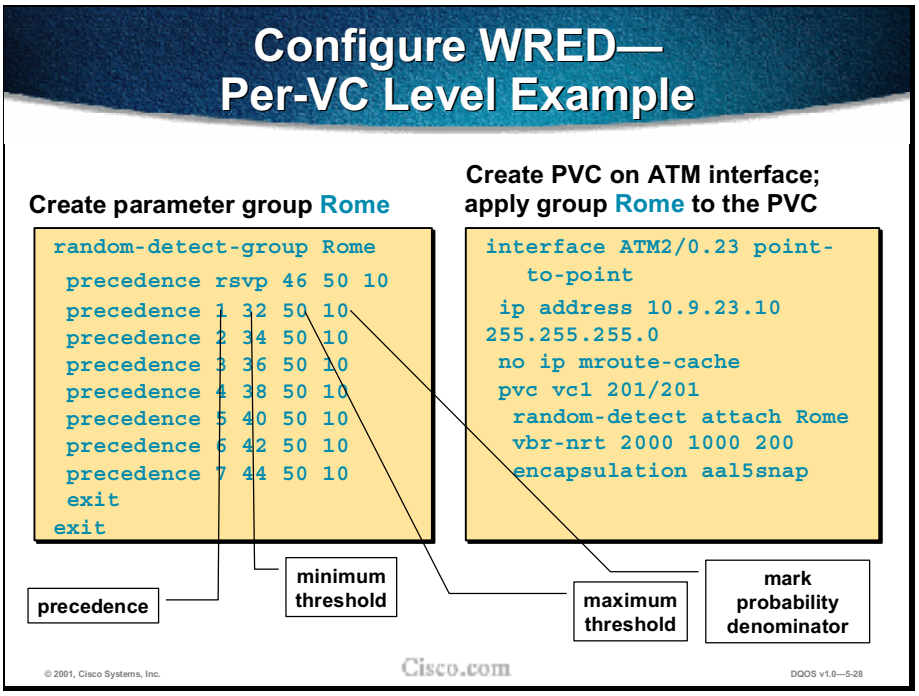
Also, two new arguments are added to the existing per-VC-level **random-detect-group** command:

- *dscp-based*—Use the DSCP value of a packet to calculate drop probability for the packet
- *prec-based*—Use the IP Precedence value of a packet to calculate drop probability for the packet (default)

These arguments are optional and mutually exclusive. That is, if you use the *dscp-based* argument, you cannot use the *prec-based* argument with the same command.

For additional details on DiffServ-compliant WRED and per-VC configuration, see the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/dtdswred.htm#xtocid1550421



The example shown is actually recommended for distributed WRED (DWRED); distributed WRED is simply an implementation of WRED for high-speed applications using the Versatile Interface Processor (VIP).

The commands on the left side of the slide create a DWRED group called *Rome.* The commands on the right side create a PVC on an ATM interface and then apply the DWRED group *Rome* to that PVC.

Display WRED—Per-VC Settings

Display per-VC settings:

```
show queueing random-detect interface atm2/0.23 vc 201/201
 random-detect group Rome:
 exponential weight 9
class     min-threshold    max-threshold    mark-prob.
----------------------------------------------------
0         30               50               1/10
1         32               50               1/10
2         34               50               1/10
3         36               50               1/10
4         38               50               1/10
5         40               50               1/10
6         42               50               1/10
7         44               50               1/10
rsvp      46               50               1/10
```


The **show queuing** command displays the current settings for each of the IP Precedences.

Configure WRED—Class Level

1 router(config)#

class map *class-map-name*

- **Create class map.**

2 router(config-cmap)#

match access group *[access-group* | **name** *access-group name]*

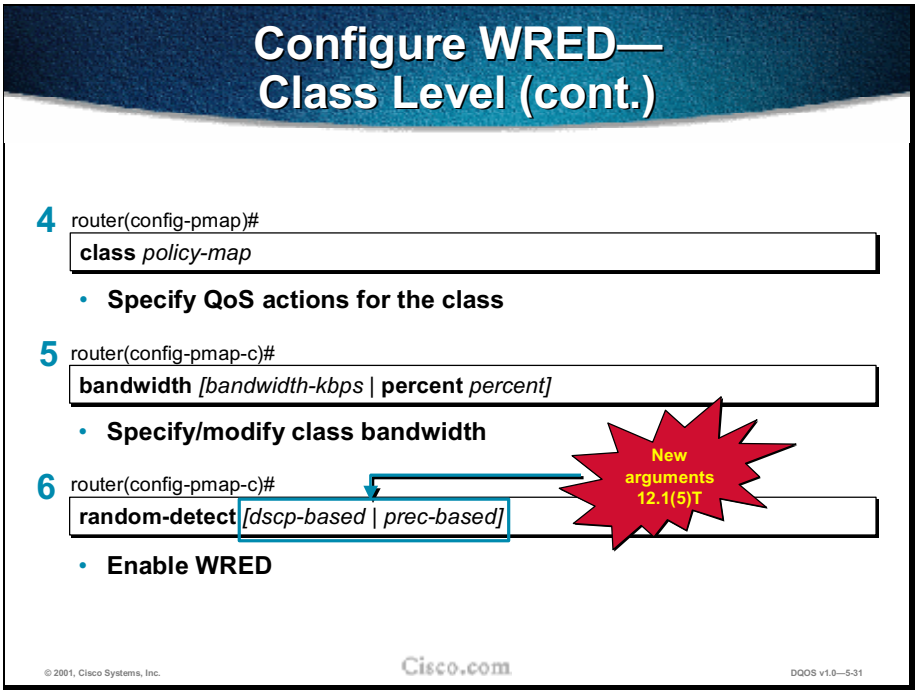
- **Configure match criteria based on access control list.**

3 router(config)#

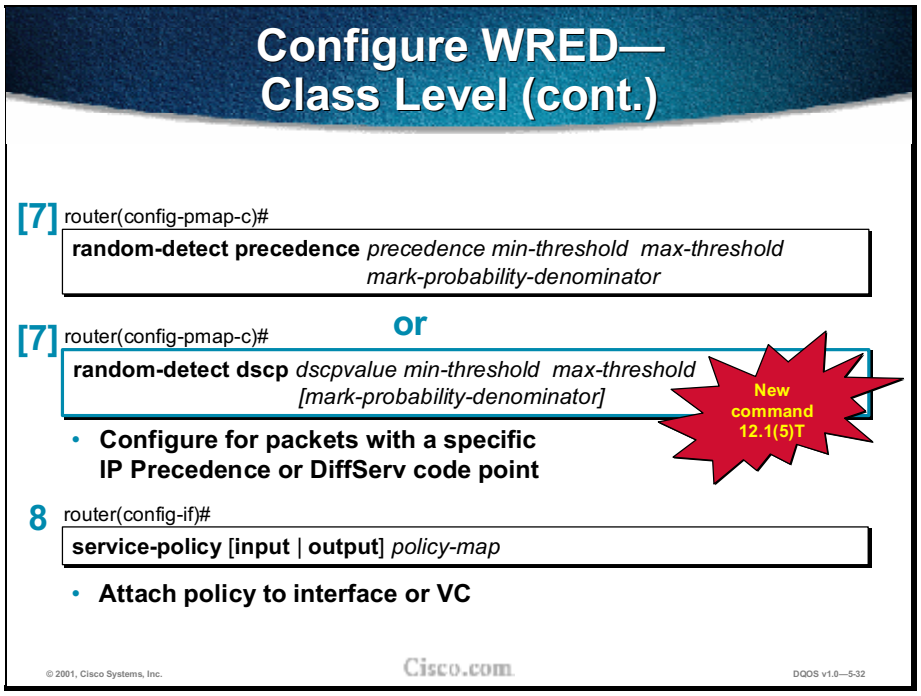
policy map *policy-map*

- **Create/modify policy map**

The slides show the steps for configuring WRED to use the IP Precedence value or the DSCP value when it calculates the drop probability. These are the commands to use at the class level within policy maps.



The optional *dscp-based* and *prec-based* arguments were added in IOS 12.1(5)T. If not specified, IP Precedence base is assumed.



Note that there are two possible forms for Step 7 for specifying either IP Precedence or DSCP values. If you use the DSCP form, you must first specify *dscp-based* in Step 6 (see the previous slide).

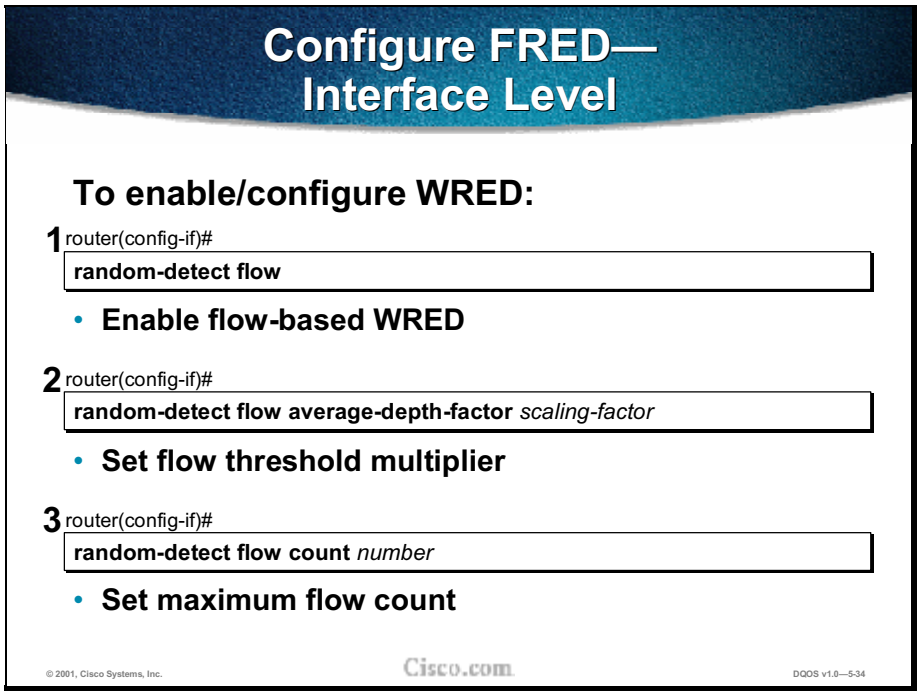
The DSCP value can be a number from 0 to 63, or it can be one of the following standard per-hop behaviors (PHB):

- ef, af11, af12, af13, af21, af22, af23, af31, af32, af33, af41, af42, af43, cs1, cs2, cs3, cs4, cs5, or cs7

Configure WRED—Class Level Example

Enable WRED to use DSCP value 8 for class c1; attach the service policy to output interface or VC p1 :

```
(config)# class-map c1
(config-cmap)# match access-group 101
(config)# policy-map p1
(config-pmap)# class c1
(config-pmap-c)# bandwidth 48
(config-pmap-c)# random-detect dscp-based
(config-pmap-c)# random-detect dscp 8 24 40
(config-if)# service-policy output p1
```

To enable flow-based WRED, use the **random-detect flow** interface configuration command.

— random-detect flowYou must use this command to enable flow-based WRED before you can use the **random-detect flow average-depth-factor** and **random-detect flow count** commands to further configure the parameters of flow-based WRED.

To set the multiplier to be used in determining the average depth factor for a flow when flow-based WRED is enabled, use the **random-detect flow average-depth-factor** interface configuration command:

— random-detect flow average-depth-factor *scaling-factor*Use this command to specify the scaling factor that flow-based WRED should use in scaling the number of buffers available per flow and in determining the number of packets allowed in the output queue for each active flow. This scaling factor is common to all flows. The outcome of the scaled number of buffers becomes the per-flow limit. If this command is not used and flow-based WRED is enabled, the average depth scaling factor defaults to 4.

To set the flow count threshold for flow-based WRED, use the **random-detect flow count** interface configuration command:

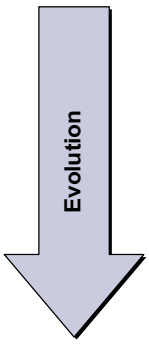
— random-detect flow count *number*

Configure FRED—Interface-Level Example

After enabling/configuring WRED, enable and configure flow-based WRED:

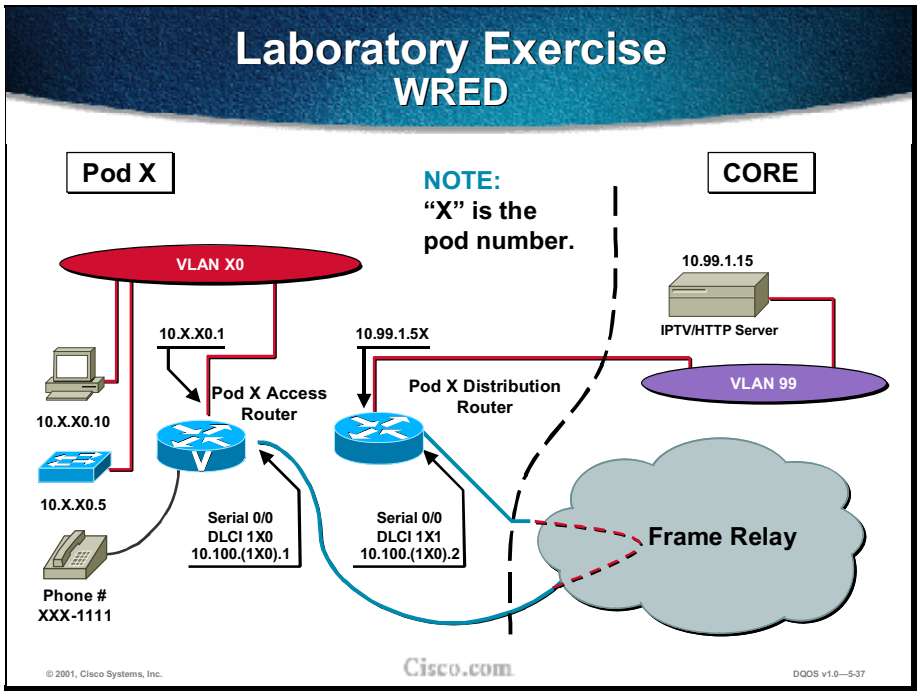
```
interface serial1
 ip address 10.200.14.250 255.255.255.253
 random-detect
 random detect flow
 random detect flow average-depth-factor 8
 random detect flow count 16
```

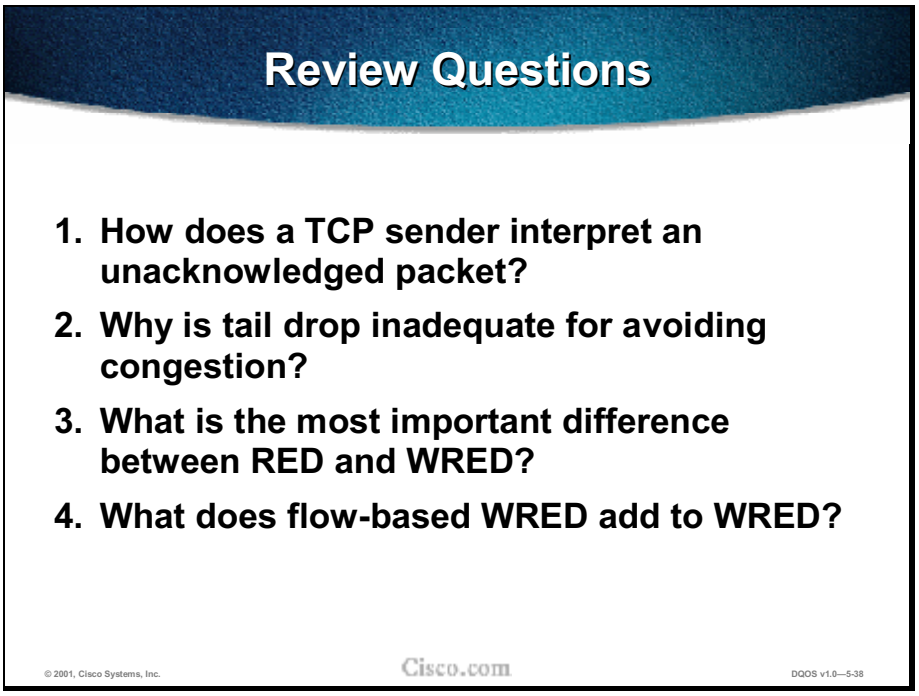

Recap of Packet-Dropping Techniques



Behavior

	Precedence Aware	Flow Aware
Tail Drop	No	No
RED	No	No
WRED	Yes	No
FRED	Yes	Yes


Laboratory Exercise
WRED
Pod X
CORE
NOTE:
"X" is the
pod number.
VLAN X0
10.99.1.15
IPTV/HTTP Server
10.X.X0.1
10.99.1.5X
Pod X Access
Router
Pod X Distribution
Router
VLAN 99
10.X.X0.10
10.X.X0.5
Serial 0/0
DLCI 1X0
10.100.(1X0).1
Serial 0/0
DLCI 1X1
10.100.(1X0).2
Frame Relay
Phone #
XXX-1111
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—5-37



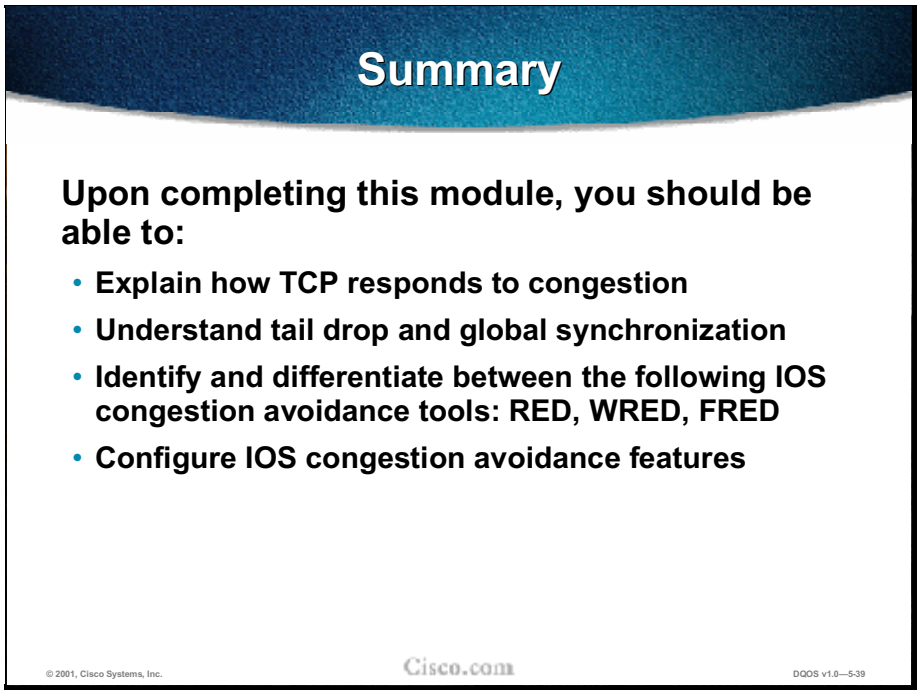
Q1) How does a TCP sender interpret an unacknowledged packet?

Q2) Why is tail drop inadequate for avoiding congestion?

Q3) What is the most important difference between RED and WRED?

Q4) What does flow-based WRED add to WRED?

Summary


Summary
Upon completing this module, you should be able to:
• Explain how TCP responds to congestion
• Understand tail drop and global synchronization
• Identify and differentiate between the following IOS congestion avoidance tools: RED, WRED, FRED
• Configure IOS congestion avoidance features
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—5-39

6

Link Efficiency Tools

Overview

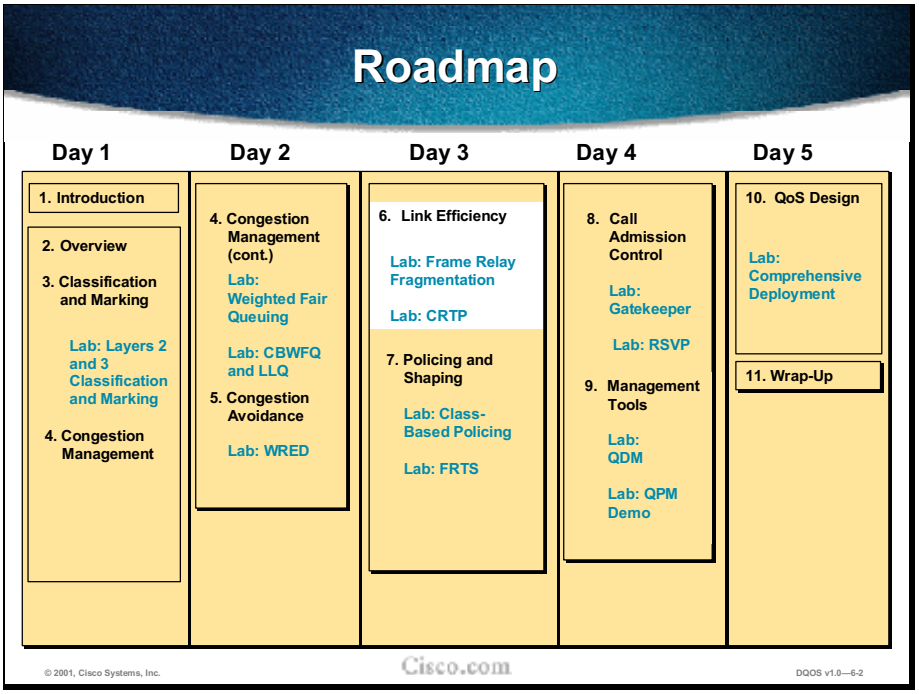
This chapter reviews various link efficiency tools, including configuration. Tools covered include link fragmentation and interleaving, Frame Relay fragmentation (using FRF.11, FRF.12 and CRTP). It concludes with a lab exercise.

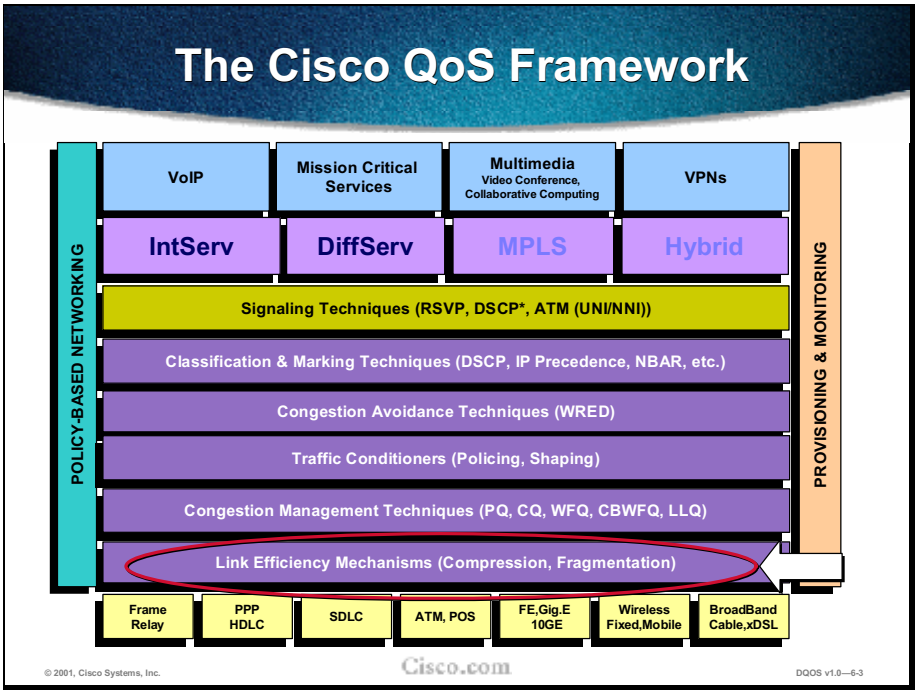
Objectives

Upon completing this chapter, you will be able to:

- Understand the need for link efficiency tools
- Understand available LFI techniques, including MLP interleaving and Frame Relay fragmentation using FRF.11 Annex C or FRF.12
- Understand Compressed Real-Time Protocol (CRTP) header as a tool for improving link efficiency
- Configure and monitor various LFI methods and CRTP

Outline





Some Definitions

Term	Definition
LFI	Link fragmentation and interleaving
MLP Interleaving	LFI for multilink PPP media
FRF.12	LFI for Frame Relay data PVCs
FRF.11 Annex C	LFI for Frame Relay VoFR PVCs
CRTP	Header compression for Real-Time Transport Protocol

It is useful before proceeding to provide working definitions for terms commonly used to describe link efficiency tools.

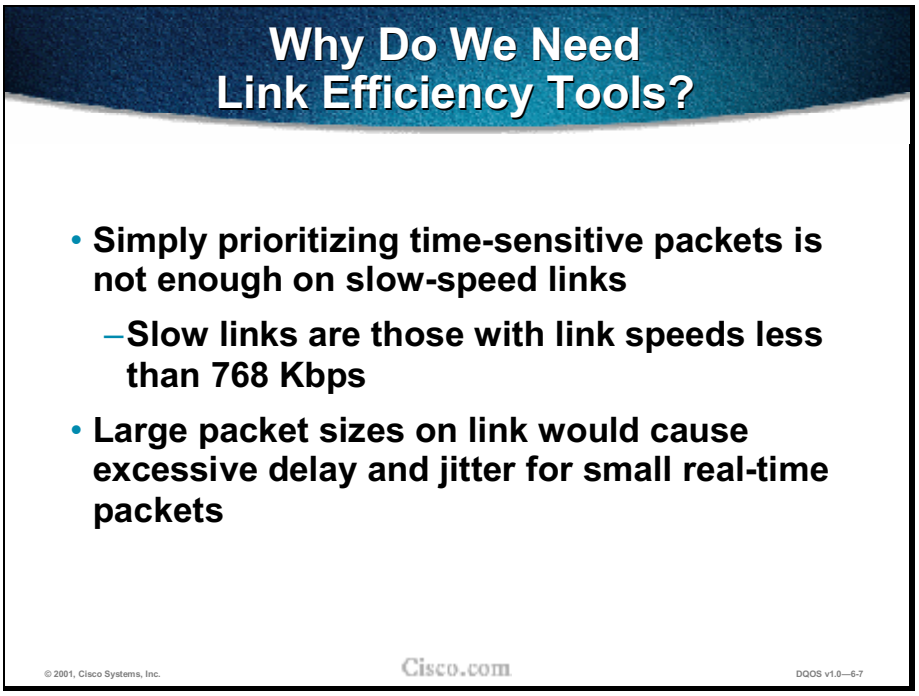
Serialization Delay

Serialization delay is the time that it takes to place bits on a circuit

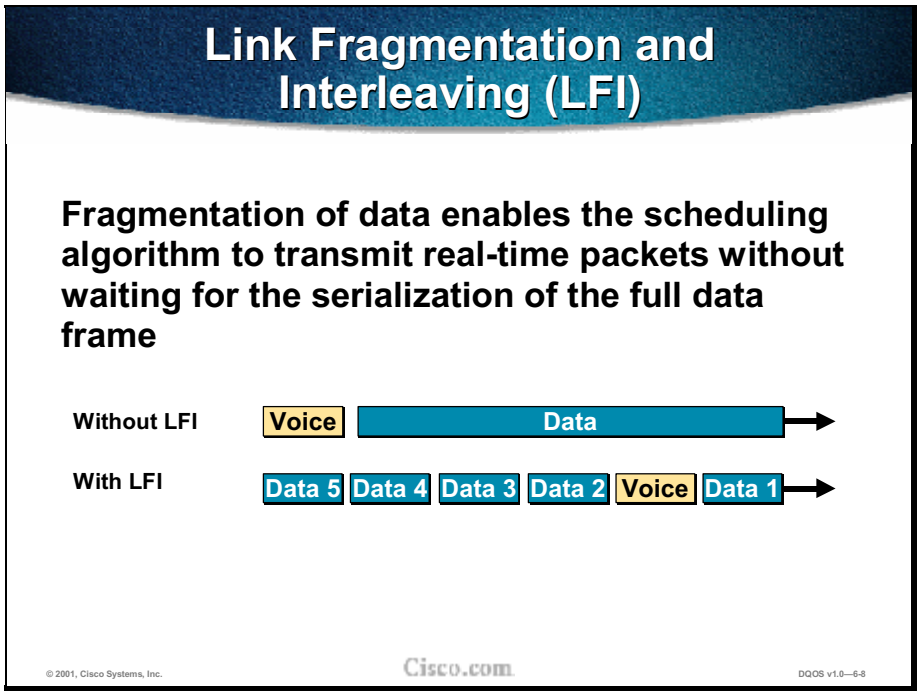
- **Function of Link Speed and Packet Size**

Link Speed	Packet Size					
	64 bytes	128 bytes	256 bytes	512 bytes	1024 bytes	1500 bytes
56 Kbps	9 ms	18 ms	36 ms	72 ms	144 ms	214 ms
64 Kbps	8 ms	16 ms	32 ms	64 ms	128 ms	187 ms
128 Kbps	4 ms	8 ms	16 ms	32 ms	64 ms	93 ms
256 Kpbs	2 ms	4 ms	8 ms	16 ms	32 ms	46 ms
512 Kbps	1 ms	2 ms	4 ms	8 ms	16 ms	23 ms
768 Kbps	0.640	1.28 ms	2.56 ms	5.12 ms	10.24 ms	15 ms

The higher the circuit speed, the less the serialization delay. For example, it takes about 5 microseconds to send a 1024-byte packet over a 1.544-Mbps line. The type of compression being used has an important effect on serialization delay. For example, if voice data is compressed into 8-kbps streams (using G.729), the resulting delay should be about 20 microseconds.

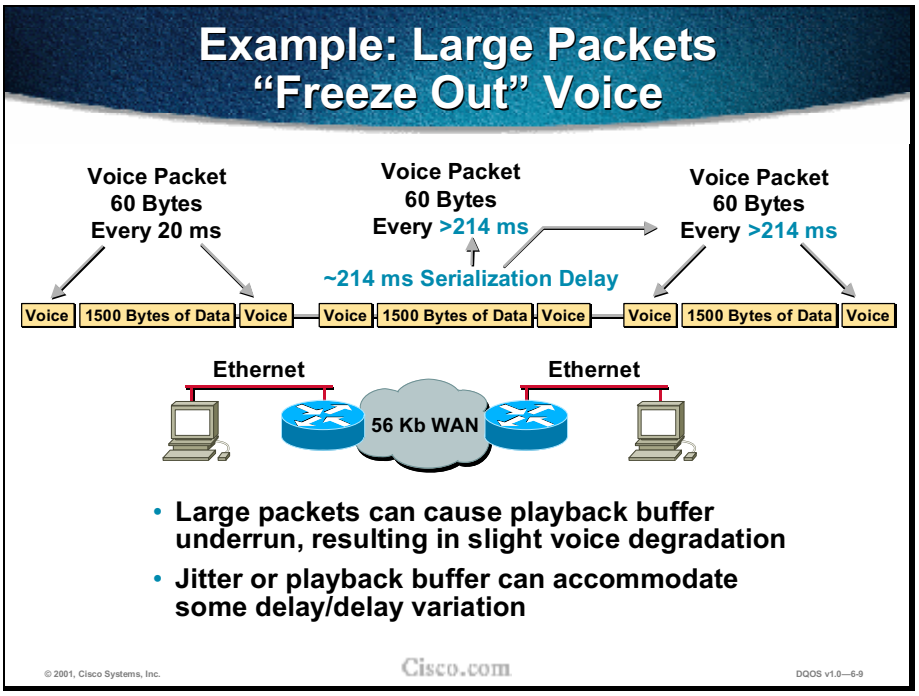


Unacceptable queuing delays for small real-time packets exist regardless of use of QoS features such as Resource Reservation Protocol (RSVP) and weighted fair queuing (WFQ) and use of voice compression algorithms such as code excited linear prediction compression (CELP), which reduces the inherent bit rate from 64 kbps to as low as 8 kbps. Despite these measures, real-time delay continues to exist because per-packet header overhead is too large, and large maximum transmission units (MTUs) are needed to produce acceptable bulk transmission efficiency.



Link fragmentation and interleaving (LFI) is used on slow links to ensure that voice packets get sent without a lot of delay. While a data packet is being serialized (sent bit by bit on the serial link), the voice packet must wait. Large data packets can mean a relatively long wait on a slow WAN link. By fragmenting large data packets, LFI allows the voice packet to be sent sooner, reducing overall latency.

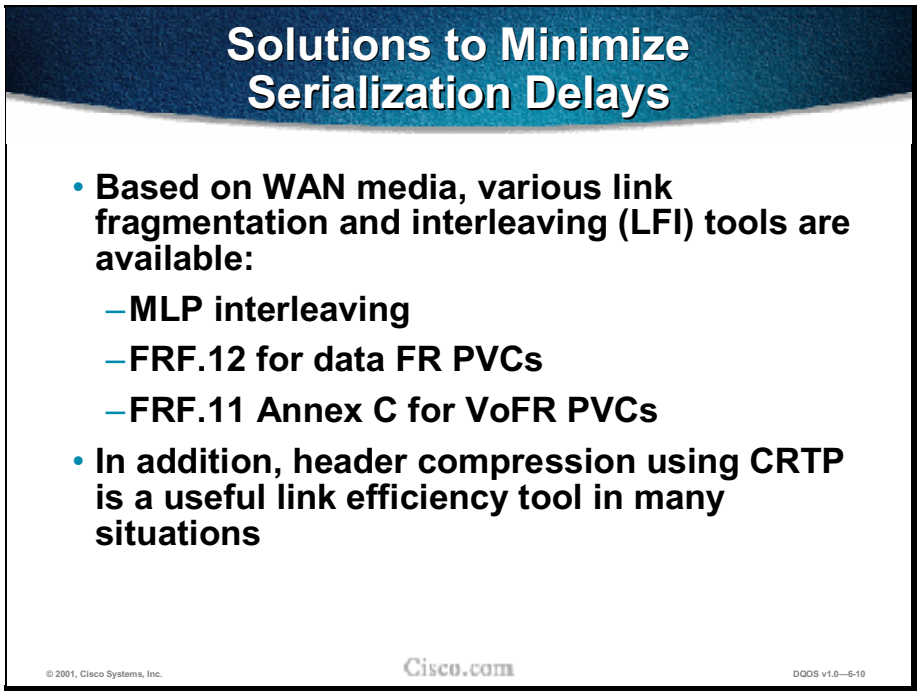
LFI is dependent on the media type. It is possible on multilink PPP (MP) links using LFI for MLP.



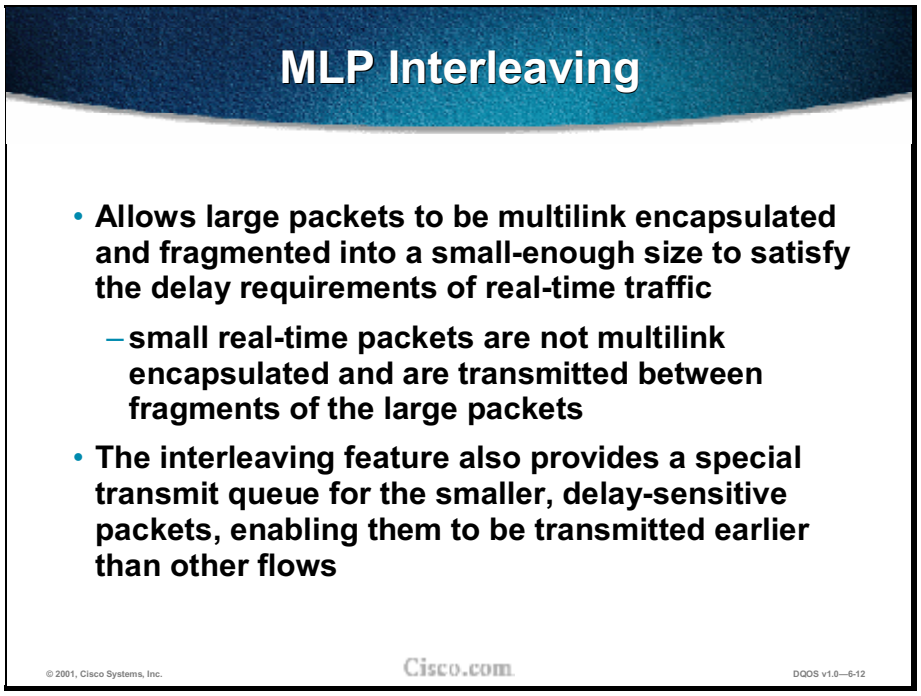
Delay is the time it takes for a packet to reach the receiving endpoint after being transmitted from the sending endpoint. This time is termed the end-to-end delay, and it consists of two components: fixed network delay and variable network delay.

Jitter is the delta, or difference, in the total end-to-end delay values of two voice packets in the voice flow. To compensate for these delay variations between voice packets in a conversation, VoIP endpoints use jitter buffers to turn the delay variations into a constant value so that voice can be played out smoothly.

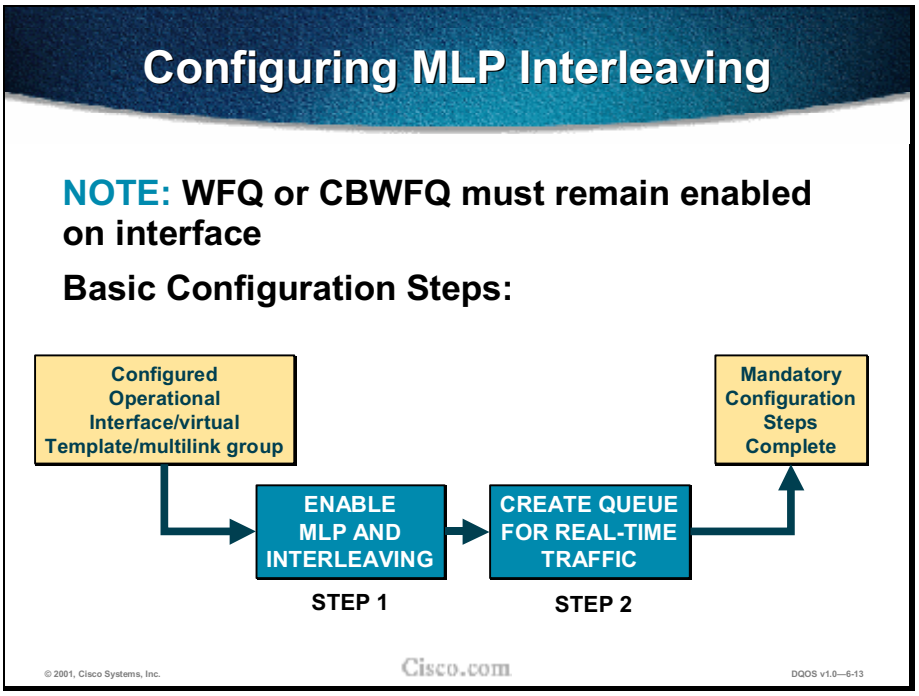
The G.114 standard states that a one-way delay budget of 150 ms is acceptable for high voice quality in an integrated network. A large MTU of 1500 bytes takes 215 milliseconds to traverse a 56-kbps line, which exceeds the delay target. Therefore, to limit the delay of real-time packets on relatively slow bandwidth links—links such as 56-kpbs Frame Relay or 64-kbps ISDN B channels—a method for fragmenting larger packets and queuing smaller packets between fragments of the large packet is needed.



Cisco IOS offers two link-layer efficiency mechanisms—LFI for Multilink Point-to-Point Protocol (MLP) and a CRTP header—that work with queuing and traffic shaping to improve the efficiency and predictability of the application service levels.



The Cisco IOS LFI feature uses Cisco's implementation of MLP, which supports the fragmentation and packet sequencing specifications in RFC 1717. LFI allows reserve queues to be set up so that Real-Time Protocol (RTP) streams can be mapped into a higher-priority queue in the configured weighted fair queue set.



Task Command

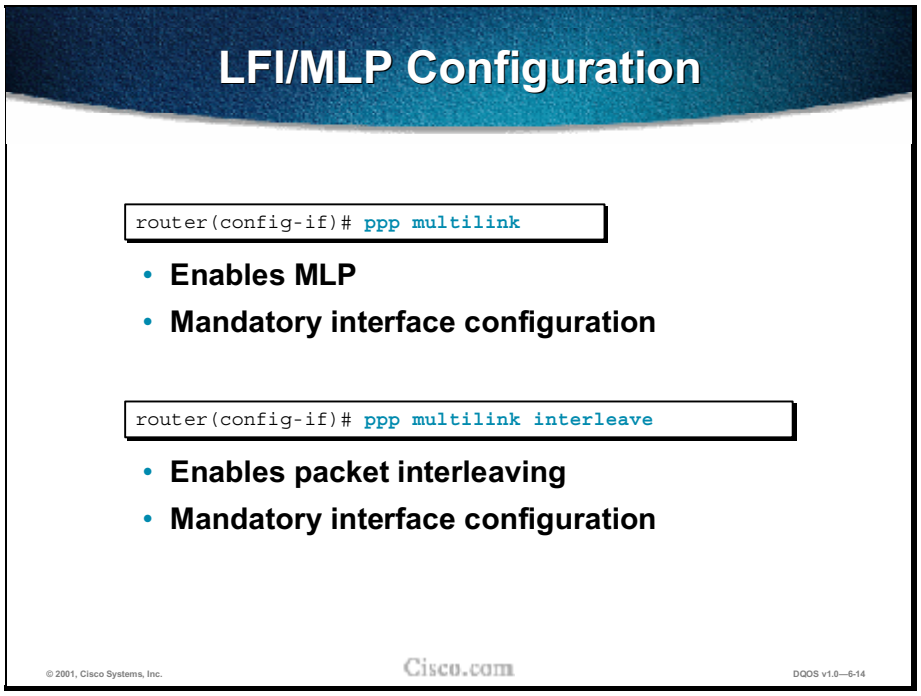
Step 1 Enable Multilink PPP. **ppp multilink**

Step 2 Enable real-time packet interleaving. **ppp multilink interleave**Optionally, configure a maximum fragment delay. **ppp multilink fragment-delay** *milliseconds*

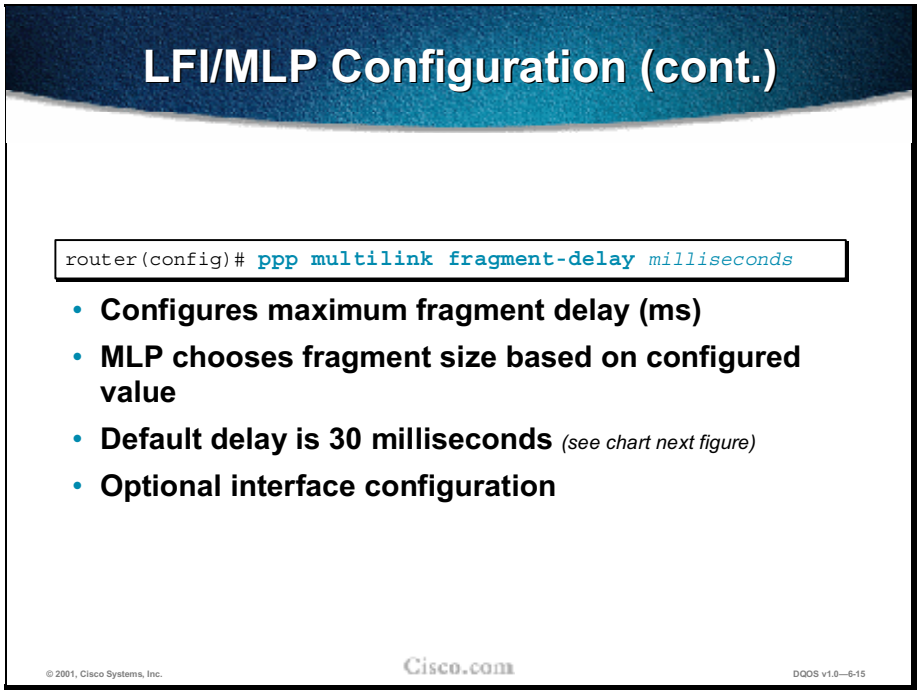
Step 4 Reserve a special queue for real-time packet flows to specified destination UDP ports, allowing real-time traffic to have higher priority than other flows.

ip rtp reserve *lowest-UDP-port range-of-ports [maximum-bandwidth]*

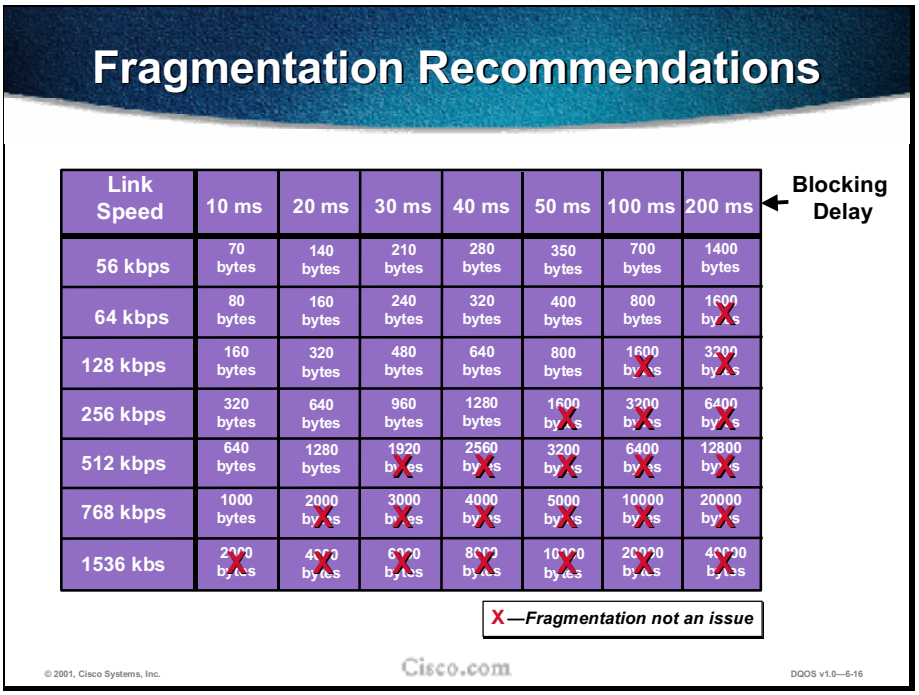
Step 5 or virtual templates only, apply the virtual template to the multilink bundle.1**multilink virtual-template**



As discussed previously, Multilink PPP allows packets to be fragmented and the fragments to be sent at the same time over multiple point-to-point links to the same remote address. Once MLP has been enabled on an interface, the interleave function must be individually specified.



Fragment delay values are a function of the overall delay budget analysis. As mentioned previously, when real-time traffic is involved, a one-way delay of less than150 milliseconds is considered optimal for voice traffic. Fragment sizes map to serialization delay.

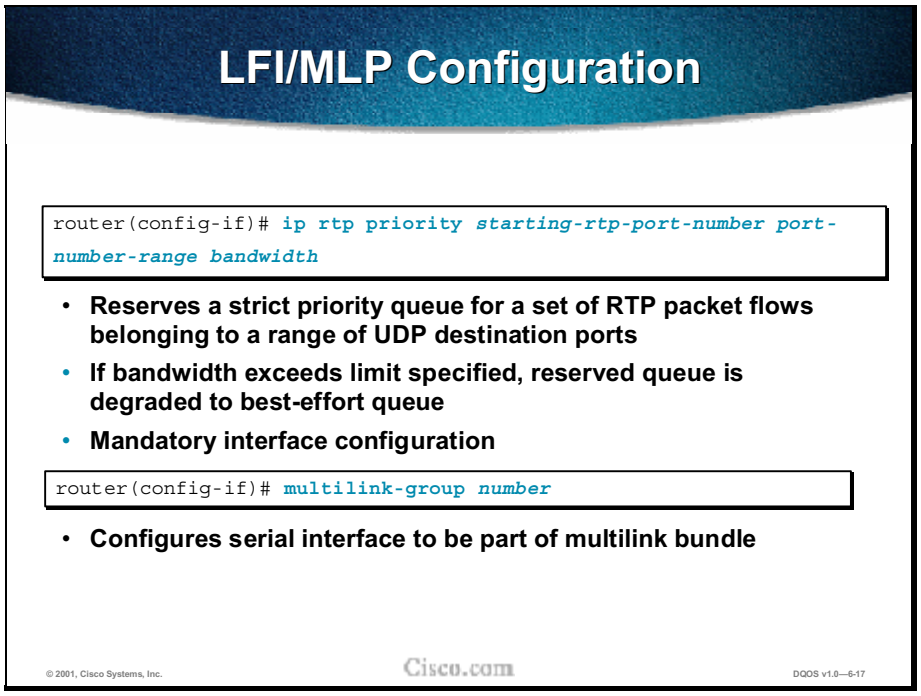


A 10-ms blocking delay is the recommended target to use for setting fragmentation size. To calculate the recommended fragment size, divide the recommended 10 ms of delay by one byte of traffic at the provisioned line clocking speed, as follows:

- Fragment_Size = (Max_Allowed_Jitter * Link_Speed_in_kbps) / 8

For example:

- Fragment_Size = (10 ms * 56) / 8 = 70 bytes

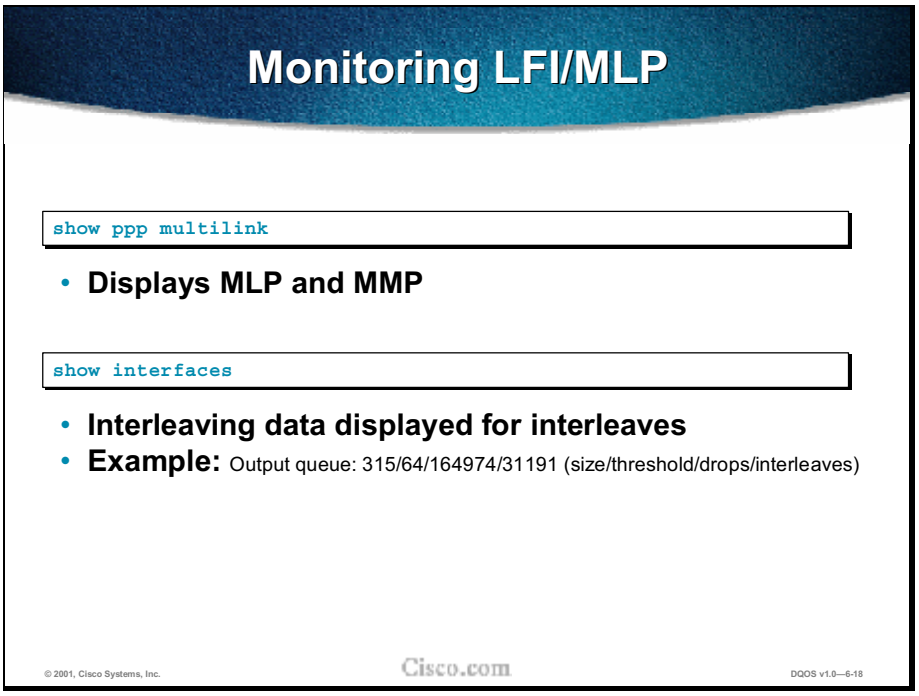


ip rtp reserve lowest-UDP-port range-of-ports [maximum-bandwidth]

Reserve a special queue for real-time packet flows to specified destination User Datagram Protocol (UDP) ports, allowing real-time traffic to have higher priority than other flows. If the bandwidth exceeds the limit specified, the reserved queue is degraded to a best-effort queue. (Use of this command is helpful in improving delay bounds of real-time traffic, such as voice streams, by giving them a higher priority.)

multilink virtual-template

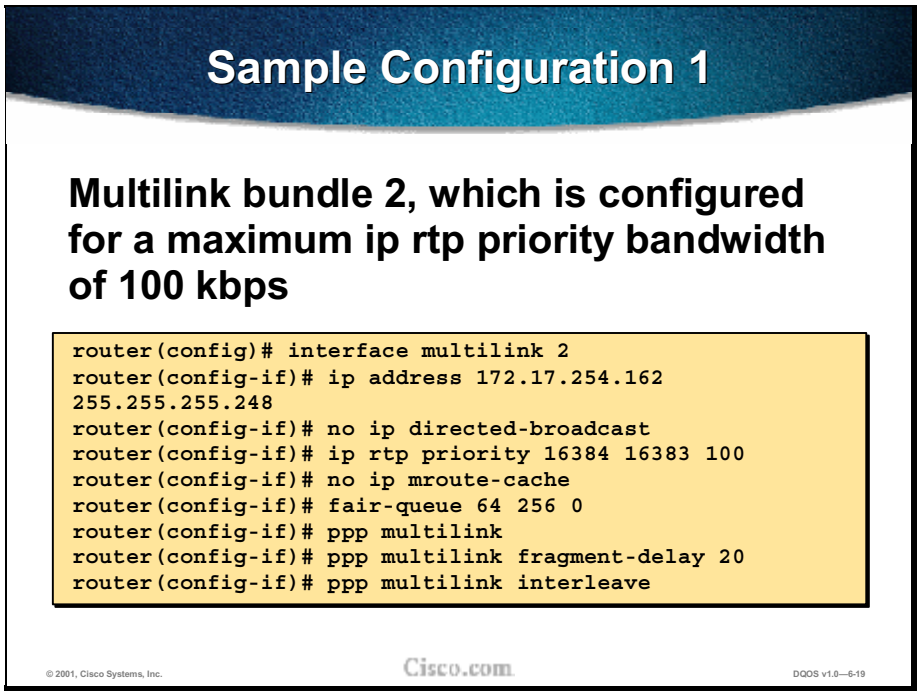
For virtual interface templates only, apply the virtual interface template to the multilink bundle. (This step is not used for ISDN or dialer interfaces.)



show ppp multilink

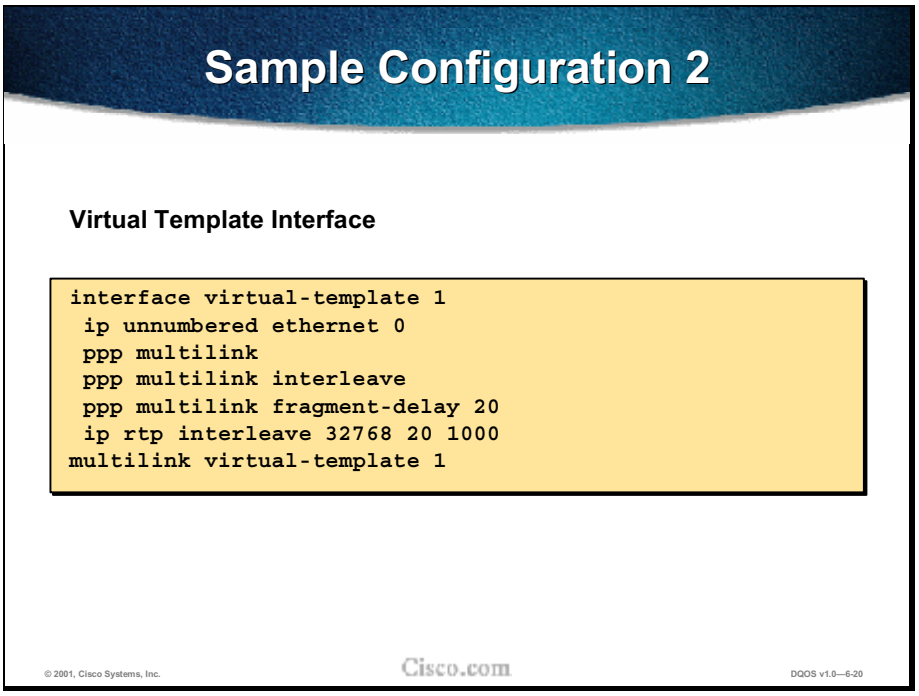
Display MLP and multilink bundle information.

Interleaving statistics can be displayed by using the **show interfaces** command, specifying the particular interface on which interleaving is enabled. Interleaving data is displayed only if there are interleaves.

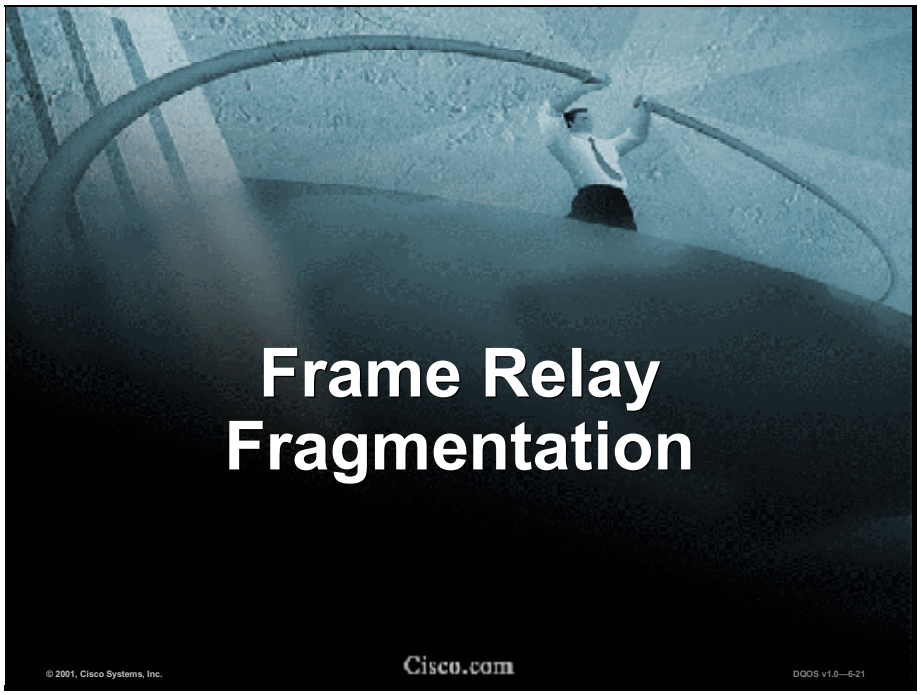


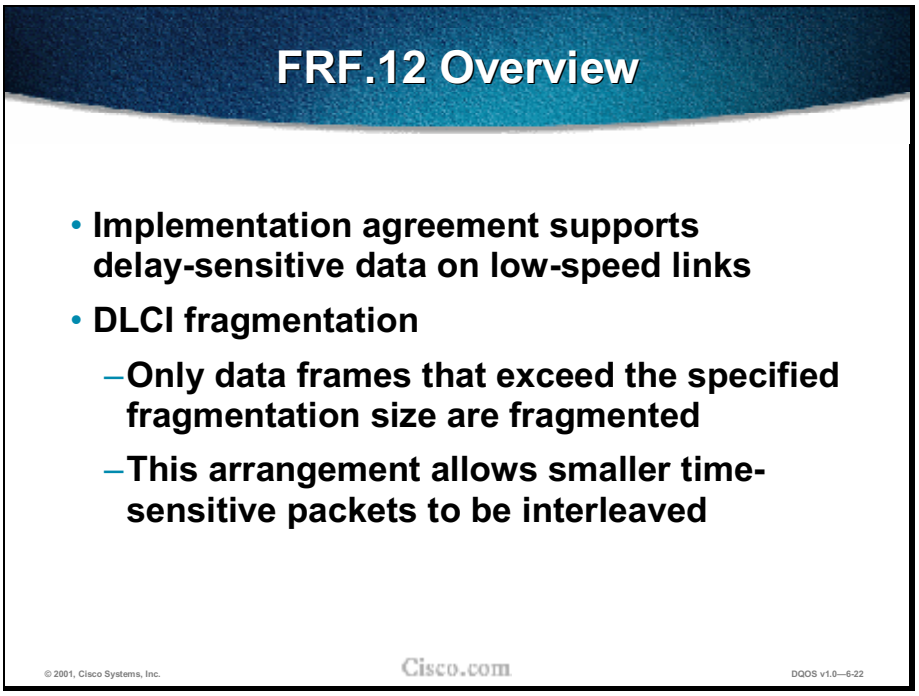
Physical interfaces have static, fixed configurations. Virtual access interfaces, however, are created dynamically on demand (the various uses are discussed in the next section of this document). They are also freed when they are no longer needed. Hence, the source of configuration of virtual access interfaces must be anchored by other means.

These various methods by which a virtual access gains its configuration are via the virtual template interface and/or RADIUS and TACAC+ records that reside on an authentication server. The latter method is called per-user virtual profiles. Because virtual access interfaces can be configured using a global virtual template, virtual access interfaces for various users can inherit identical configurations from one virtual template interface. For example, the network administrator may choose to define a common PPP authentication method (CHAP) for all virtual access users of the system. For specific per-user tailored configurations, the network administrator may define interface configurations—such as PAP authentication—specific to the user in the virtual profile. In short, the general-to-specific configuration scheme available to the virtual access interfaces allows the network administrator to tailor interface configurations common to all users and/or to tailor individually to the user.



The above example defines a virtual template interface that configures MLP interleaving and a maximum real-time traffic delay of 20 ms, and then applies that virtual template to the MLP bundle.

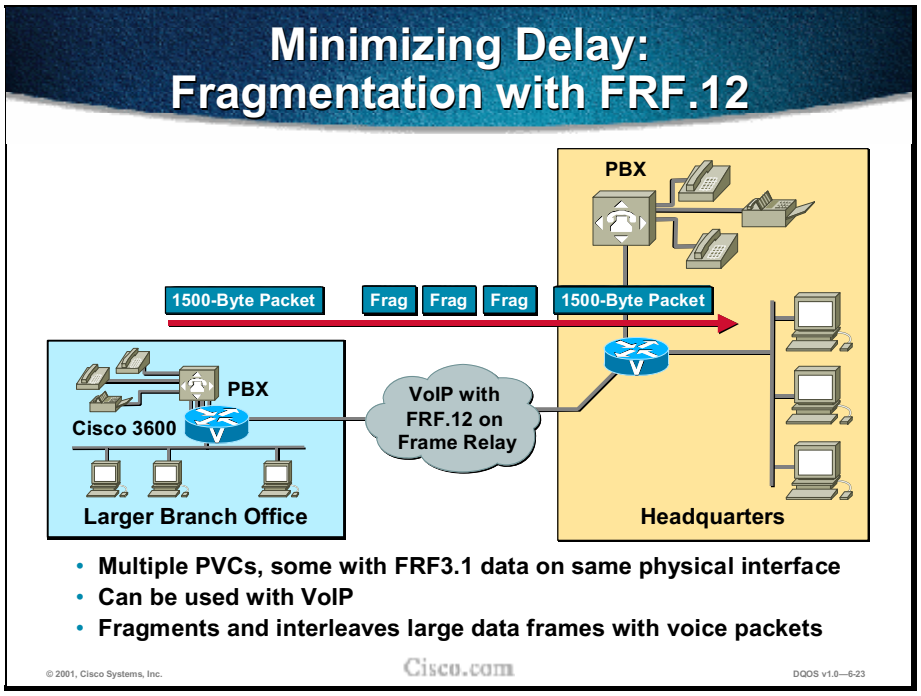

Frame Relay
Fragmentation
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—6-21



FRF.12 (Frame Relay fragmentation) is an implementation agreement defined to support voice and other real-time delay-sensitive data on low-speed links. It accommodates variations in frame sizes in a manner that allows a mixture of real-time and non-real-time data.

FRF.12 stipulates that when fragmentation is turned on for a data-link connection identifier (DLCI), only data frames that exceed the specified fragmentation size are fragmented. This arrangement allows small Voice over IP (VoIP) packets, which are not fragmented because of their size, to be interleaved as frames between large data packets that have been fragmented into smaller frames. This improves the serialization delay for packets leaving the router and prevents the voice packets from waiting on large data packets to be processed.

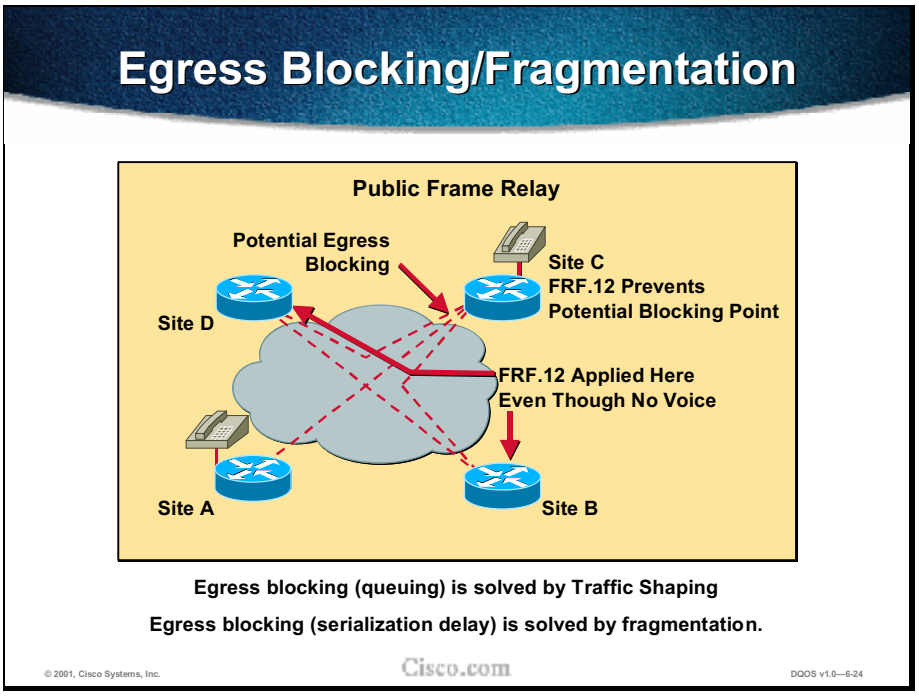
In a VoIP implementation, Frame Relay (Layer 2 protocol) cannot distinguish between VoIP and data frames. FRF.12 fragments all packets larger than the fragment size setting.



FRF.12 has no knowledge of what is in the data frame; it simply fragments any frame on the interface larger than a (generally) configurable packet size.

FRF.12 is not used to fragment voice frames (FRF.11), rather it fragments the non-FRF.11 that exceed the configured threshold.

FRF.12 alone is not enough; some form of queuing is needed to interweave the voice frames with the data frame fragments. FRF.12 is transparent to the data; it fragments based only on the packet size. Therefore it works well when Voice over IP is the encapsulation method (versus Voice over Frame Relay).

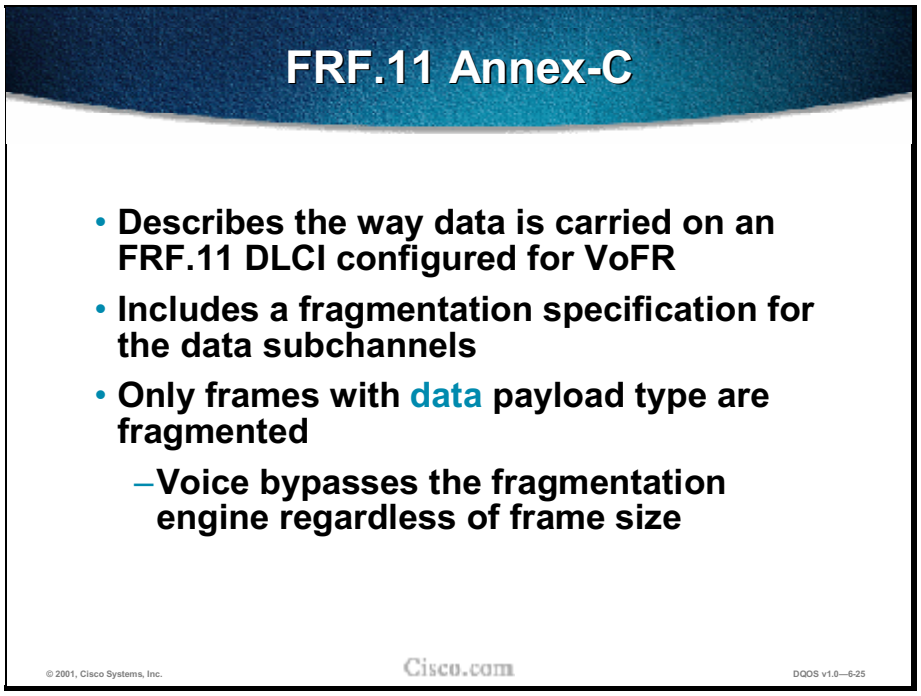


This network illustrates one of the challenges of designing Voice over Frame Relay networks.

If residency of a large data frame can block a voice frame from being placed on the link at the ingress to the network, what will happen if large data frames from nonvoice sites are in the queue ahead of voice frames being placed on low-speed links at the egress of the Frame Relay network to the destination CPE?

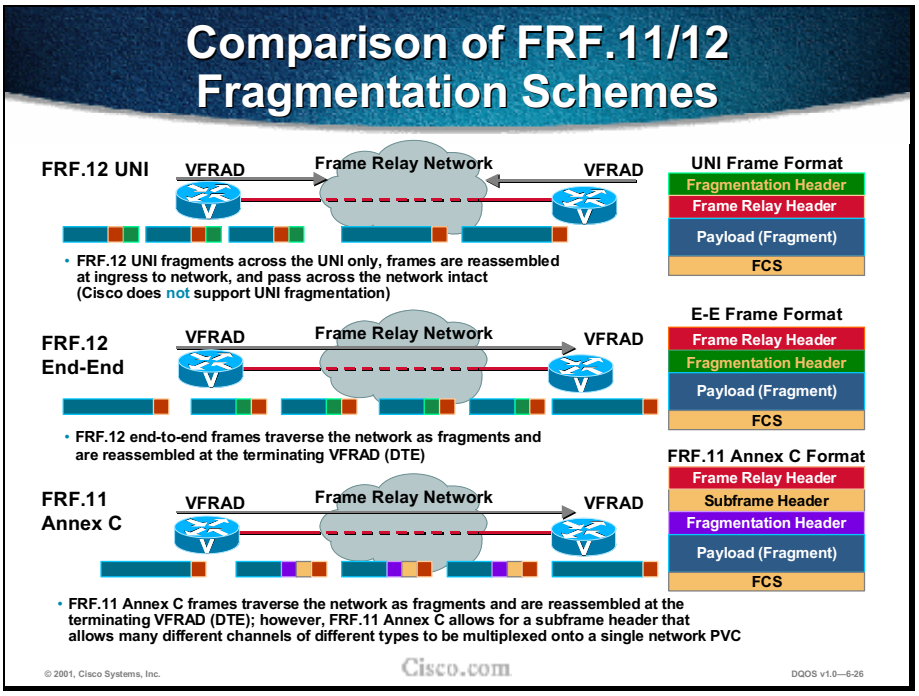
This problem is frequently referred to as egress blocking. Fragmenting nonvoice-enabled sites does not solve the problem entirely, but it does allow voice frames to be queued and transmitted in between data fragments from a large data frame.

Voice frames can still be trapped in a queue behind a large number of small data fragments because multiple sites are transmitting data concurrently and there is a higher speed of the backbone trunks, relative to the egress link speed. This is especially true if nonvoice sites are bursting to the voice egress. Priority queuing from the service provider can be effective in helping to solve this problem.



FRF.11 Annex C fragmentation describes the way data is carried on an FRF.11 DLCI (configured for VoFR). FRF.11 Annex C includes a fragmentation specification for the data subchannels.

Only frames with data payload type are fragmented. Frame Relay distinguishes voice frames from non-real-time data frames because the FRF.11 payload specifies the traffic type. Therefore, regardless of the voice frame size, it bypasses the fragmentation engine.



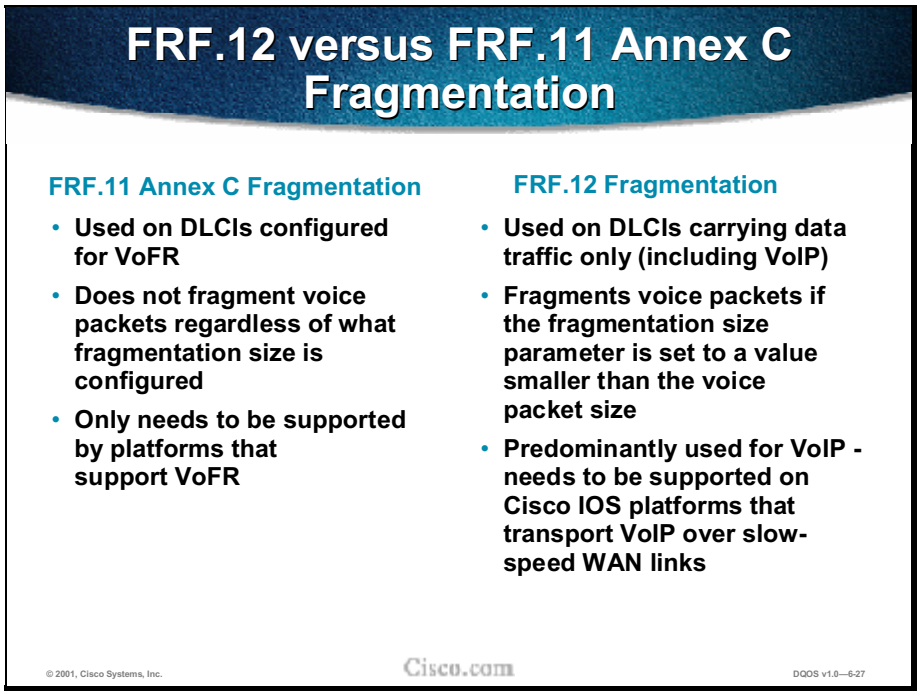
There are two forms of FRF.12: a User-Network Interface (UNI) form and an end-to-end form.

In the UNI form the network actually participates in the FRF.12 procedure. The Frame Relay DTE fragments data packets exceeding the threshold and interweaves voice packets. The Frame Relay DCE, the network node, receives the fragmented data segments with the voice frames in between. In the UNI procedure, the Frame Relay DCE network node holds the first data fragment and reassembles the data frame to be forwarded across the network. Voice frames (or any other frames received in between smaller than the fragmentation threshold) are forwarded ahead of the reassembled data frame. Please note that this is a special case where it is actually desirable for the network to forward frames out of order or in priority order.

In the UNI form, the fragmentation header is *before* the Frame Relay header and is stripped before the frame is forwarded across the network.

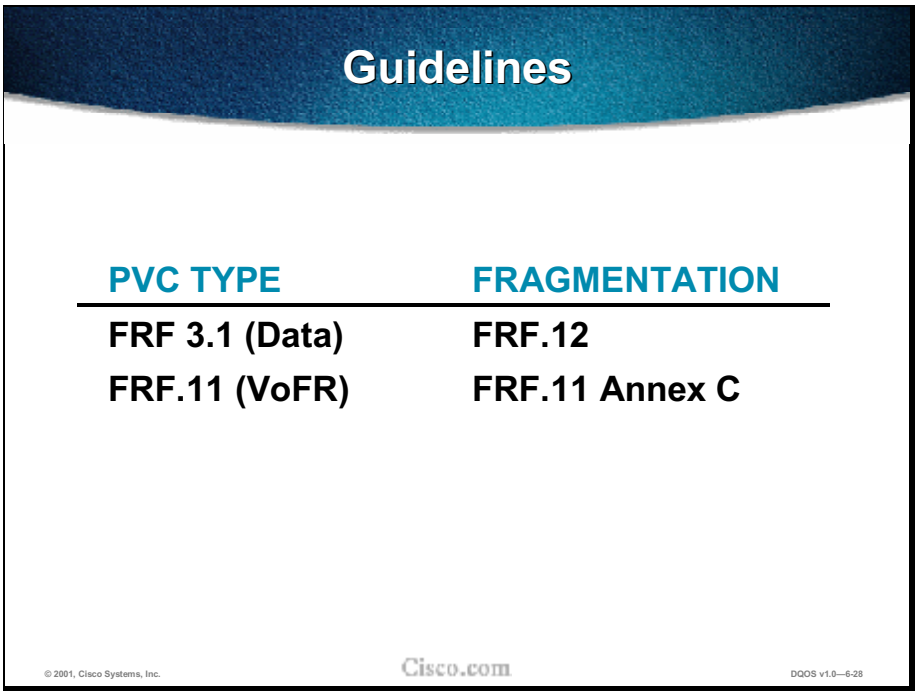
In the end-to-end procedure, the network is not aware that the data packet has been fragmented. The fragments from the data packet and the interwoven voice frames are handled transparently by the network nodes. The remote Frame Relay DTE receives and holds the first data fragment until the remaining data fragments arrive. Voice frames (or interwoven data frames) are forwarded immediately.

In the end-to-end form, the fragmentation header is *inside* the Frame Relay header and is passed transparently by the network.

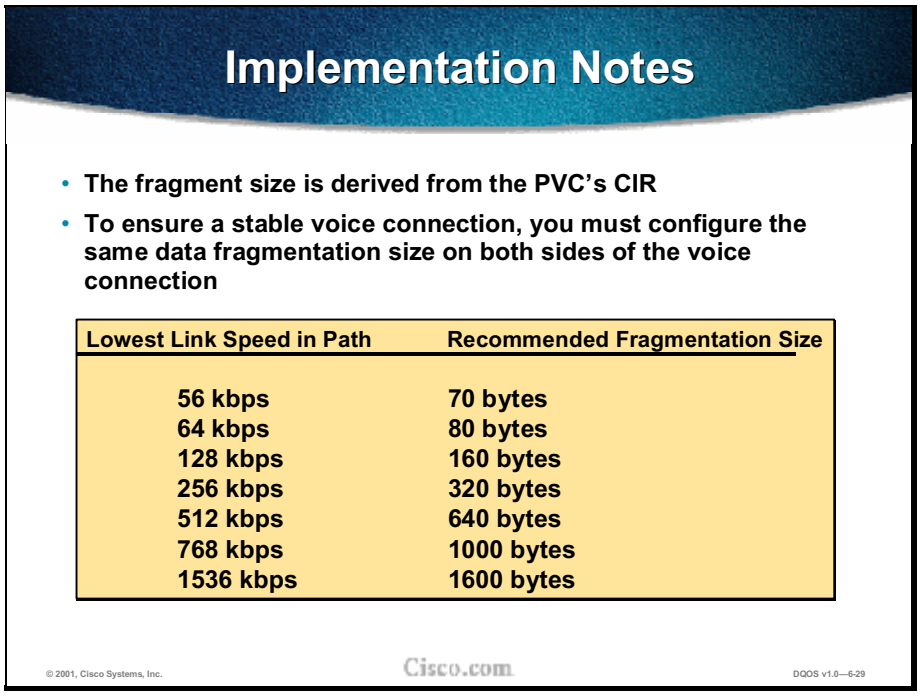


If a PVC is not configured for VoFR, it uses FRF.3.1 data encapsulation. If fragmentation is turned on for this DLCI, it uses FRF.12 for the fragmentation headers. PVCs carrying VoIP use FRF.12 fragmentation because VoIP is a Layer 3 technology that is transparent to Layer 2 Frame Relay. VoIP and VoFR can be supported on different PVCs on the same interface but not on the same PVC.

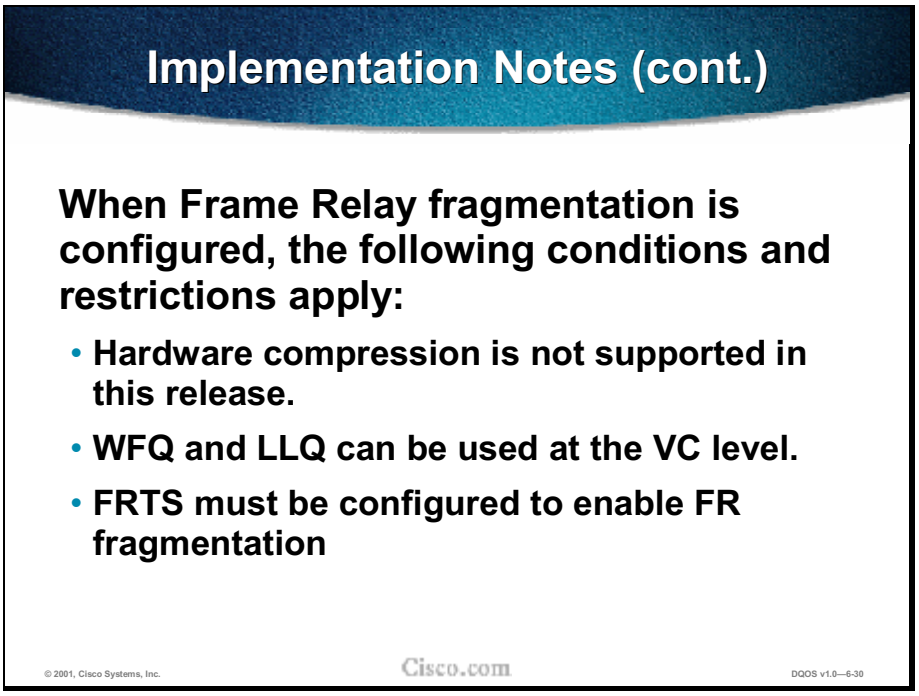
FRF.12 fragments voice packets if the fragmentation size parameter is set to a value smaller than the voice packet size. FRF.11 Annex C (VoFR) does not fragment voice packets, regardless of what fragmentation size is configured. FRF.11 Annex C needs to be supported only by platforms that support VoFR. Because FRF.12 is predominantly used for VoIP, it is important to support FRF.12 as a general feature on Cisco IOS platforms that transport VoIP over slow-speed WAN links.



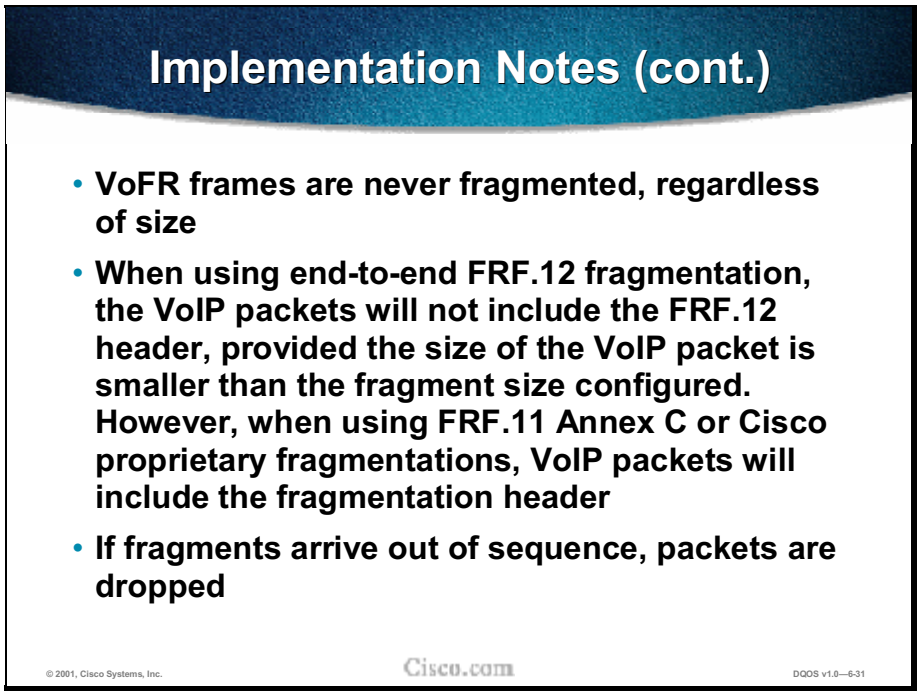
A Frame Relay PVC runs either FRF.12 or FRF.11—never both, since they are mutually exclusive. If the PVC is configured for VoFR, it uses FRF.11. If fragmentation is turned on for this PVC, it uses FRF.11 Annex C (or the Cisco derivative) for the fragmentation headers.



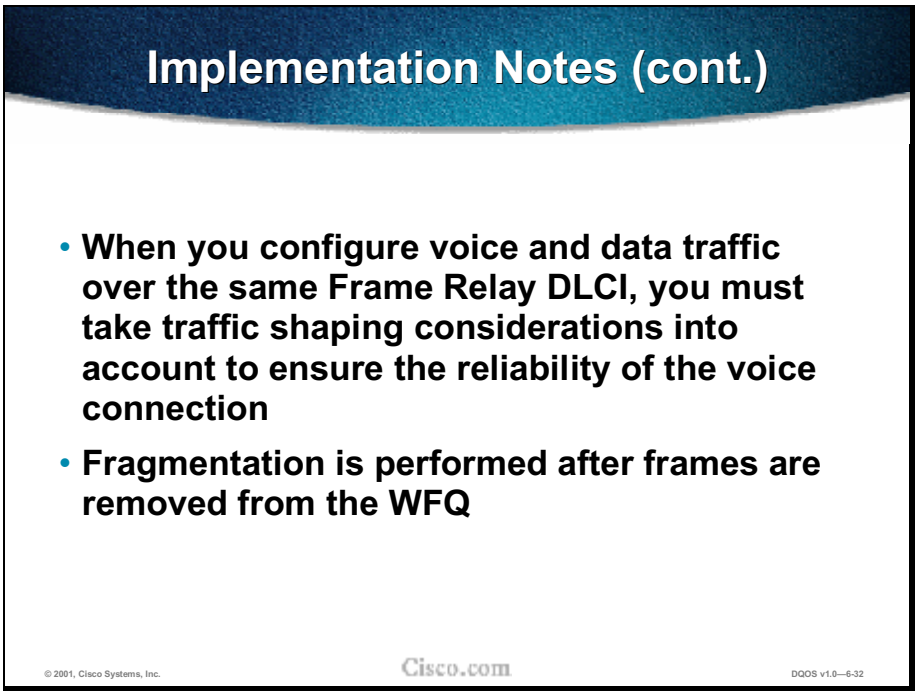
Set the fragment size so voice packets are not fragmented and do not experience a serialization delay greater than 20 milliseconds. Set the fragmentation size based on the lowest port speed between the routers. For example, if there is a hub-and-spoke Frame Relay topology where the hub has a T1 speed and the remote routers have 64-K port speeds, the fragmentation size should be set for the 64-K speed on both routers. Any other PVCs that share the same physical interface should configure the fragmentation to the size used by the voice PVC. Use the following chart to determine the fragmentation size values.



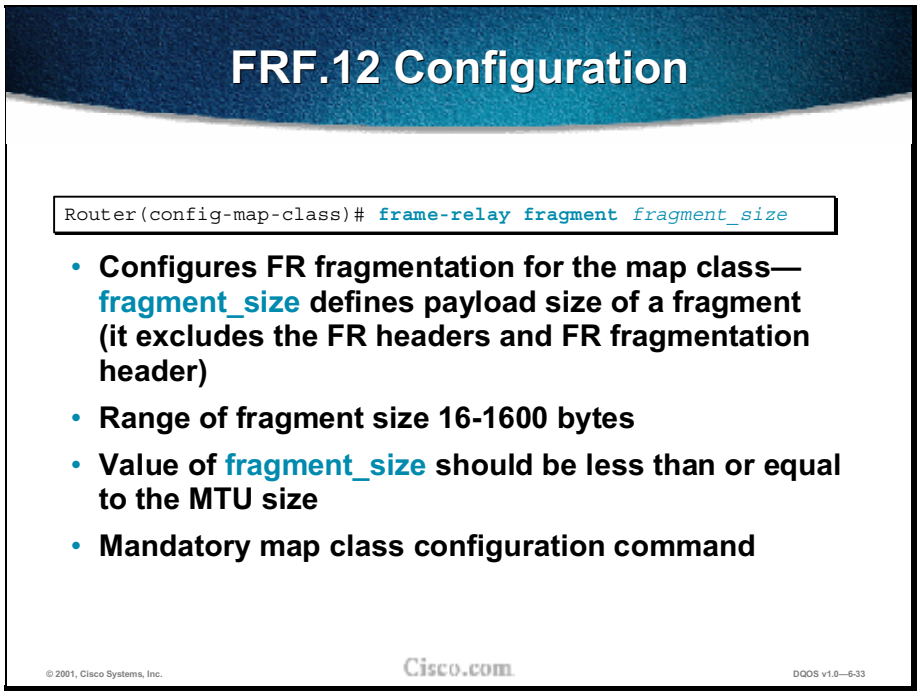
When deploying a solution using fragmentation and interleaving on a Frame Relay backbone, it is a good idea to be aware of the key issues highlighted in this figure.



Additional issues to be aware of are listed in the frame.

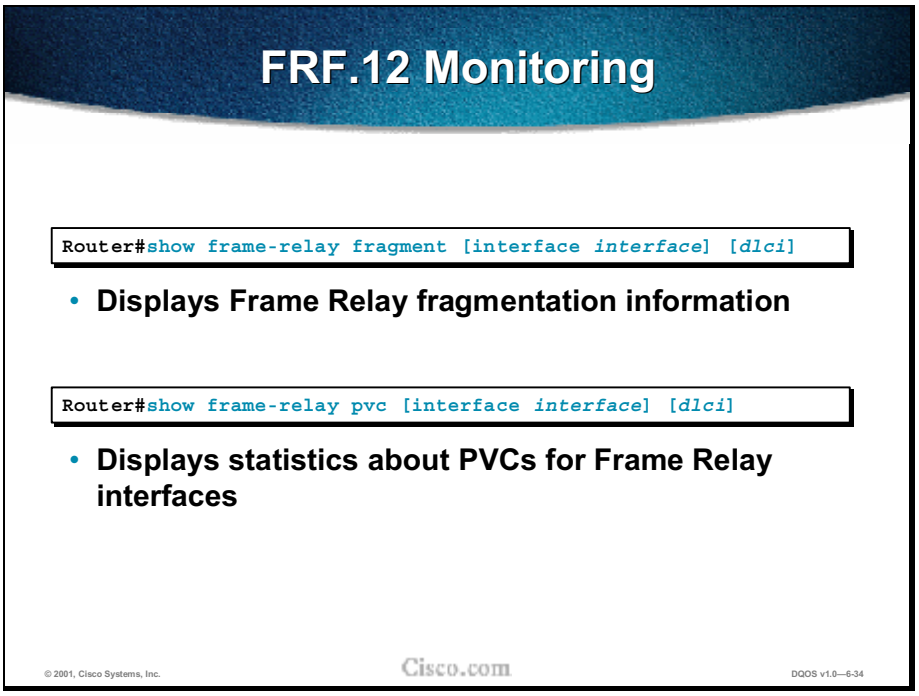


Do not exceed the CIR of the PVC. Because voice quality cannot tolerate much delay, any queuing of voice packets in the Frame Relay cloud must be minimized. When CIR is exceeded, depending on the provider and how congested the rest of the Frame Relay network is, packets will usually begin to be queued in the Frame Relay cloud. Because customers have many different Frame Relay providers and differing amounts of congestion across their sites, it is difficult to forecast what will work. Maintaining values at (or below) CIR on the PVCs that use voice has been proven to work consistently.



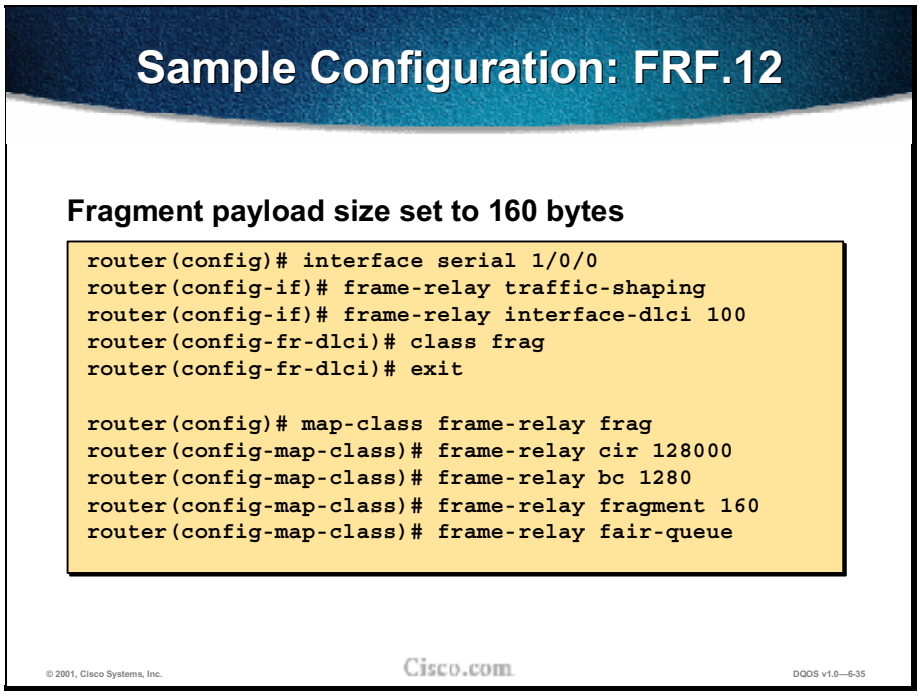
Configuration prerequisites:

- Frame Relay traffic shaping must be enabled on the interface.
- WFQ or low latency queuing (LLQ) must be configured on the PVC.

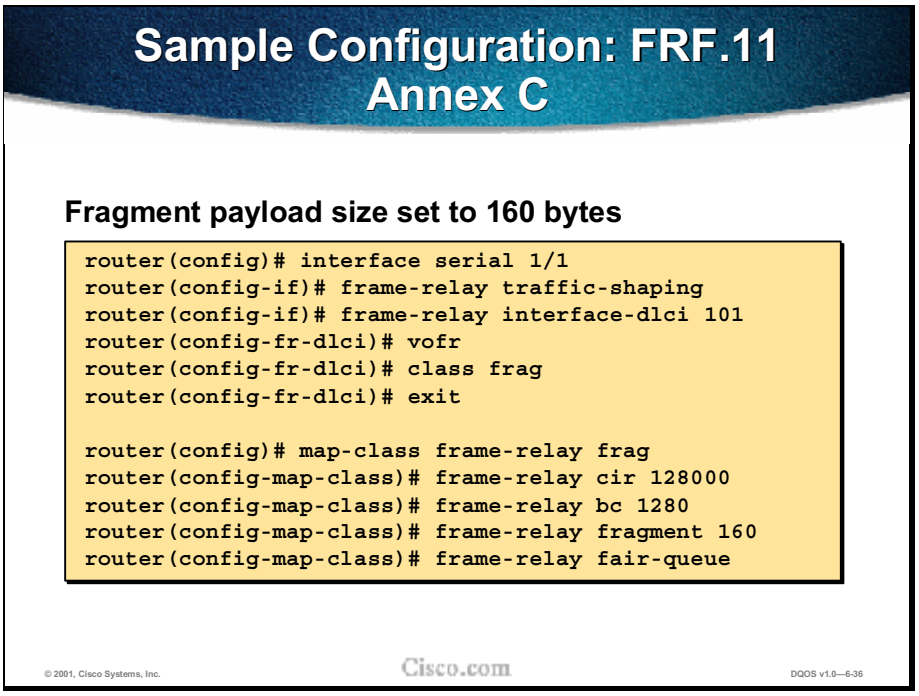


To display information about the Frame Relay fragmentation taking place in your Cisco router, enter the **show frame-relay fragment** command from privileged EXEC mode.

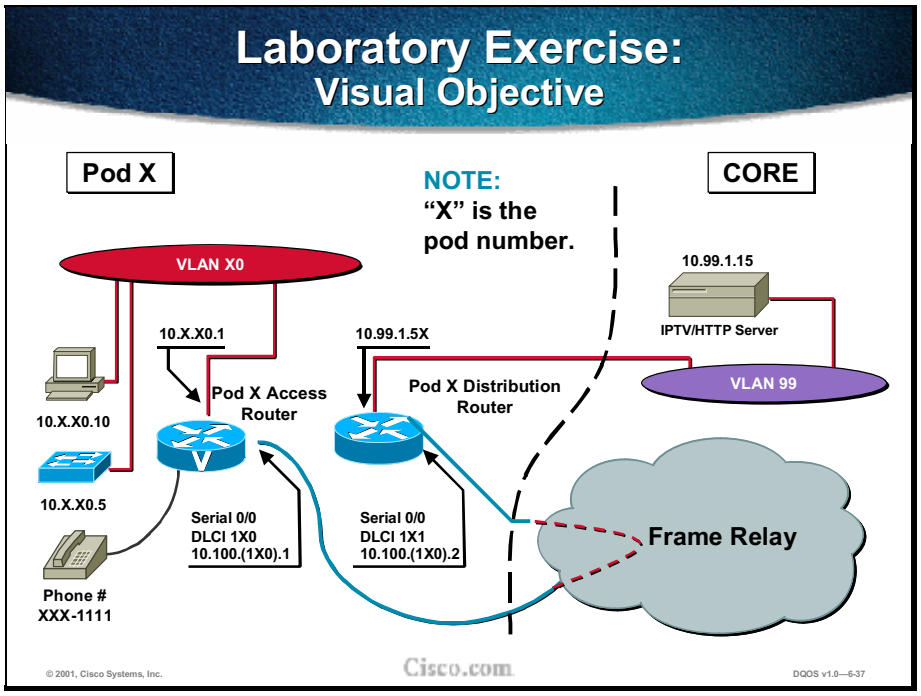
To display statistics about PVCs for Frame Relay interfaces, use the **show frame-relay pvc** command from privileged EXEC mode.

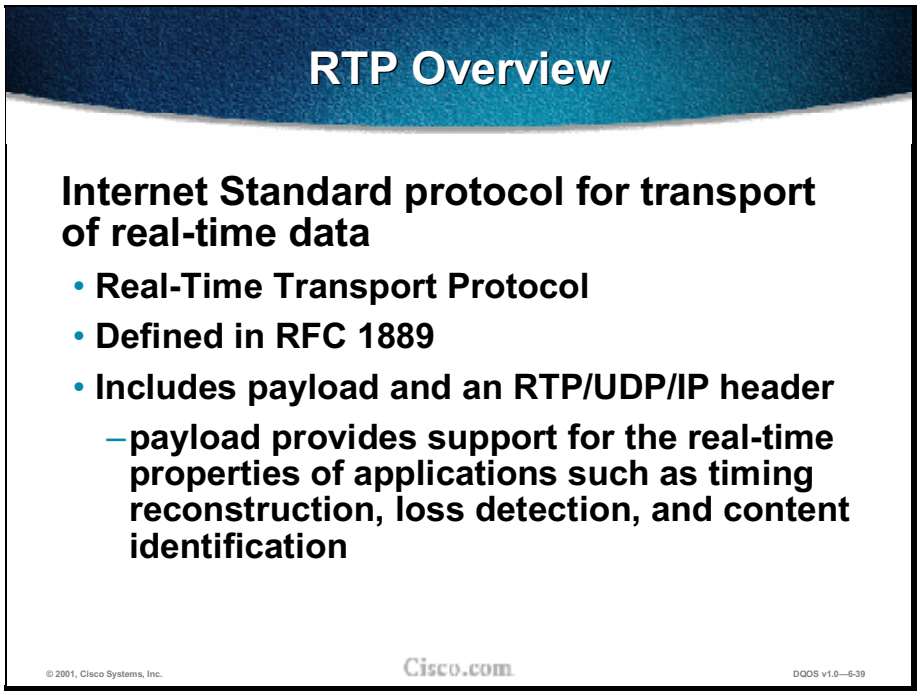


The above example shows how to enable pure end-to-end FRF.12 fragmentation for the *frag* map class. The fragment payload size is set to 40 bytes. Frame Relay traffic shaping is required on the PVC; the only queuing type supported on the PVC when fragmentation is configured is weighted fair queuing (WFQ).

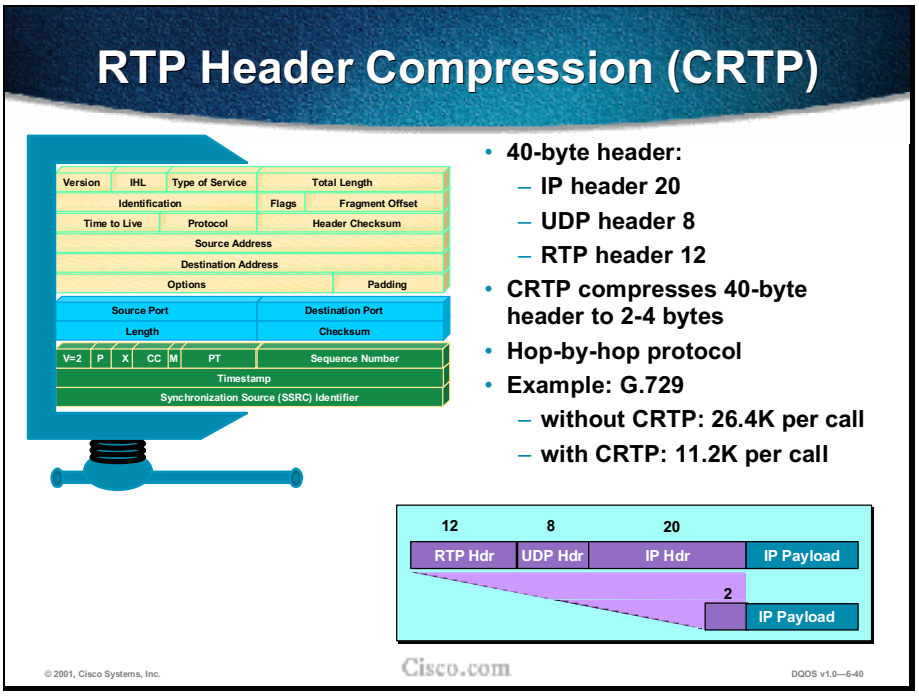


The above example shows how to enable FRF.11 Annex C fragmentation for data on a Cisco MC3810 PVC configured for VoFR. Note that fragmentation must be configured if a VoFR PVC is to carry data. The fragment payload size is set to 40 bytes. Frame Relay traffic shaping is required on the PVC; the only queuing type supported on the PVC when fragmentation is configured is weighted fair queuing (WFQ).


Laboratory Exercise:
Visual Objective
Pod X
CORE
NOTE:
"X" is the
pod number.
VLAN X0
10.99.1.15
IPTV/HTTP Server
10.X.X0.1
10.99.1.5X
Pod X Access
Router
Pod X Distribution
Router
VLAN 99
10.X.X0.10
10.X.X0.5
Serial 0/0
DLCI 1X0
10.100.(1X0).1
Serial 0/0
DLCI 1X1
10.100.(1X0).2
Frame Relay
Phone #
XXX-1111
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—6-37

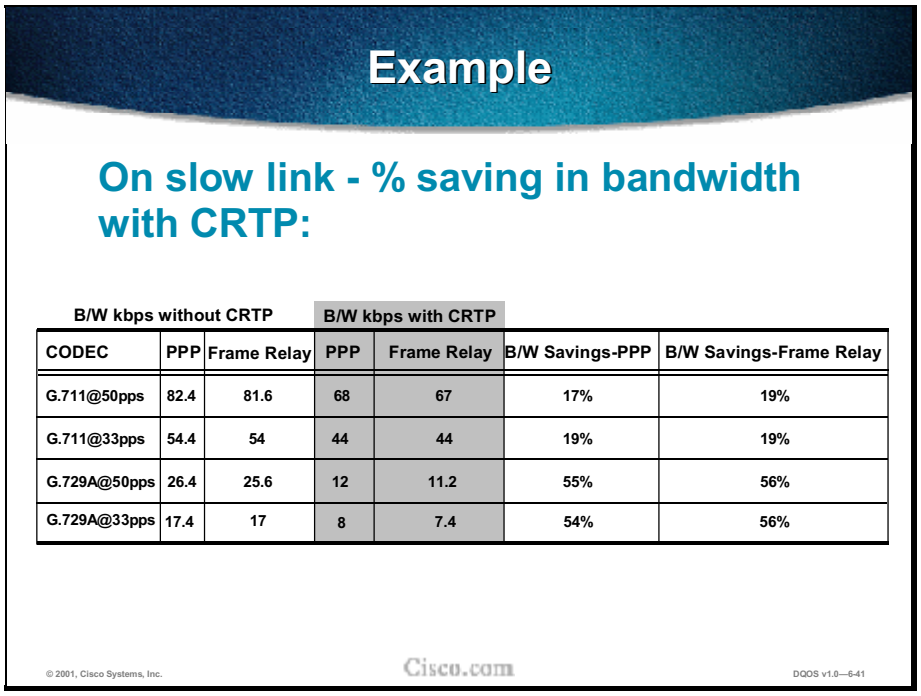


VoIP protocols typically use Real-Time Transport Protocol (RTP) for the media stream (the speech path). RTP uses UDP as its transport protocol. Voice signaling traffic often uses TCP as its transport medium. The IP layer provides routing and network-level addressing, while the link layer protocol controls and directs the transmission of the information over the physical medium.

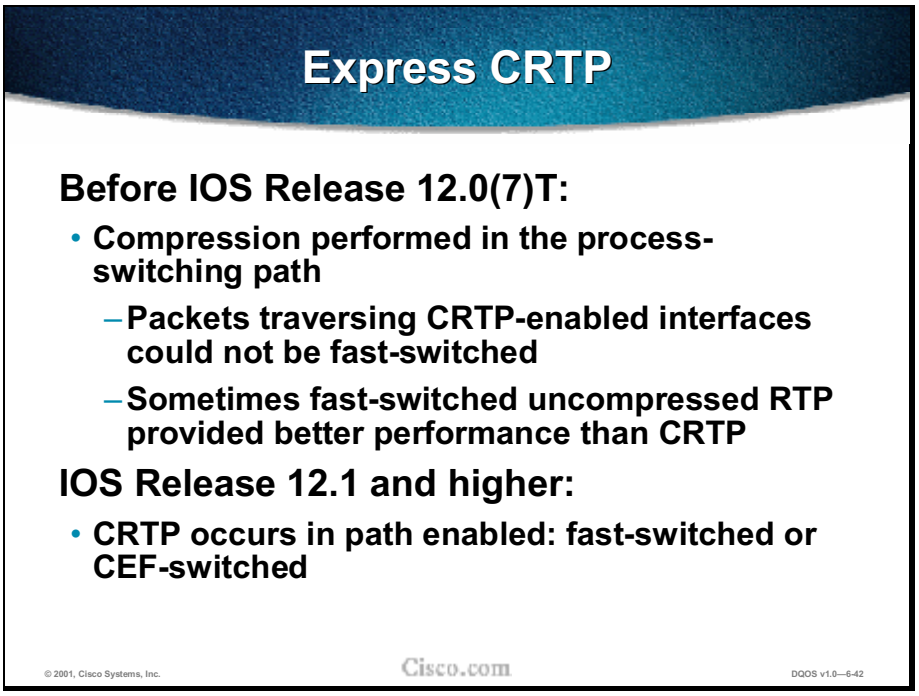


Compressed Real-Time Transport Protocol, or CRTP, is used on a link-by-link basis to compress the IP/UDP/RTP from 40 bytes to 2 to 4 bytes most of the time. In a packet voice environment when framing speech samples every 20 milliseconds, this scenario generates a payload of 20 bytes. The total packet size comprises an IP header (20 bytes), a UDP header (8 bytes), and an RTP header (12 bytes), combined with a payload of 20 bytes. It is evident that the size of the header is twice the size of the payload. When generating packets every 20 milliseconds on a slow link, the header consumes a large portion of the bandwidth.

Although fields in headers change in every packet, the difference from packet to packet is often constant. The decompressor can reconstruct the original header without any loss of information. CRTP compresses IP/RTP/UDP headers from 40 bytes to 2 to 5 bytes.

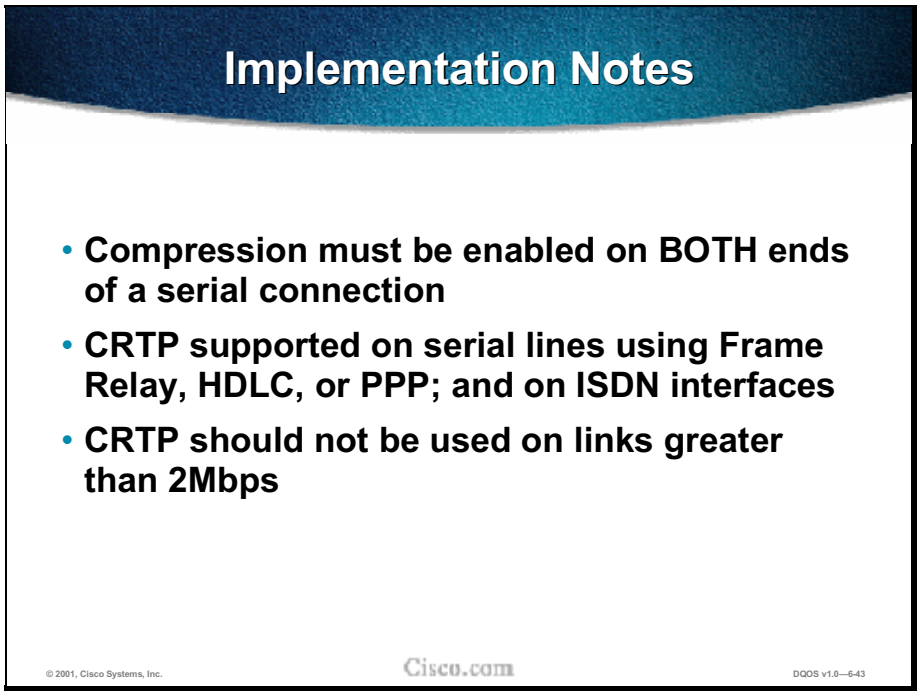


The table provides an idea of the bandwidth savings when CRTP is used as a link efficiency tool. With G.711 voice the voice payload is large enough to make the compression ratio relatively smaller. However, when compressed voice is used, the payload-to-IP header ratio becomes small enough to make the bandwidth savings significant.

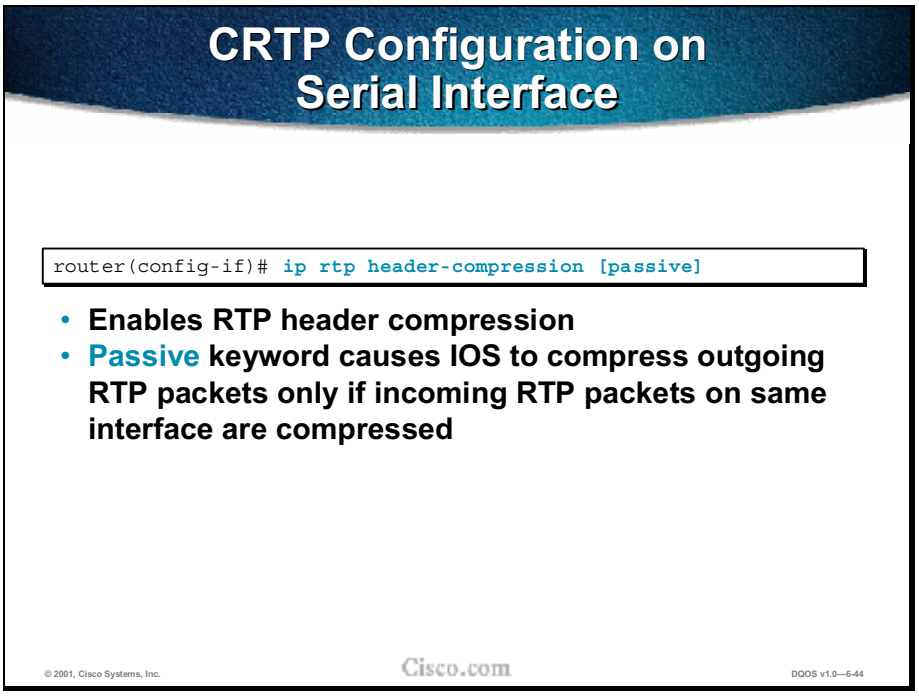


As of Cisco IOS Release 12.0(7)T, if TCP or RTP header compression is enabled, it occurs by default in the fast-switched path or the Cisco Express Forwarding-switched (CEF-switched) path, depending on which switching method is enabled on the interface. Furthermore, the number of TCP and RTP header compression connections is increased to 1,000 connections each.

Before this feature, such compression was performed in the process-switching path. That meant that packets traversing interfaces that had ::TCP or RTP:: header compression enabled were queued and passed up to the process to be switched. This procedure slowed down transmission of the packet, and therefore some users preferred to fast-switch uncompressed TCP and RTP packets.



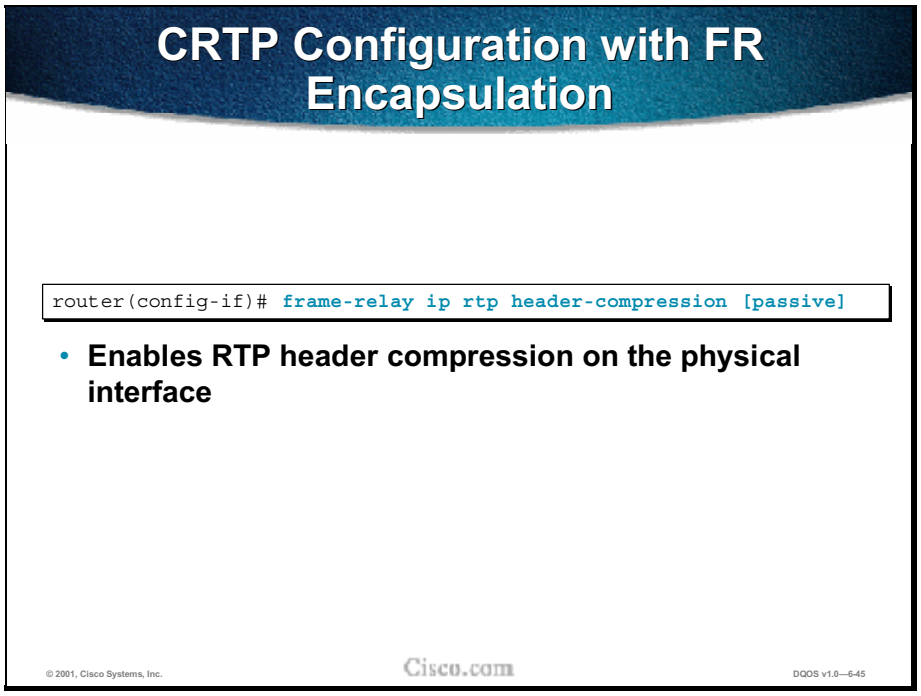
You should not use CRTP on any high-speed interfaces—that is, anything over 768 kbps speed—because the trade-off of increased processor overhead is not desirable.



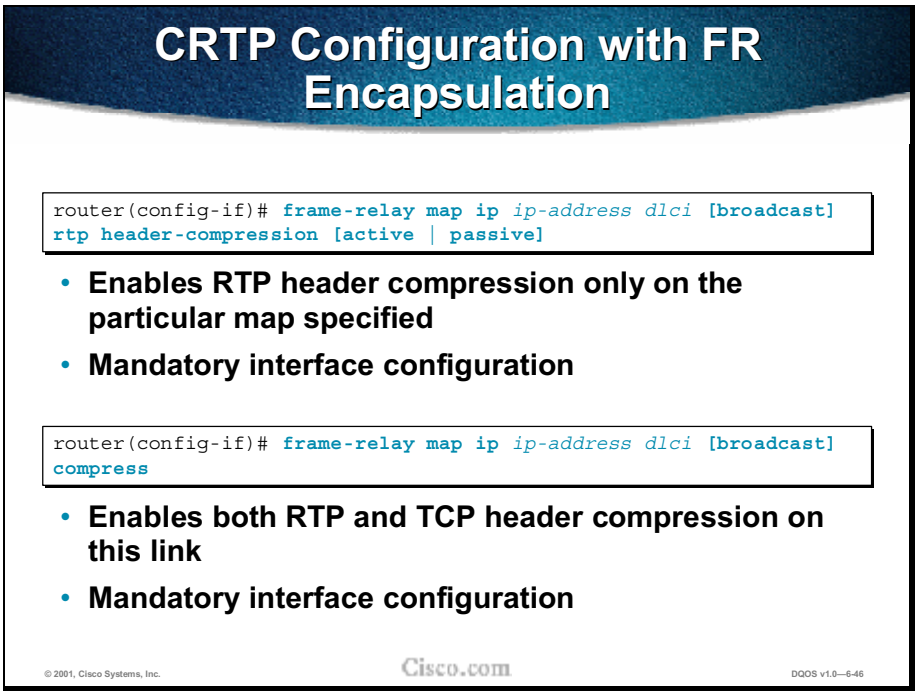
To enable CRTP header for serial encapsulations HDLC or PPP, use the following command in interface configuration mode (you must enable compression on both ends of a serial connection):

ip rtp header-compression [passive]—Enables RTP header compression

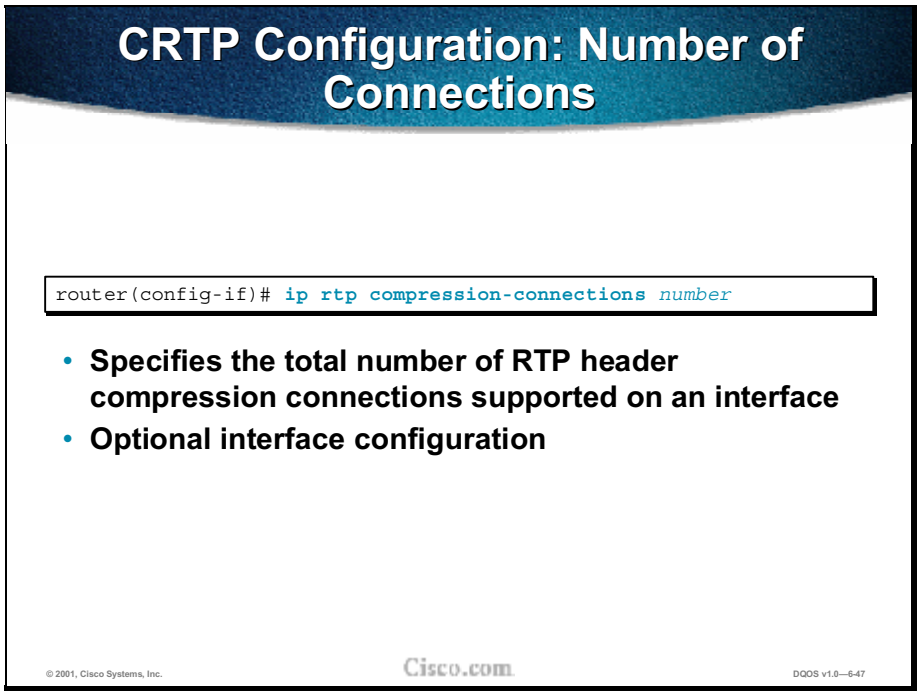
If you include the *passive* keyword, the software compresses outgoing RTP packets only if incoming RTP packets on the same interface are compressed. If you use the command without the *passive* keyword, the software compresses all RTP traffic.



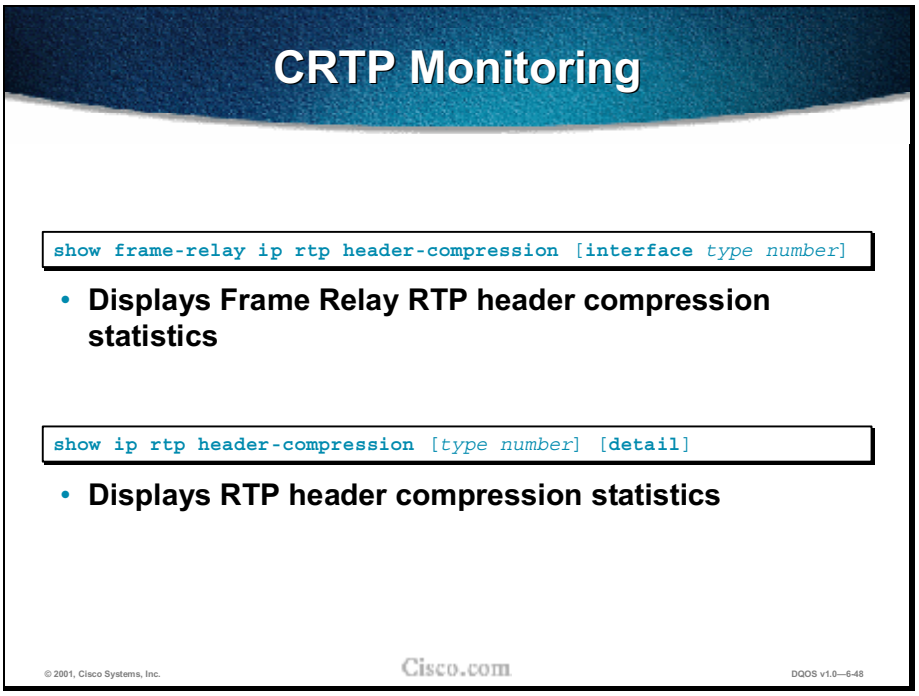
All the interface maps inherit the header compression configured on the interface. The *passive* keyword causes IOS to compress outgoing RTP packets only if incoming RTP packets on the same interface are compressed.



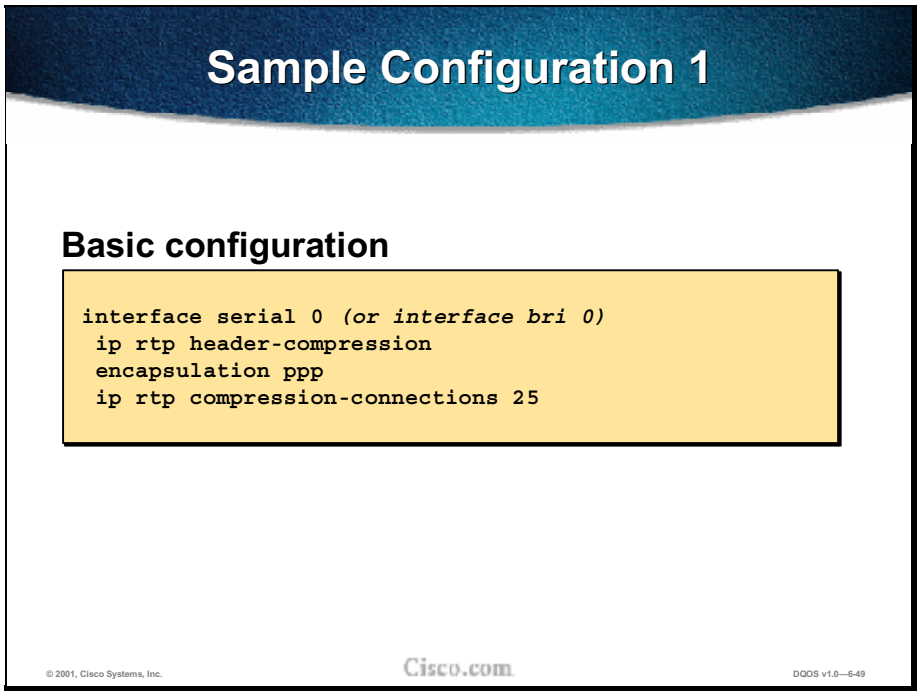
The use of FR map is discouraged. For voice configurations, you have to do map-class FR configurations; otherwise you can't get to all the QoS features you need.



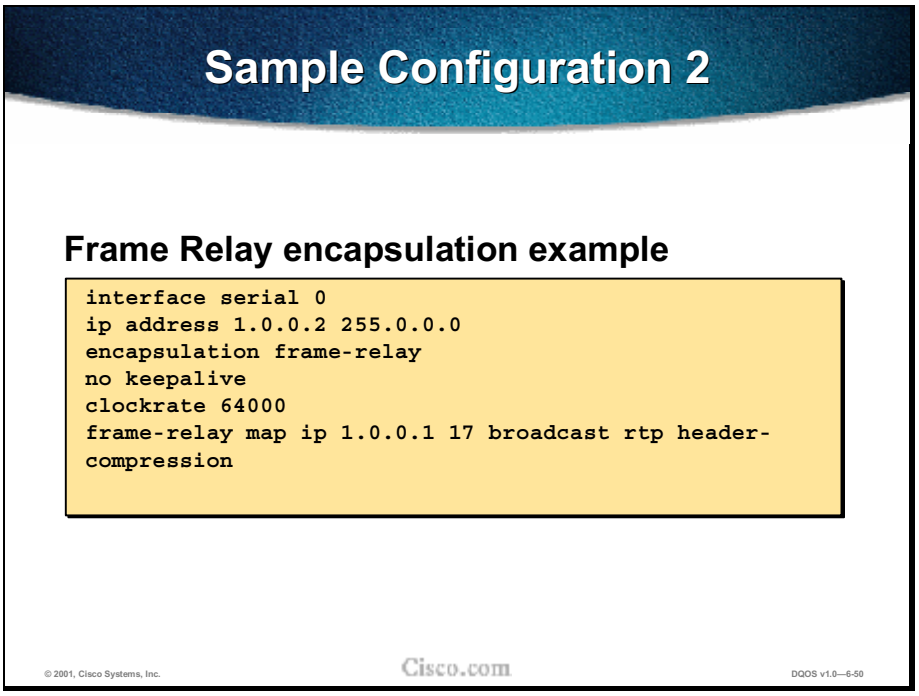
By default, 16 CRTP header compression connections are allowed per interface.



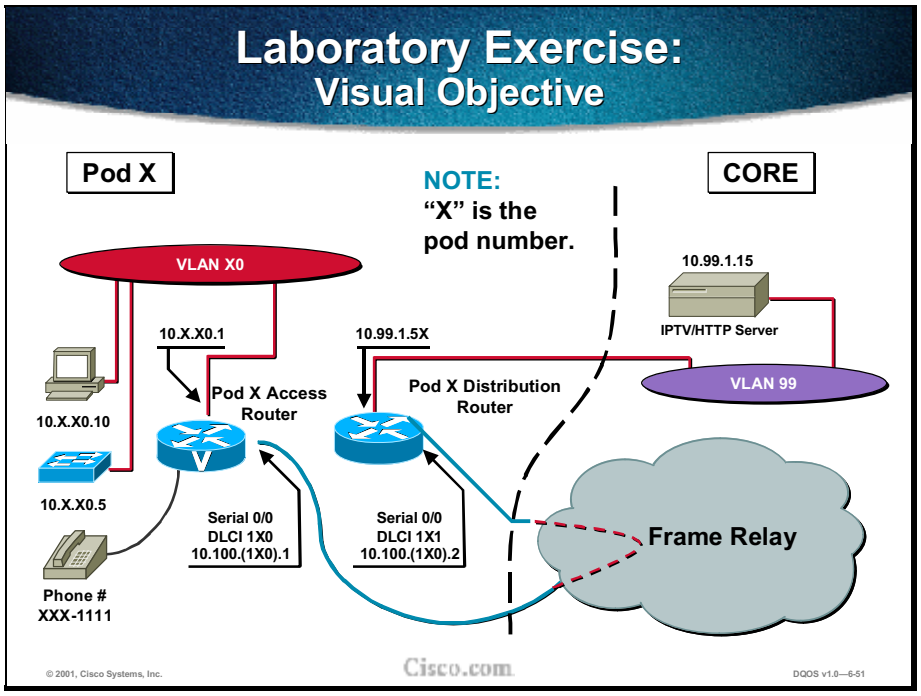
To display various routing statistics, use the above commands in EXEC mode. They can be used to display specific statistics, such as the contents of IP routing tables, caches, and databases. Information provided can be used to determine resource utilization and solve network problems. You can also display information about node reachability and discover the routing path your device's packets are taking through the network.



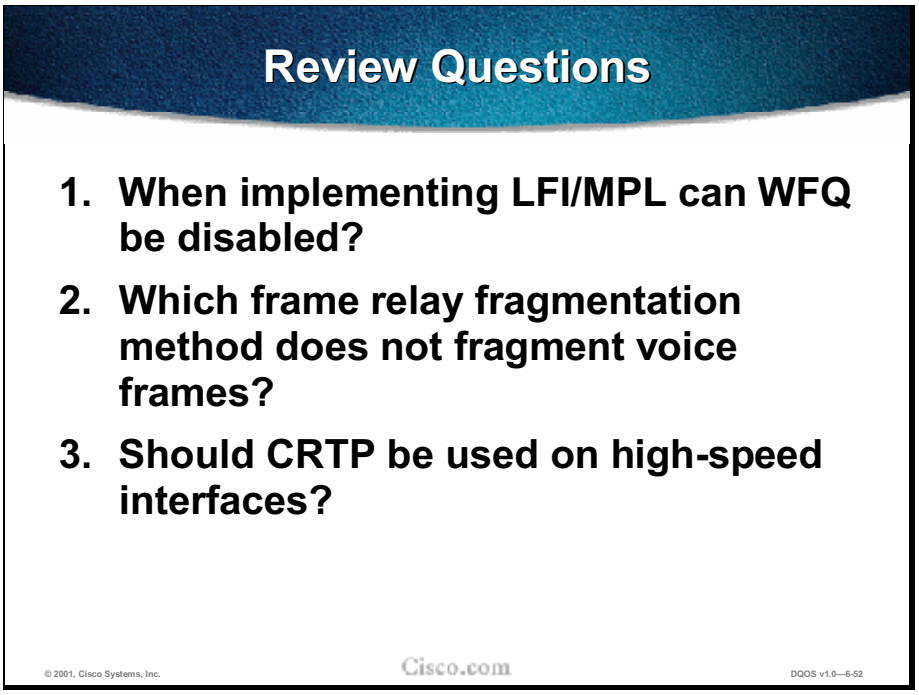
The above series of configuration commands enables RTP header compression for a serial, ISDN, or asynchronous interface. For ISDN, you also need a broadcast dialer map.



The above example for FR encapsulation enables RTP header compression on a specified map.


Laboratory Exercise:
Visual Objective
Pod X
CORE
NOTE:
"X" is the
pod number.
VLAN X0
10.99.1.15
IPTV/HTTP Server
10.X.X0.1
10.99.1.5X
Pod X Access
Router
Pod X Distribution
Router
VLAN 99
10.X.X0.10
10.X.X0.5
Serial 0/0
DLCI 1X0
10.100.(1X0).1
Serial 0/0
DLCI 1X1
10.100.(1X0).2
Frame Relay
Phone #
XXX-1111
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—6-51

Review Questions



Q1) When implementing LFI / MLP can WFQ be disabled?

A) No, WFQ must be enabled.

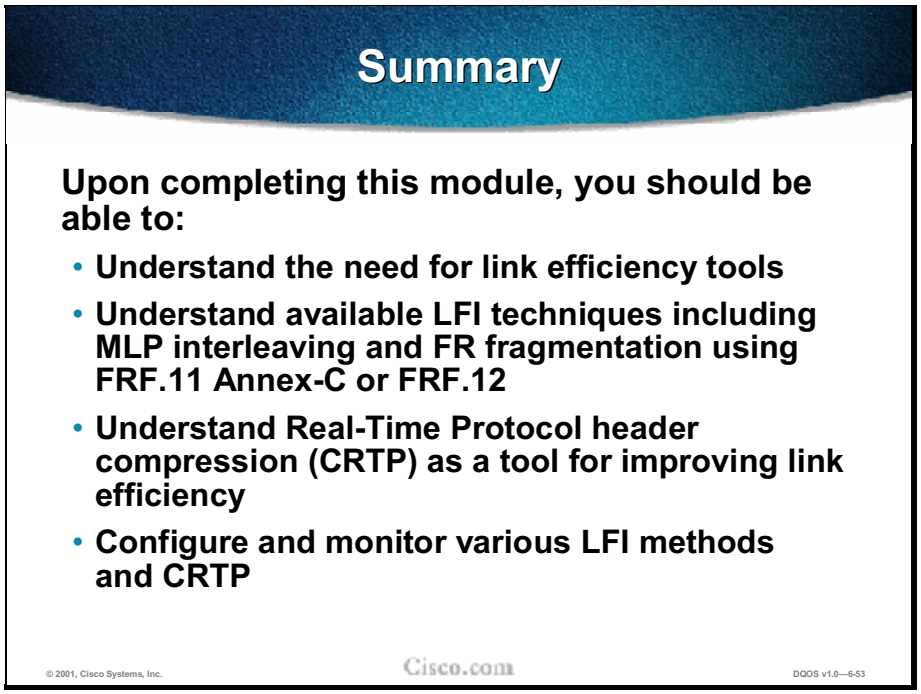
Q2) Which frame relay fragmentation method does not fragment voice frames?

A) FRF.11 Annex-C

Q3) Should CRTP be used on high-speed interfaces?.

A) No, CRTP is CPU intensive and the tradeoff is not worth it.

Summary


Summary
Upon completing this module, you should be able to:
• Understand the need for link efficiency tools
• Understand available LFI techniques including MLP interleaving and FR fragmentation using FRF.11 Annex-C or FRF.12
• Understand Real-Time Protocol header compression (CRTP) as a tool for improving link efficiency
• Configure and monitor various LFI methods and CRTP
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—6-53

7

Shaping and Policing

Overview

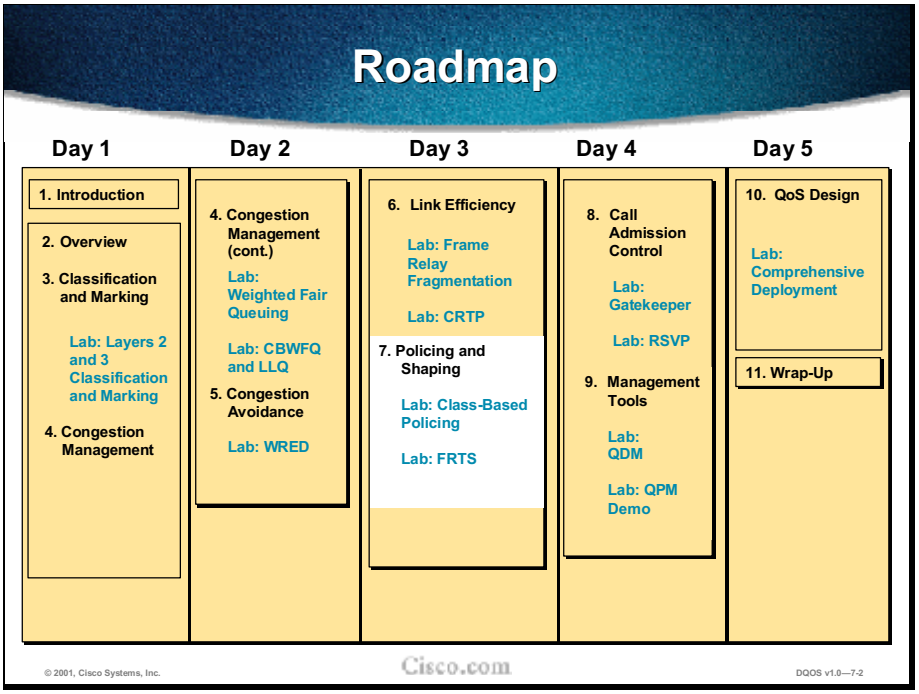
This chapter surveys the policing and traffic shaping features in Cisco IOS and describes the problems they solve. Topics include differences in policing and shaping, token buckets, rate limiting using committed access rate (CAR) and class-based policing, class-based shaping, Frame Relay traffic shaping (FRTS), and generic traffic shaping (GTS).

Objectives

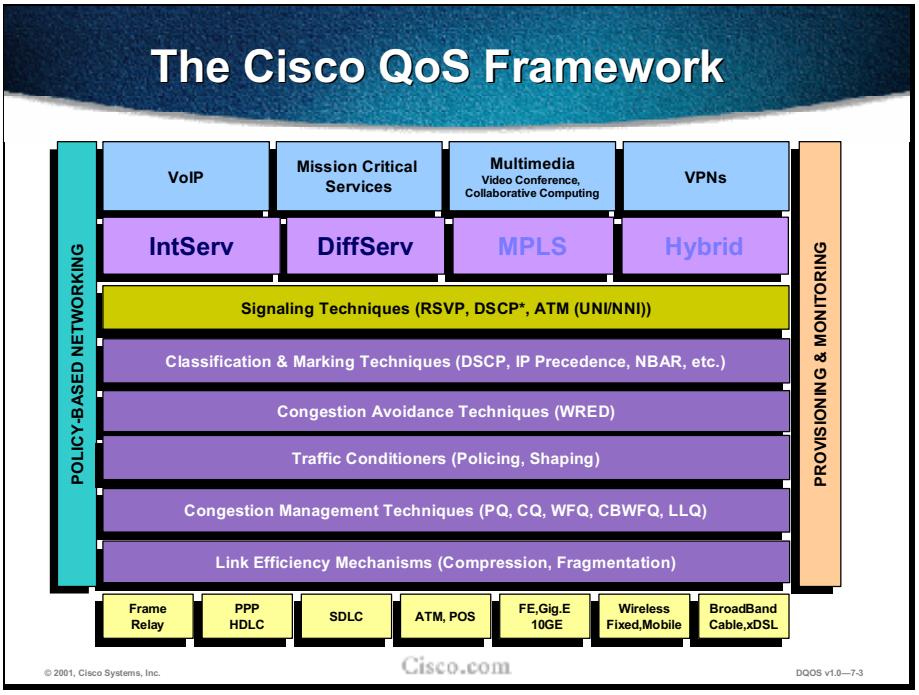
Upon completing this chapter, you will be able to:

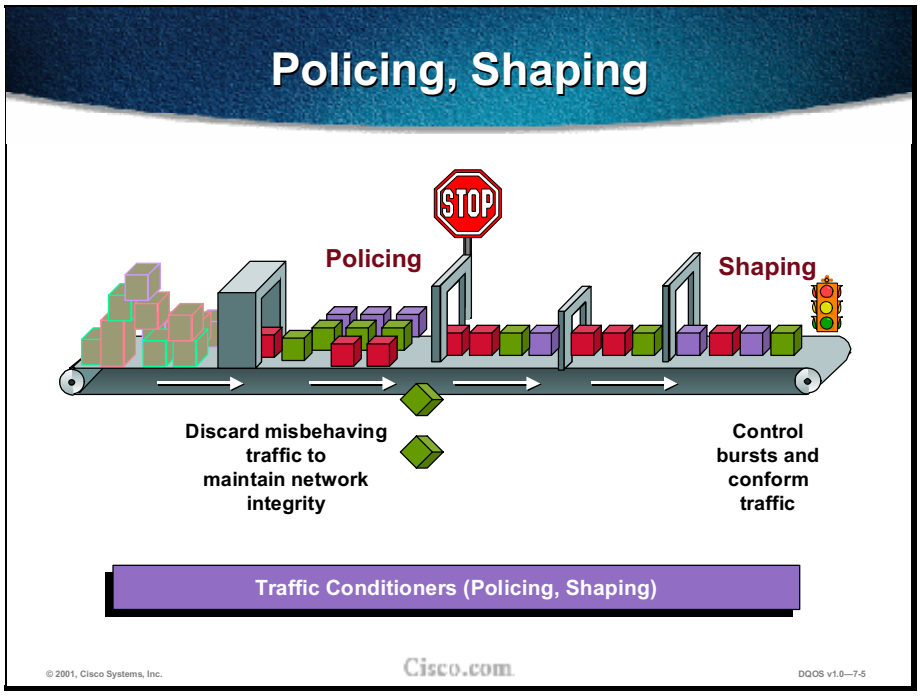
- Describe the difference between policing and shaping and how each one relates to QoS
- Describe various mechanisms for policing, when to apply each, and how to configure them
- Identify the various types of traffic shaping, their differences, and how to apply each
- Configure the different types of traffic shaping

Outline



The figure shows the plan for the week.





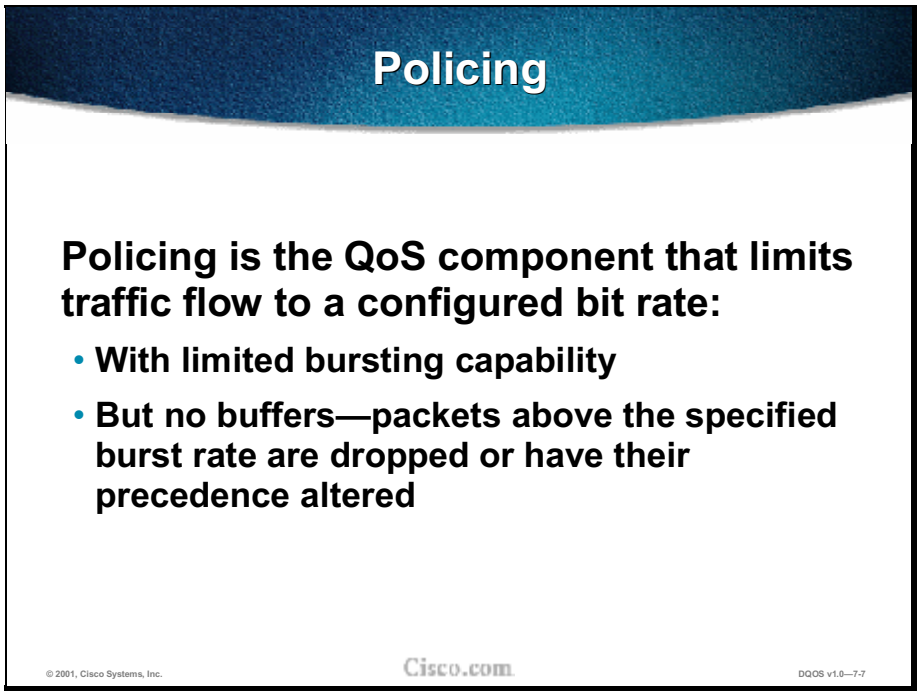
Both policing and shaping mechanisms occur within the network. They use the already marked ToS or DSCP bits discussed elsewhere.

With policing, the rate at which traffic can flow is capped. This is done inbound in order to control how fast the data is sent.

With shaping, the bursts are smoothed out for a steadier flow of data. Reduced burstiness helps reduce congestion in a network core.

Agenda

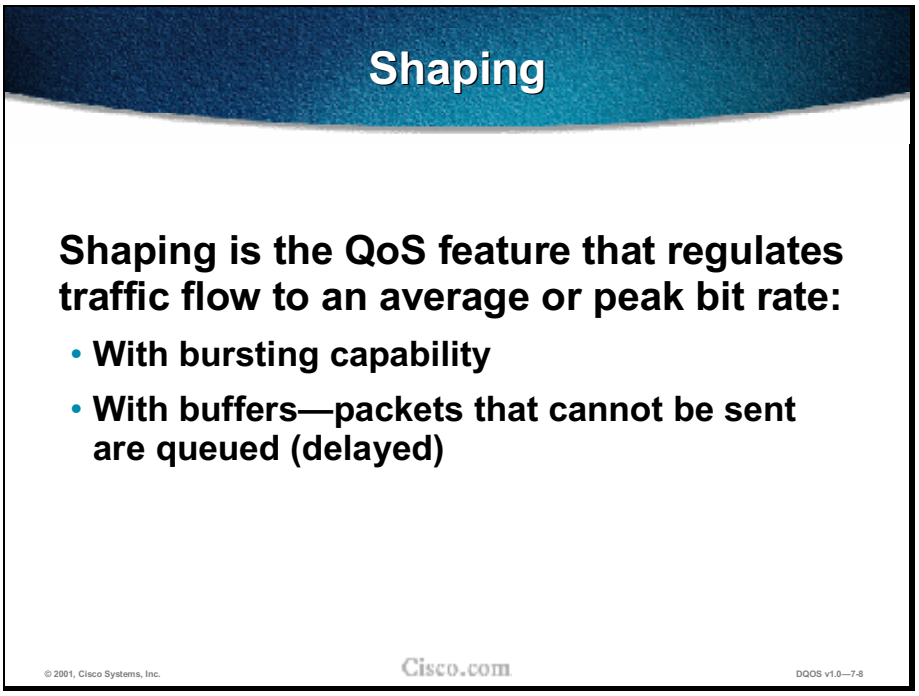
- Token Bucket
- Policing
  - Rate Limiting in CAR
  - Class-Based Policing
- Traffic Shaping
  - Class-Based Shaping
  - Generic Traffic Shaping (GTS)
  - Distributed Traffic Shaping (DTS)
  - Frame Relay Traffic Shaping (FRTS)



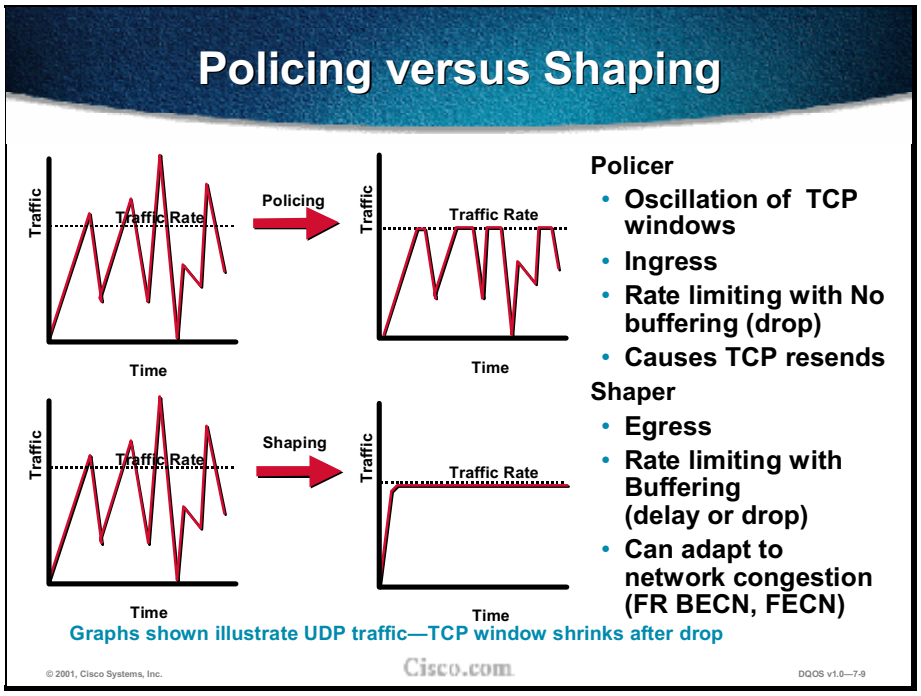
A policer typically drops nonconforming traffic.

For example, the rate-limiting policer, committed access rate (CAR), will either drop the packet or rewrite its IP Precedence, resetting the packet header's ToS bits.

Class-based policing may also be implemented using MQC.



A shaper typically delays excess traffic using a buffer, or mechanism, to hold packets and shape the flow when the data rate of the source is higher than expected.



The diagram shows the effect of policing and shaping on traffic flow.

Both policing and shaping ensure that traffic does not exceed a bandwidth limit. Policing and shaping both limit bandwidth but with a different impact on traffic:

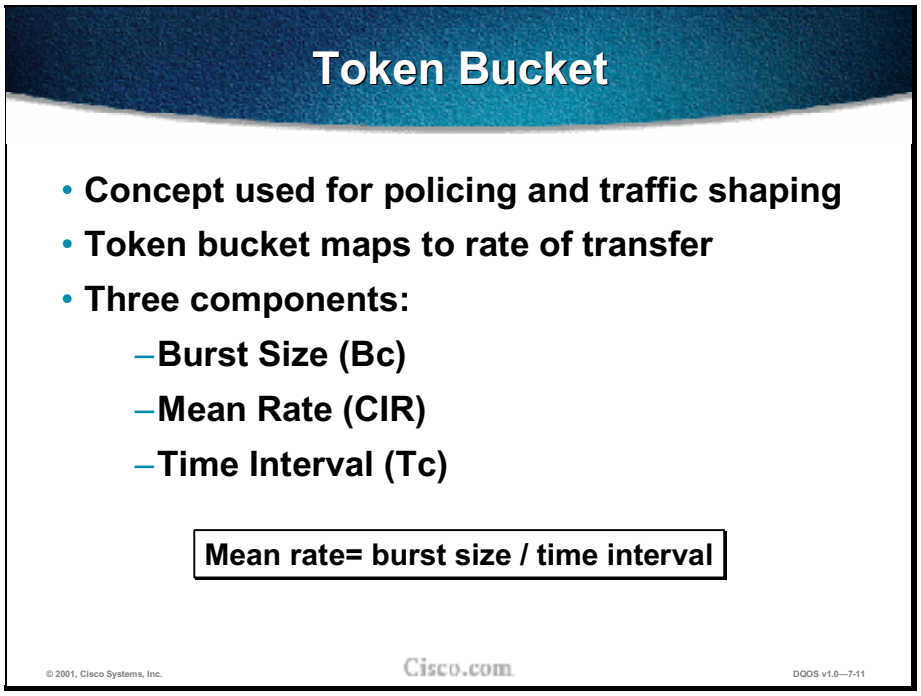
- Policing drops more often—there are more resends
- Shaping adds variable delay.

Traffic shaping smoothes traffic by storing traffic above the configured rate in a queue.

When a packet arrives at the interface for transmission, the following happens:

- If the queue is empty; the traffic shaper processes the arriving packet:
  - If possible, the traffic shaper sends the packet.
  - Otherwise, the packet is placed in the queue.
- If the queue is not empty, the packet is placed in the queue.

When there are packets in the queue, the traffic shaper removes the numbers of packets it can send from the queue every time interval.



A token bucket is used to manage a device that regulates the data flow. For example, the regulator might be a traffic policer, such as Frame Relay traffic shaping (FRTS) or generic traffic shaping (GTS). A token bucket itself has no discard or priority policy. A token bucket discards tokens and does not manage the transmission queue if the flow overdrives the regulator. Neither committed access rate (CAR) nor FRTS and GTS implement either a true token bucket or a true leaky bucket.

Mean rate:

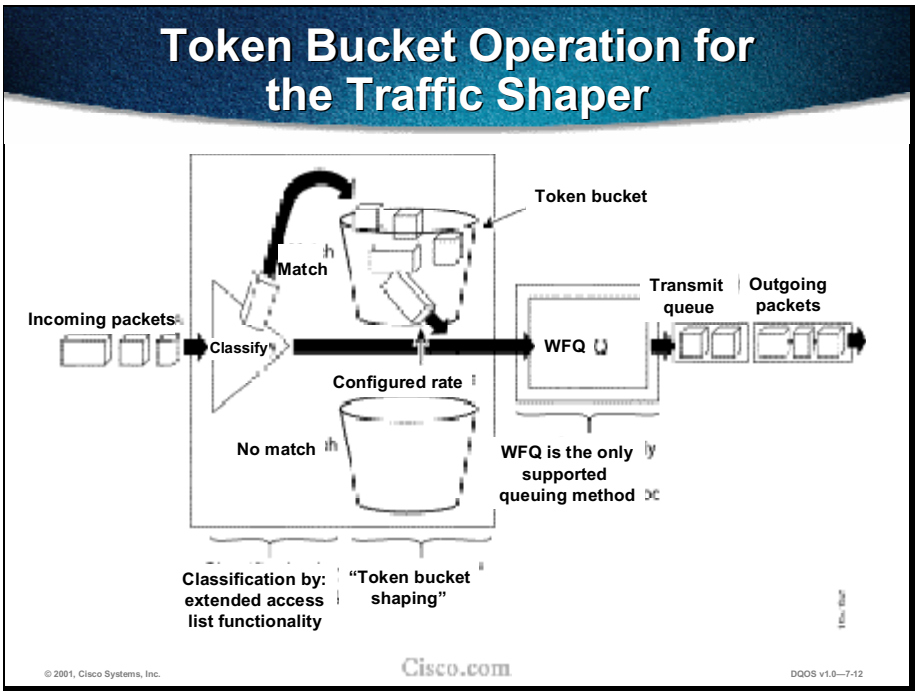
- Committed information rate (CIR)
- Specifies average data sent/forwarded per unit of time

Burst size:

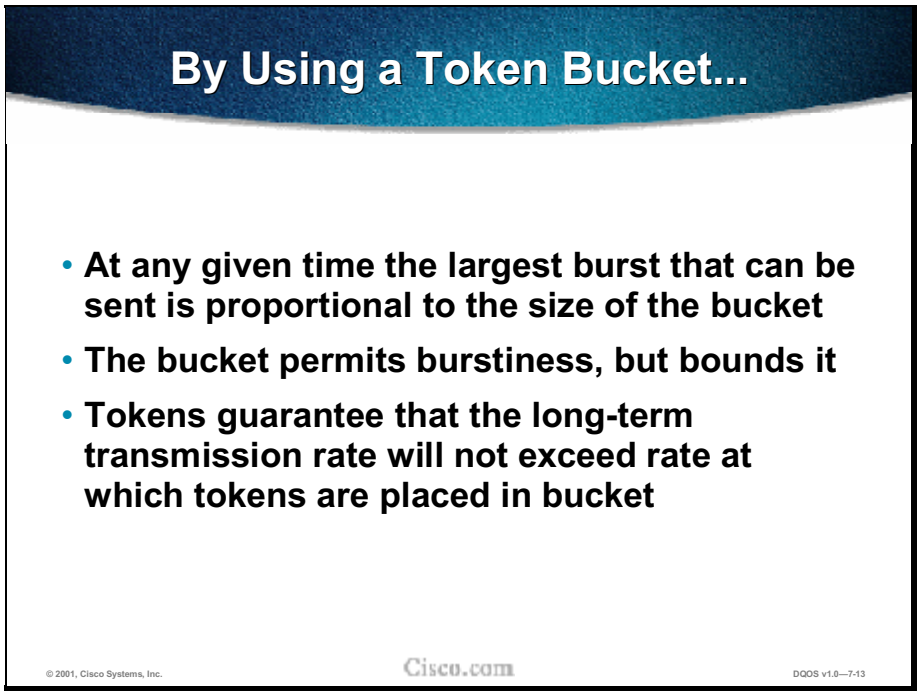
- Committed burst size (Bc)
- Specified in bits (or bytes) per burst
- Indicates how much data can be sent within given unit of time without creating scheduling concerns

Time interval:

- Measurement interval
- Time quantum in seconds per burst



- Used to manage the regulating device
  - Itself has no discard or priority policy
- Tokens put into bucket at a certain rate
  - Each token has permission to send fixed number of bits into network
- Bucket itself has a specified capacity
- If the bucket fills to capacity, newly arriving tokens are discarded
  - Discarded tokens not available to future packets
- To send a packet, regulator removes from the bucket the number of tokens that represent size of packet
  - If not enough tokens are in bucket to send packet, regulator may:
- Wait for enough tokens to accumulate in bucket (traffic shaping), or
- Discard packet or mark it down (policing)



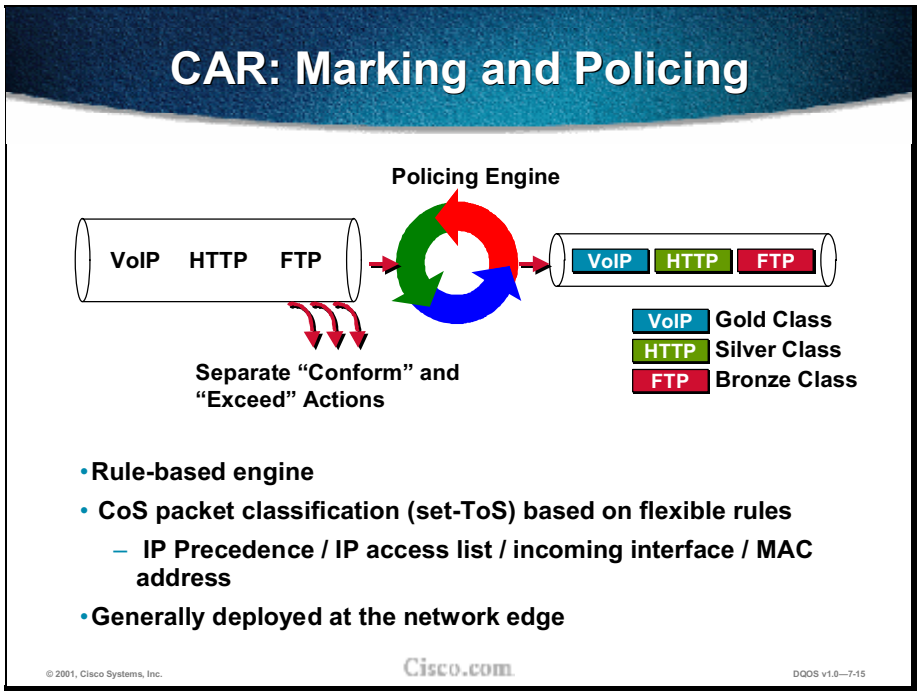
Bc is known as burst capacity. Be is excess burst capacity. Tc is the time interval constant. CIR is the committed information rate. These terms derive from Frame Relay technology.

In the token bucket metaphor, tokens are put into the bucket at a certain rate, Bc tokens every Tc seconds. The bucket itself has a specified capacity. If the bucket fills to capacity (Bc + Be), newly arriving tokens are discarded. Each token has permission for the source to send a certain number of bits into the network. To send a packet, the regulator must remove from the bucket a number of tokens equal in representation to the packet size.

If not enough tokens are in the bucket to send a packet, the packet either waits until the bucket has enough tokens or the packet is discarded. If the bucket is already full of tokens, incoming tokens overflow and are not available to future packets. Thus, at any time, the largest burst a source can send into the network is roughly proportional to the size of the bucket.

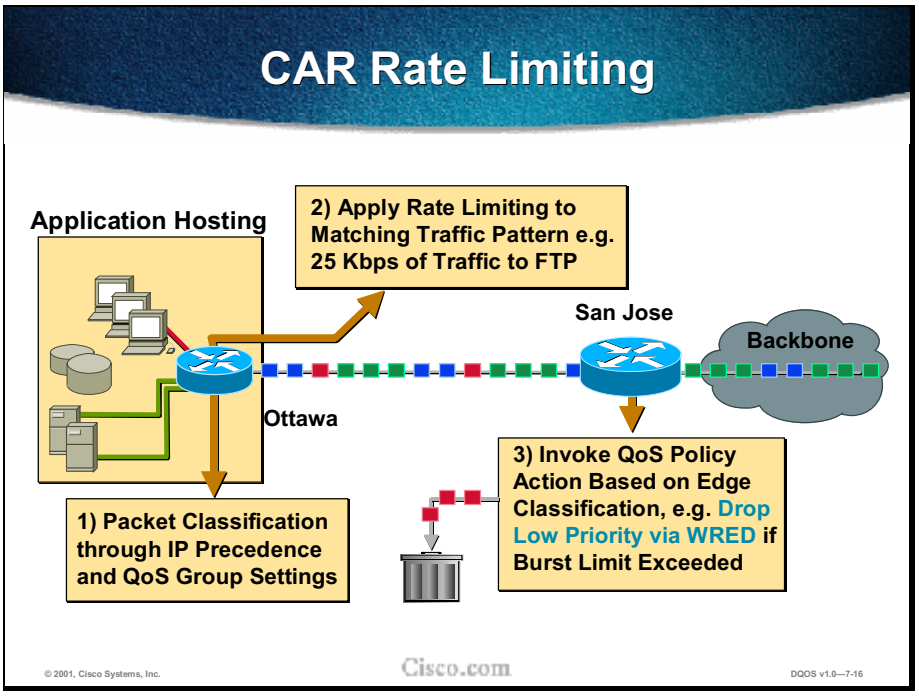
Note that the token-bucket mechanism used for traffic shaping has both a token bucket and a data buffer, or queue; if it did not have a data buffer, it would be a policer. For traffic shaping, packets that arrive that cannot be sent immediately are delayed in the data buffer.

For traffic shaping, a token bucket permits burstiness but bounds it. It guarantees that the burstiness is bounded so that the flow will never send faster than the token bucket's capacity plus the time interval divided by the established rate at which tokens are placed in the bucket. It also guarantees that the long-term transmission rate will not exceed the established rate at which tokens are placed in the bucket.



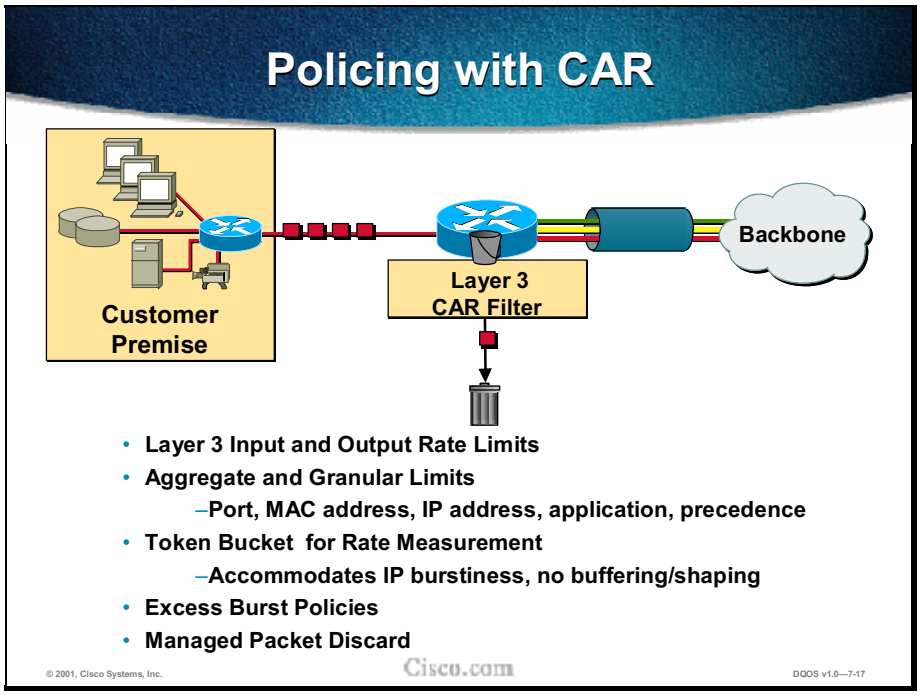
Once a packet has been measured as conforming to or exceeding a particular rate limit, the router performs one of the following actions on the packet:

- Transmit—The packet is sent.
- Drop—The packet is discarded.
- Set precedence (or perhaps DSCP bits) and transmit—The IP Precedence (ToS) bits in the packet header are rewritten. The packet is then sent. This action can be used to either color (set precedence) or recolor (modify existing packet precedence) the packet.
- Continue—The packet is evaluated using the next rate policy in a chain of rate limits. If there is not another rate policy, the packet is sent.



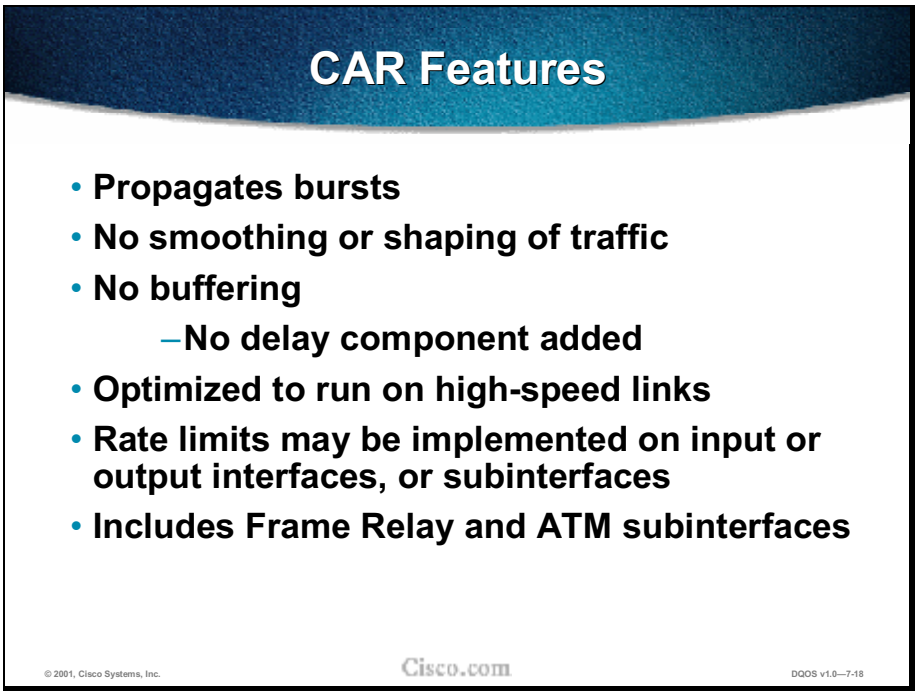
CAR provides the capability to allow the service provider to specify a policy that determines which packets should be assigned to which traffic class. The IP header already provides a mechanism to do this, namely the three precedence bits in the Type of Service field in the IP header. CAR allows the setting of policies based on information in the IP or TCP header, such as IP address, application port, physical port, or subinterface, IP protocol, etc., to decide how the precedence bits should be marked or "colored." Once marked, appropriate treatment can be given in the backbone to ensure that premium packets get premium service in terms of bandwidth allocation, delay control, and so forth.

Note that CAR can also be used to police precedence bits set externally to the network either by the customer or by a downstream service provider. Thus the network can decide to either accept or override external decisions.



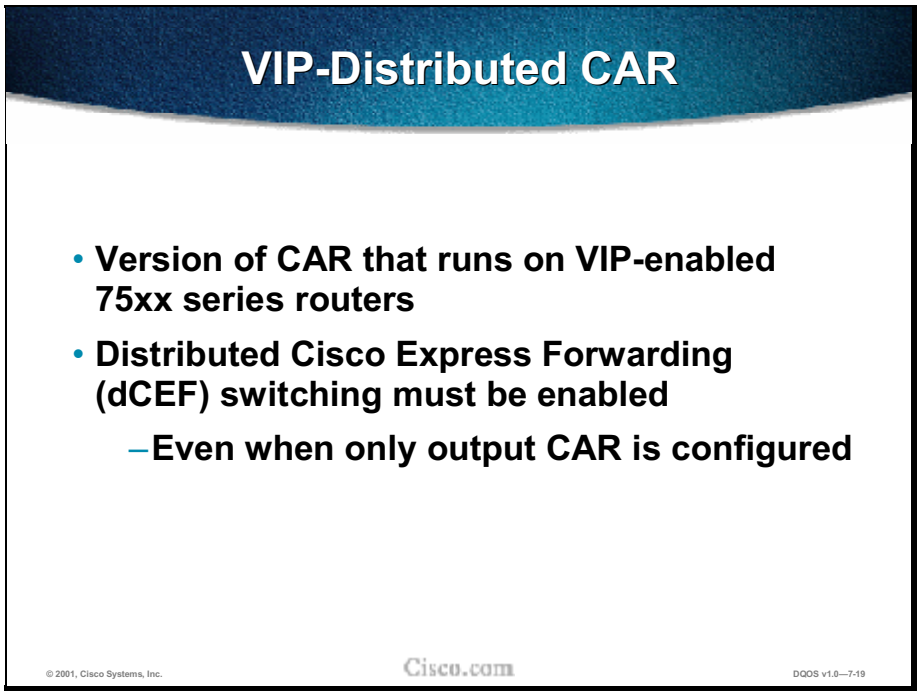
CAR's rate-limiting feature manages a network's access bandwidth policy by ensuring that traffic falling within specified rate parameters is sent, while dropping packets that exceed the acceptable amount of traffic or sending them with a different priority.

CAR is often configured on interfaces at the edge of a network to limit traffic into or out of the network.



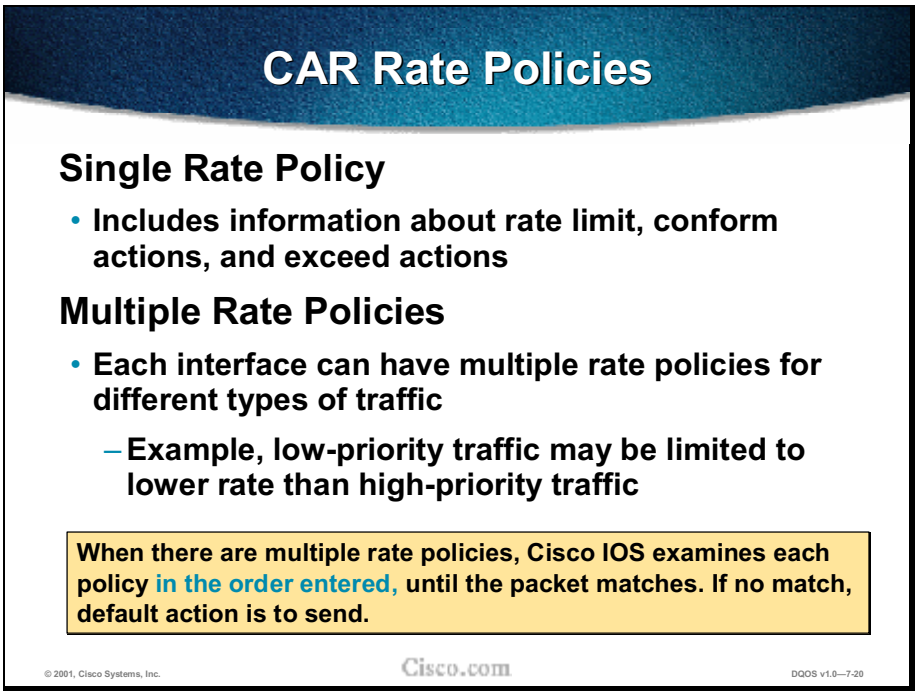
The rate-limiting function of CAR does the following:

- Allows you to control the maximum rate of traffic transmitted or received on an interface
- Gives you the ability to define Layer 3 aggregate or granular incoming or outgoing (ingress or egress) bandwidth rate limits and to specify traffic-handling policies when the traffic either conforms to or exceeds the specified rate limits
- Uses aggregate bandwidth rate limits to match all the packets on an interface or subinterface
- Uses granular bandwidth rate limits to match a particular type of traffic based on precedence, MAC address, or other parameters



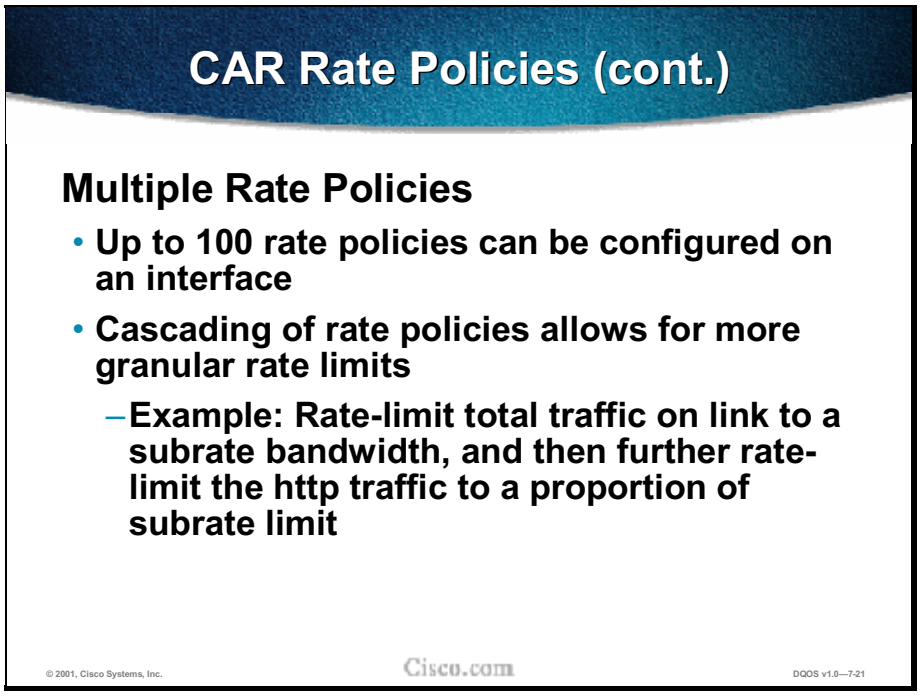
VIP-distributed CAR is a version of CAR that runs on the Versatile Interface Processor (VIP). It is supported on the Cisco 7500 routers with a VIP2-40 or greater interface processor.

Distributed Cisco Express Forwarding (dCEF) switching must be enabled on any interface that uses VIP-Distributed CAR, even when only output CAR is configured.

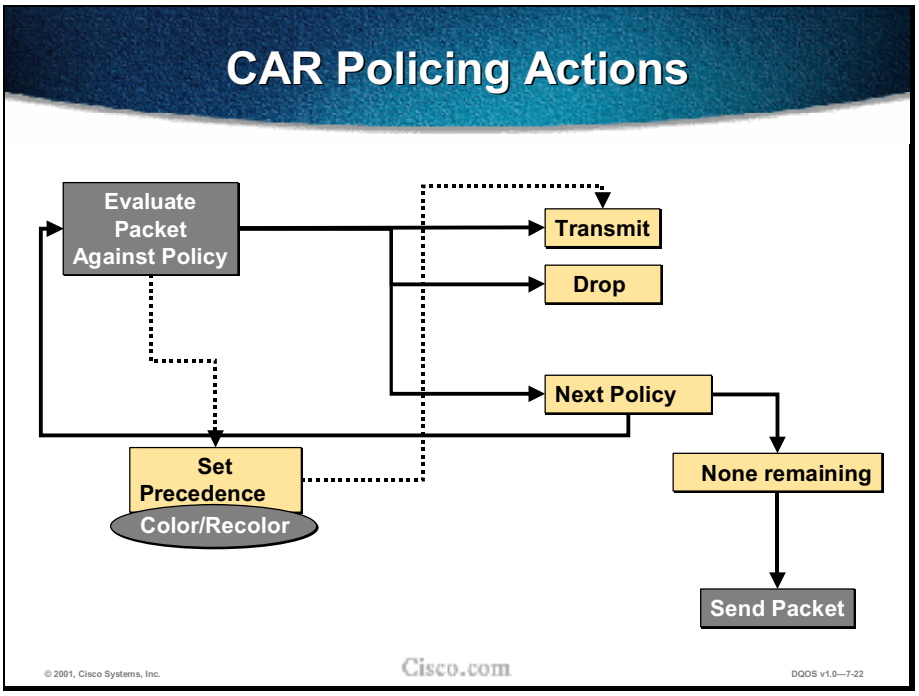


A single CAR rate policy includes information about the rate limit, conform actions, and exceed actions. Each interface can have multiple CAR rate policies corresponding to different types of traffic. For example, low-priority traffic may be limited to a lower rate than high-priority traffic. When there are multiple rate policies, the router examines each policy in the order entered, until the packet matches. If no match is found, the default action is to transmit.

Rate policies can be independent: Each rate policy deals with a different type of traffic. Alternatively, rate policies can be cascading: A packet may be compared to multiple different rate policies in succession.



Cascading of rate policies allows a series of rate limits to be applied to packets to specify more granular policies. For example, the total traffic on an access link can be rate-limited to a specified subrate bandwidth, and then the World Wide Web traffic on the same link can be limited to a given proportion of the subrate limit. CAR can be configured to match packets against an ordered sequence of policies until an applicable rate limit is encountered—that is, rate-limiting several MAC addresses with different bandwidth allocations at an exchange point. Up to a 100 rate policies can be configured on a subinterface.



Configurable actions are:

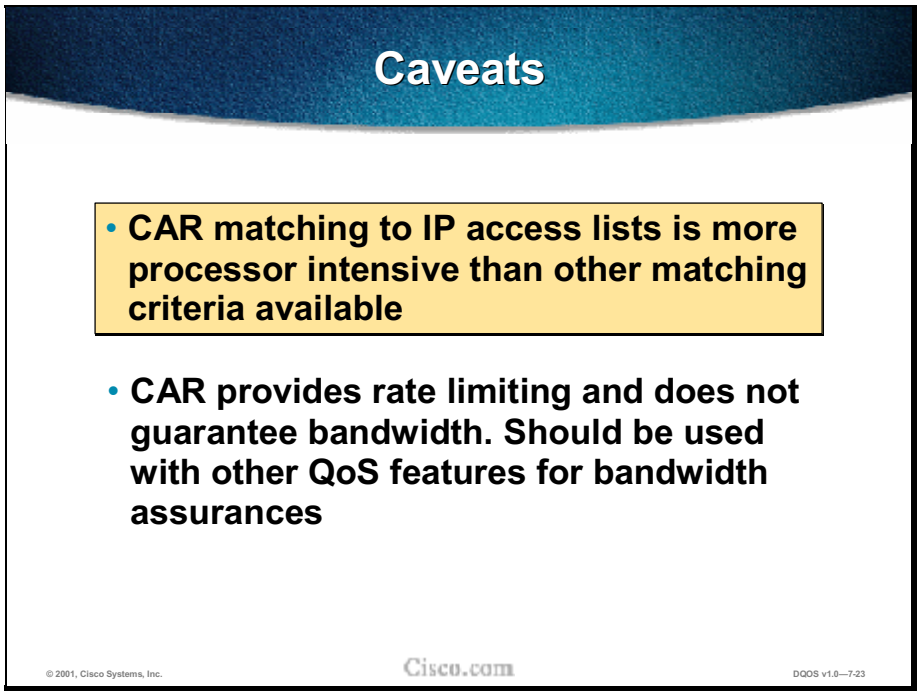
- Transmit
- Drop
- Continue (go to the next rate-limit or police statement in the list)
- Set precedence and transmit (rewrite the IP Precedence bits and transmit)
- Set precedence and continue (rewrite the IP Precedence bits and go to the next rate-limit or police statement in the list)

A maximum of 100 rate policies can be configured per interface.

Each CAR rate-limit statement is checked sequentially for a match. When a match is found, the token bucket, if there is one, is evaluated.

If the action is a continue action, the policer will go to the next rate-limit on the list to find a subsequent match. If a match is found, the traffic is subjected to the next applicable rate limit.

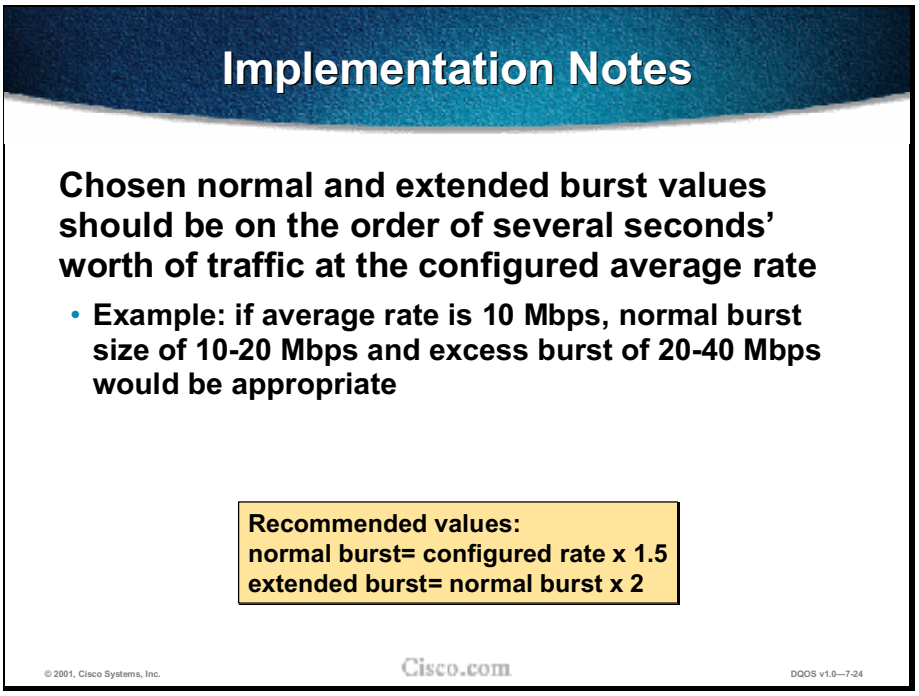
If an end of rate-limit list is encountered without finding a match or continue action, the default behavior is to transmit.



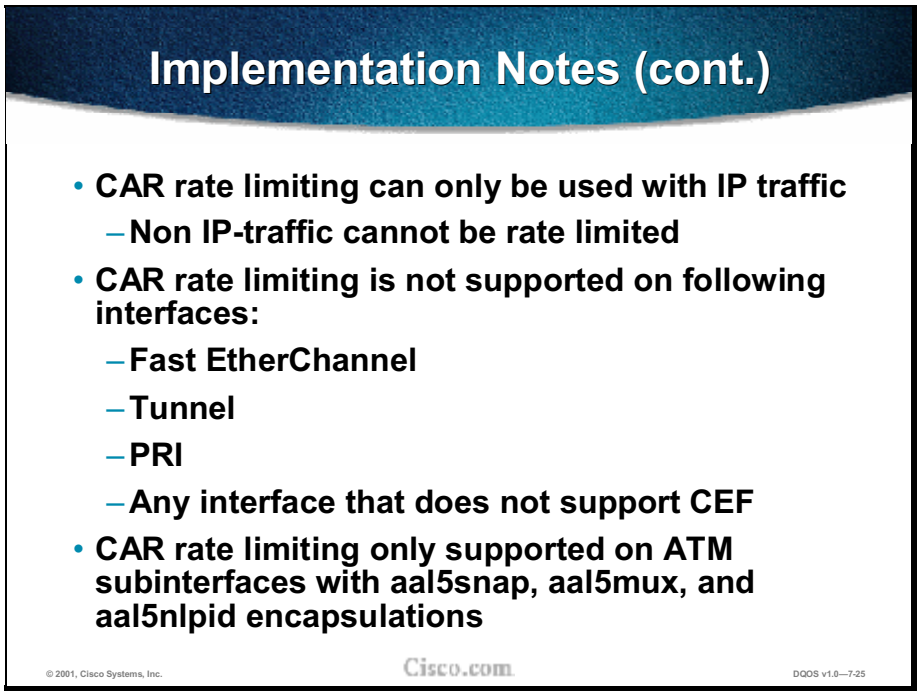
CAR rate policies may be associated with any of the following:

- Incoming interface
- All IP traffic
- IP Precedence
  - Defined by rate-limit access list
- MAC address
  - Defined by rate-limit access list
- Standard or extended IP access list

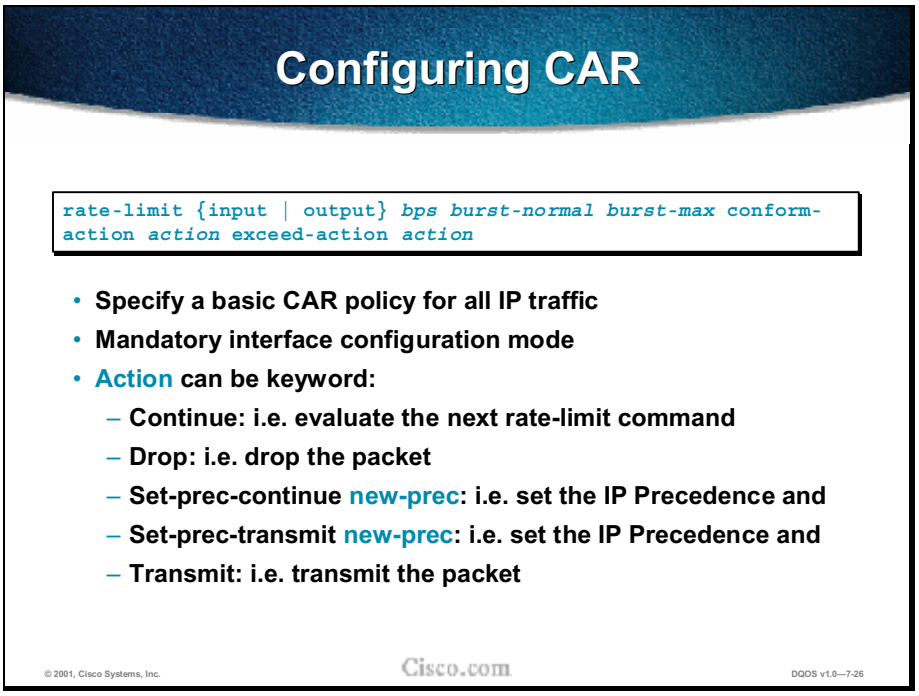
However, use of access lists is processor intensive—other matching criteria should be used whenever possible.



The guidelines described above and in ensuing figures can be used as rules of thumb for optimal performance when configuring rate limiting using CAR.



Rate limiting using CAR is applicable to IP traffic and on certain interfaces/subinterfaces only. The list of interfaces/subinterfaces that support CAR may change with newer IOS releases. It is recommended to check for new features in release updates.

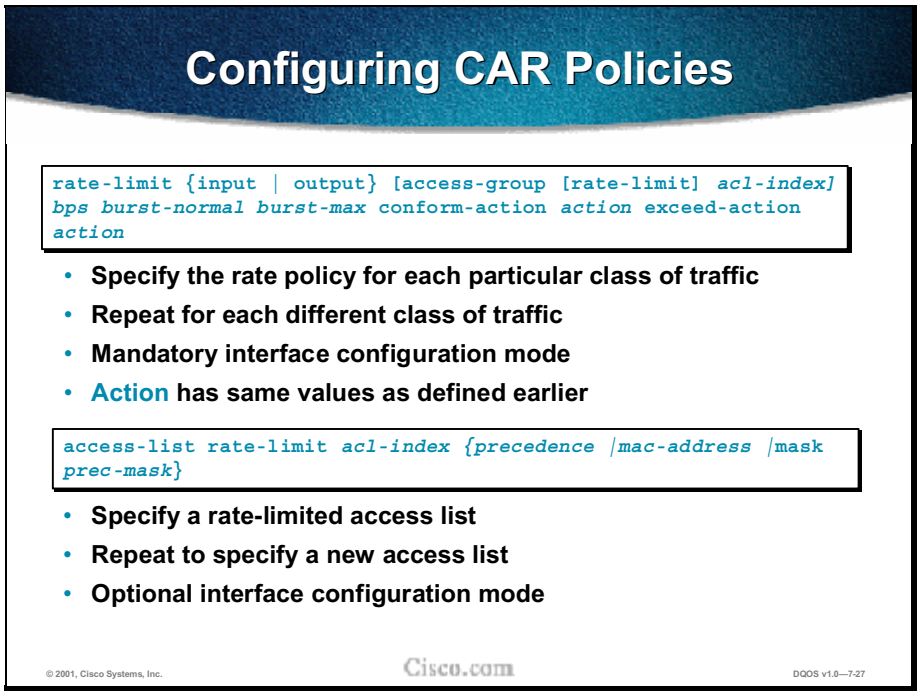


To configure CAR and distributed CAR (DCAR) policies, use the **rate-limit** interface configuration command. To remove the rate limit from the configuration, use the **no** form of this command.

Syntax: `rate-limit {input | output} [access-group [rate-limit] acl-index] bps burst-normal burst-max conform-action action exceed-action action`Action may be one of the following:

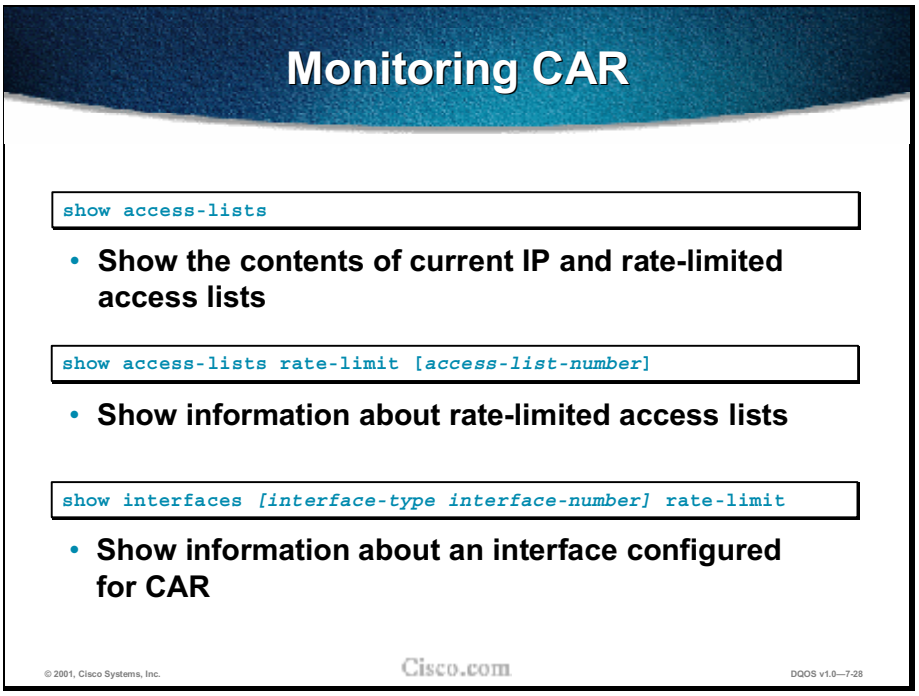
- **continue**: Evaluate the next **rate-limit** command.
- **drop**: Drop the packet.
- **set-prec-continue** *new-prec*: Set the IP Precedence and evaluate the next **rate-limit** command.
- **set-prec-transmit** *new-prec*: Set the IP Precedence and send the packet.
- **transmit**: Send the packet.

The *burst-max* parameter is sometimes left empty to use the default setting and drop excess traffic. You can think of the *burst-normal* parameter as the granularity of the measurement window. It determines how many bytes can be queued from the selected interface.



The CAR **rate-limit** command first appeared in Cisco IOS Release 11.1 CC. It is used to configure a CAR policy on an interface. To specify multiple policies, enter this command once for each policy.

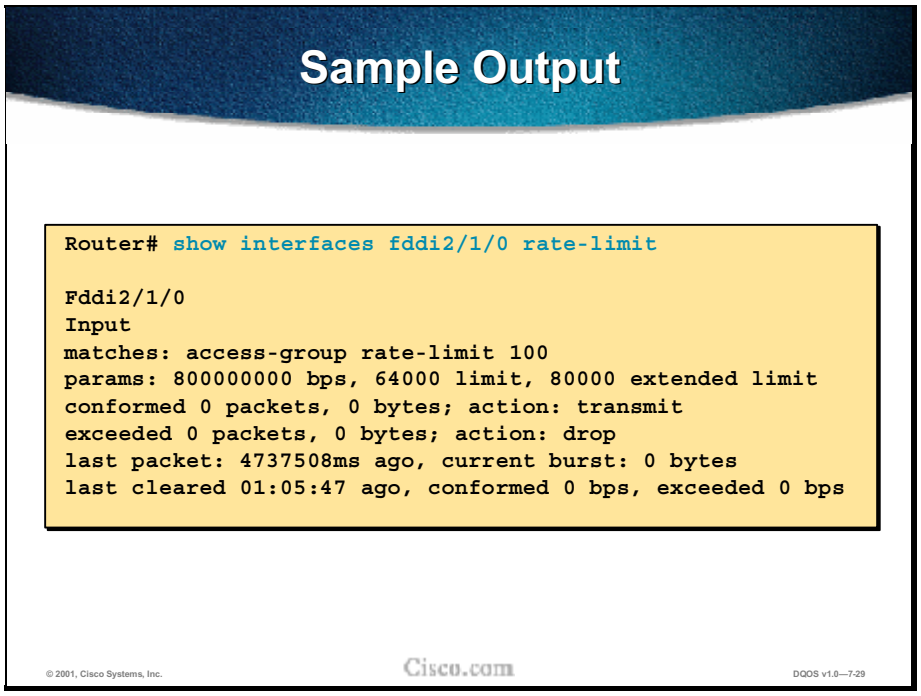
To configure an access list for use with CAR policies, use the **access-list rate-limit** global configuration command. This command first appeared in Cisco IOS Release 11.1 CC. This command classifies packets by the specified IP Precedence or MAC address for a particular CAR access list. Apply CAR policies, using the **rate-limit** command, to individual rate-limit access lists. Thus, the CAR process treats packets with different IP precedences or MAC addresses differently. Specify only one command for each rate-limit access list. If you are entering this command multiple times with the same access list number, the new command will overwrite the previous command. Use the *mask* keyword to assign multiple IP precedences to the same rate-limit list.



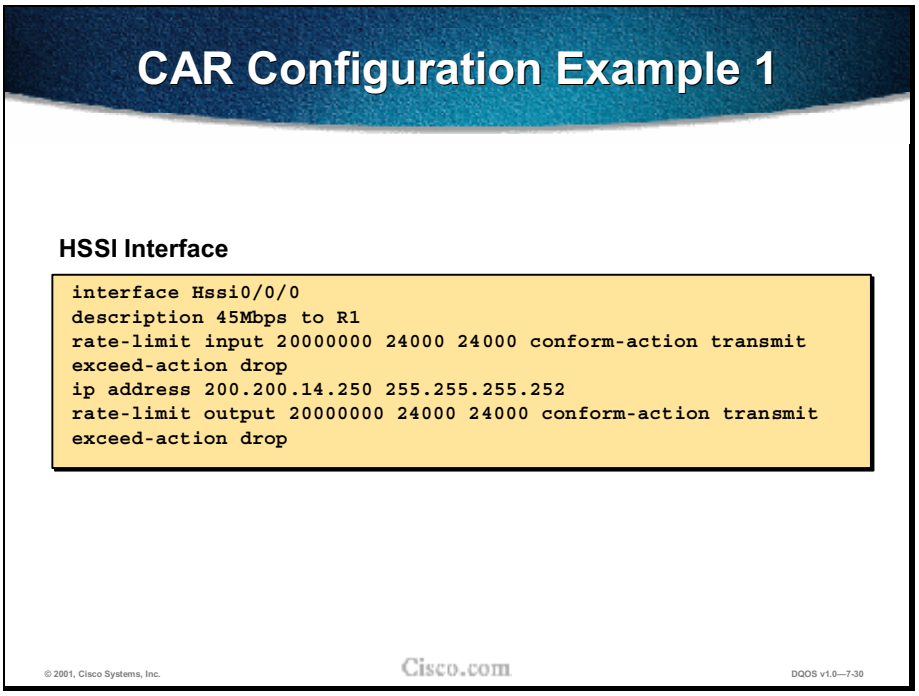
To display the contents of current IP and rate-limit access lists, use the **show access-lists** privileged EXEC command. This command appeared before Cisco IOS Release 10.0.

To display information about rate-limit access lists, use the **show access-lists rate-limit** EXEC command. This command first appeared in Cisco IOS Release 11.1 CC. Information displayed includes whether the access list is precedence-based or MAC address-based, what the IP Precedence and IP Precedence mask for packets in this rate-limit access list are, or what the MAC address for packets in this rate-limit access list.

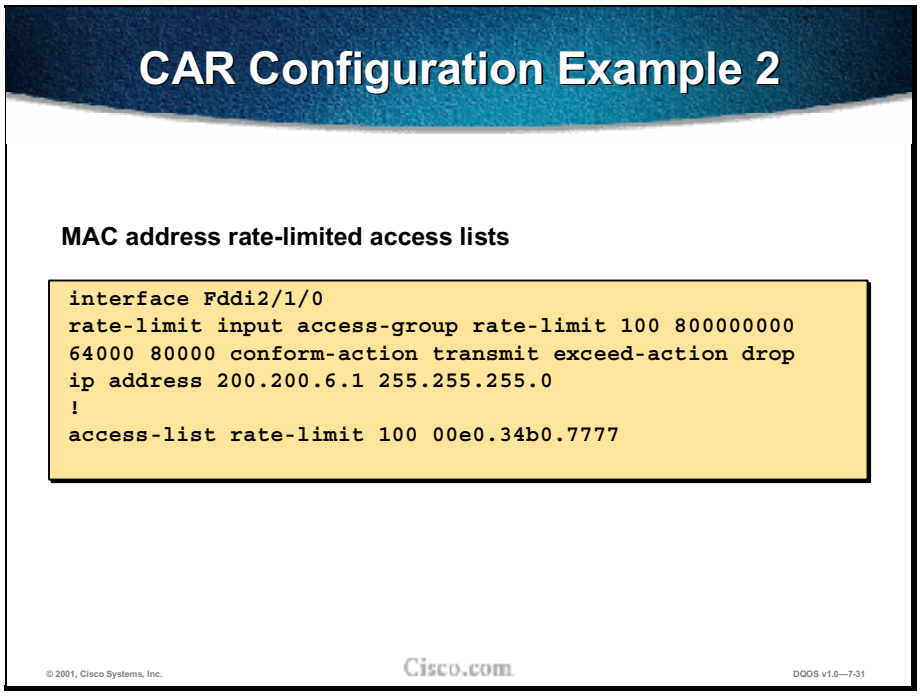
To display information about committed access rate (CAR) for an interface, use the **show interfaces rate-limit** EXEC command. This command first appeared in Cisco IOS Release 11.1 CC.



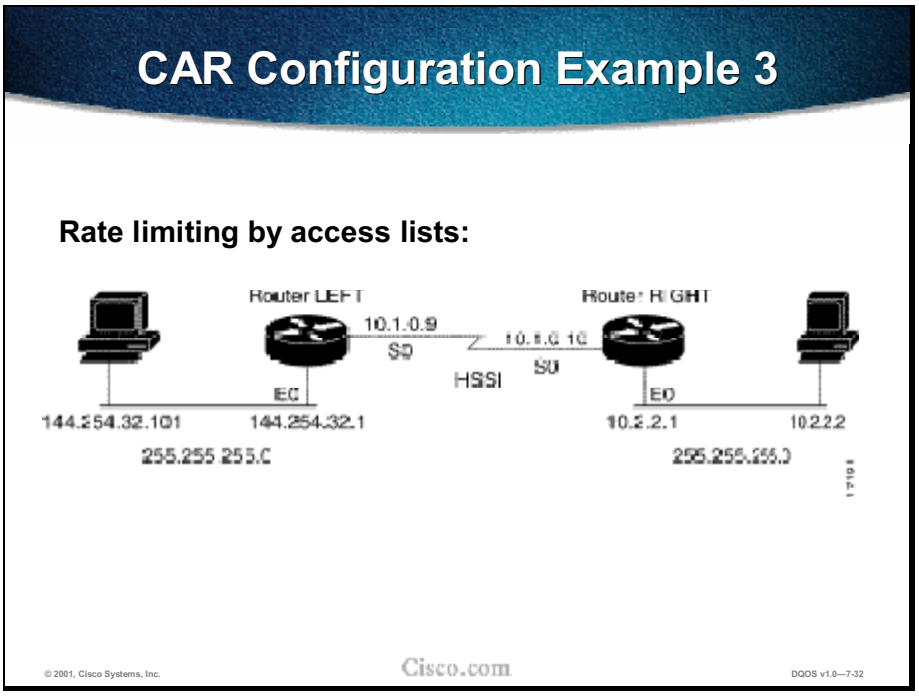
Information retrieved by the **show interface rate-limit** command includes packets that match this rate limit, parameters for this rate limit (as configured by the rate-limit command), average rate, normal burst size, excess burst size, number of packets that have conformed to the rate limit, conform action, number of packets that have exceeded the rate limit, exceed action, time since the last packet, instantaneous burst size at the current time, time since the burst counter was reset, rate of conforming traffic, rate of exceeding traffic, and rate limits applicable to packets sent out by the interface.



In this configuration example, HSSI (T3 link) is rate limited in transmissions to 20 Mbps; bursts of 24,000 bytes; all exceeding packets are dropped.



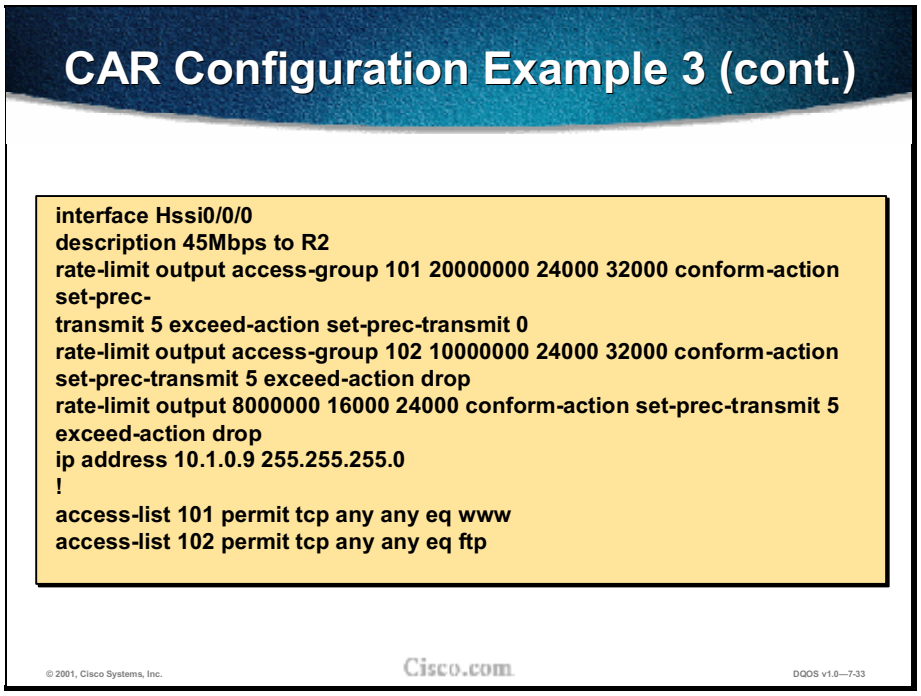
In this example, the FDDI Interface is configured for rate limiting. MAC address rate-limited access lists are used to control traffic from a particular source: Traffic from MAC address 00e0.34b0.7777 is compared to a rate limit of 80 Mbps of the 100 Mbps available on the FDDI connection—traffic that conforms to this rate is transmitted; nonconforming traffic is dropped.



Consider a network to which the following policy needs to be applied:

POLICY: All HTTP traffic is transmitted. However, the IP Precedence for HTTP traffic that conforms to the first rate policy is set to 5. For nonconforming web traffic, IP Precedence is set to 0 (best effort).

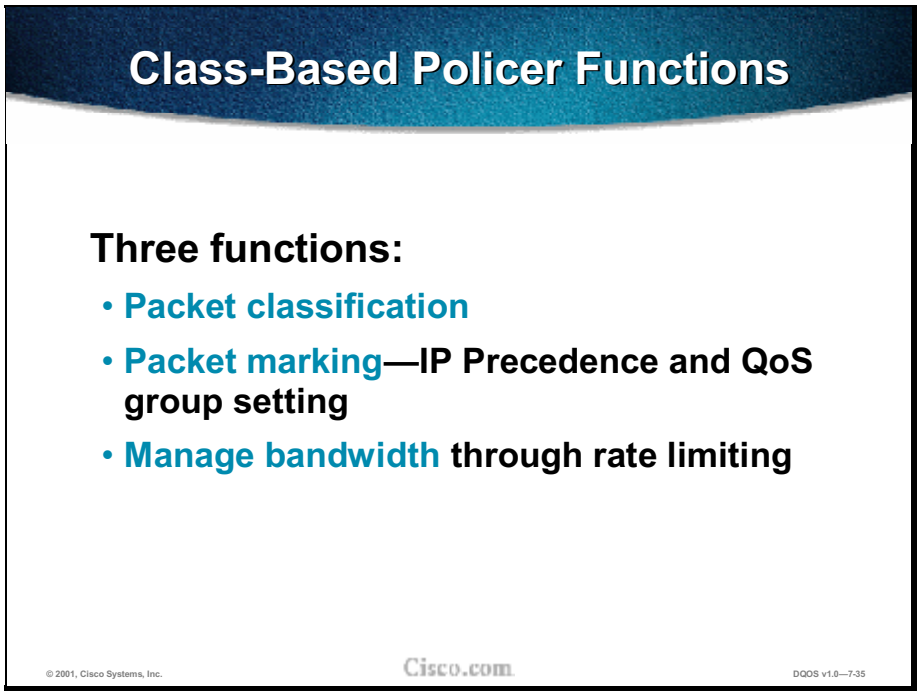
FTP traffic is transmitted with an IP Precedence of 5 if it conforms to the second rate policy. If the FTP traffic exceeds the rate policy, it is dropped.



To apply the policy, two access lists are created to classify the web and FTP traffic.

Each traffic flow is handled separately by CAR.

According to the configuration, all traffic not conforming to the two rate policies specified is limited to 8 Mbps, with a normal burst size of 16,000 bytes and an excess burst size of 24,000 bytes. Traffic that conforms is transmitted with an IP Precedence of 5. Traffic that does not conform is dropped.

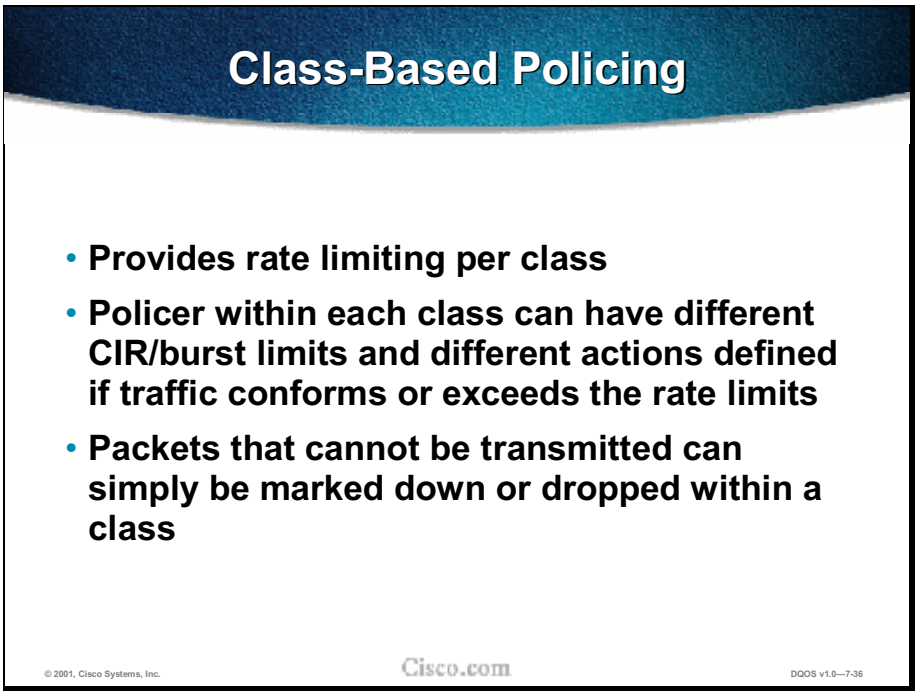


Cisco IOS now allows you to police traffic within a class to specified parameters. It supports DiffServ.

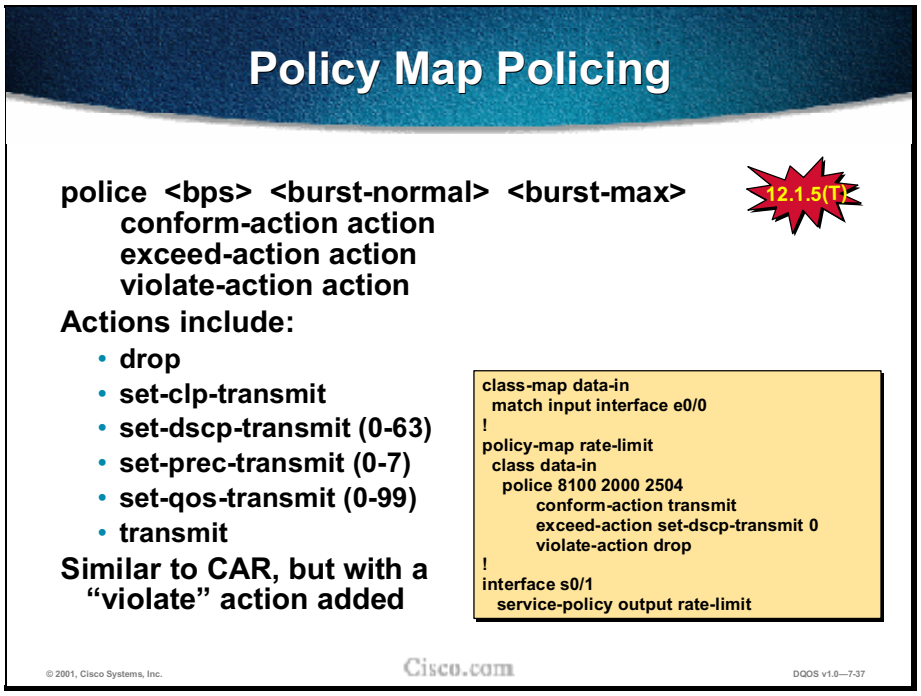
The drop-precedence bits can be set based on the committed burst (Bc) size, and the excess burst (Be) size. Packets within a class can be marked with different drop-precedence bits, depending whether:

- (a) the current packet's size is less than or equal to the conform size
- (b) the current packet's size is between conform and exceed
- (c) the current packet's size is greater than exceed (violate)

In addition to coloring the packets with the drop-precedence bits, other actions such as marking the packet to another class (a completely different DSCP), transmitting, or dropping the packets may be performed.



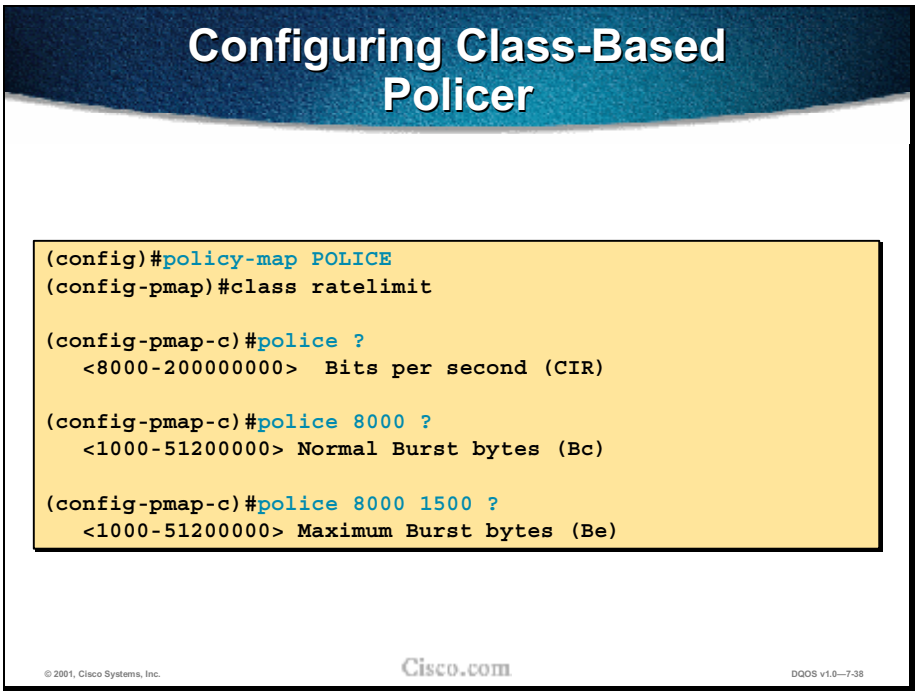
The class-based policer allows you to fully implement the DiffServ-compliant assured forwarding (AF) per-hop behavior (PHB) and along with other MQC tools allows you to construct a DiffServ-compliant network. Policing a class with the conform, exceed, and violate actions (tricolor marker) allows you to simulate weighted random early detection-like (WRED-like) behavior for flows within the class and reduce the chances of a multitude of TCP/UDP sessions being dropped in the presence of congestion. Class-based shaping and class-based policing, combined with class-based WRED, complete the tool chest for non-strict-priority classes. These mechanisms can be used to implement the DiffServ AF PHB, as well as to perform traffic conditioning (policing or shaping).



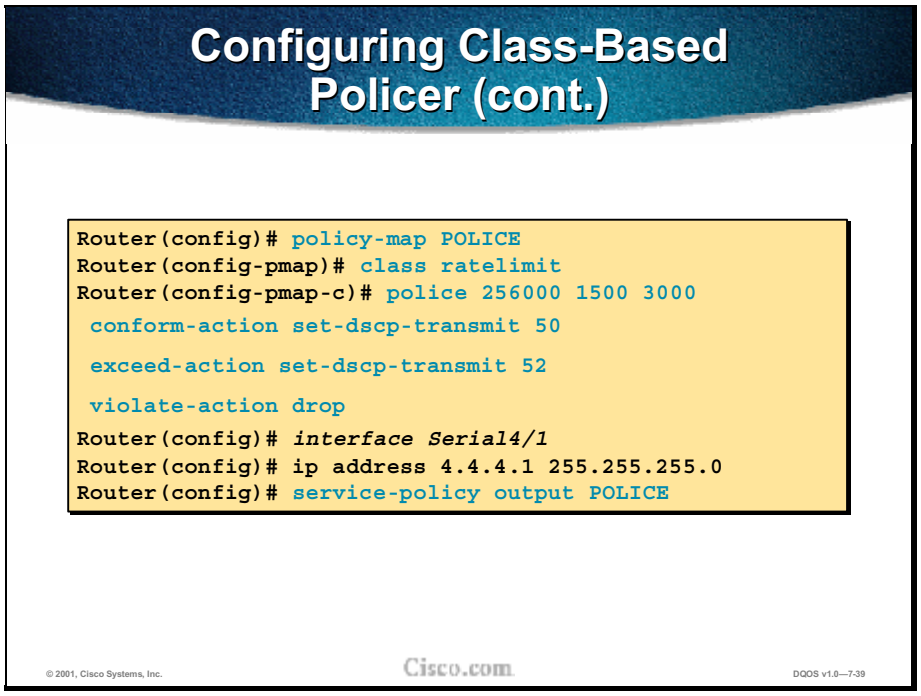
Rate limiting can be done for traffic based on its source/destination address (use ACLs for this), DSCP, IP Precedence, or QoS group. Traffic is classified based on DSCP values. Minimum bandwidth is guaranteed in the times of congestion; at the same time, rate limiting will be enforced if traffic exceeds assigned CIR.

Matching criteria for identification of traffic for rate limiting, DSCP/precedence setting, or both includes:

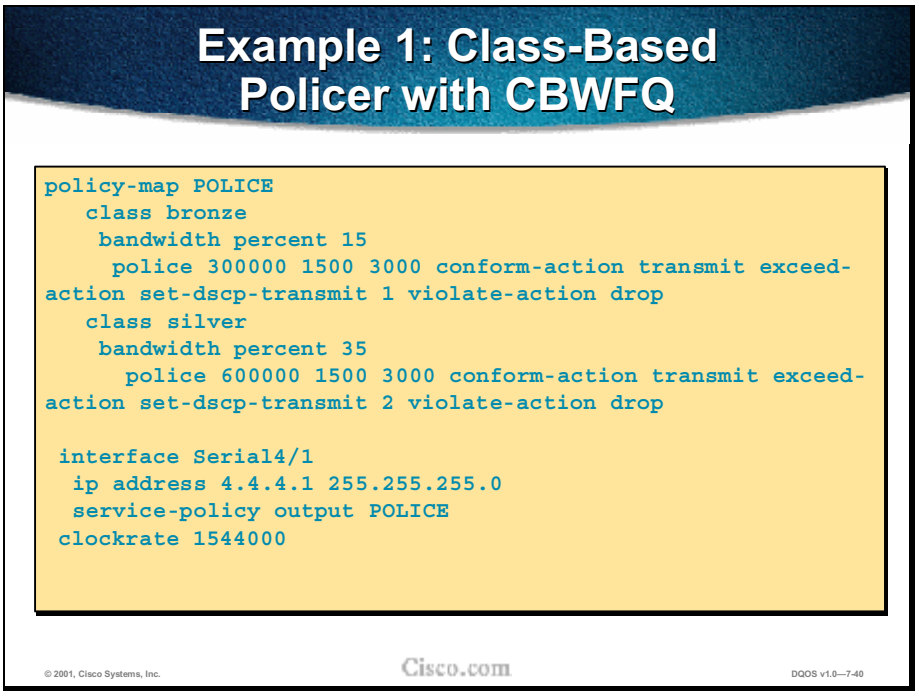
- Incoming/outgoing interface
- All/any IP traffic
- DSCP or IP Precedence value
- Standard or extended source/destination access list
- IP RTP ports
- 0-99 QoS group IDs
- CoS value
- Predefined class maps
- MPLS experimental value
- Source/destination MAC address
- NBAR protocols



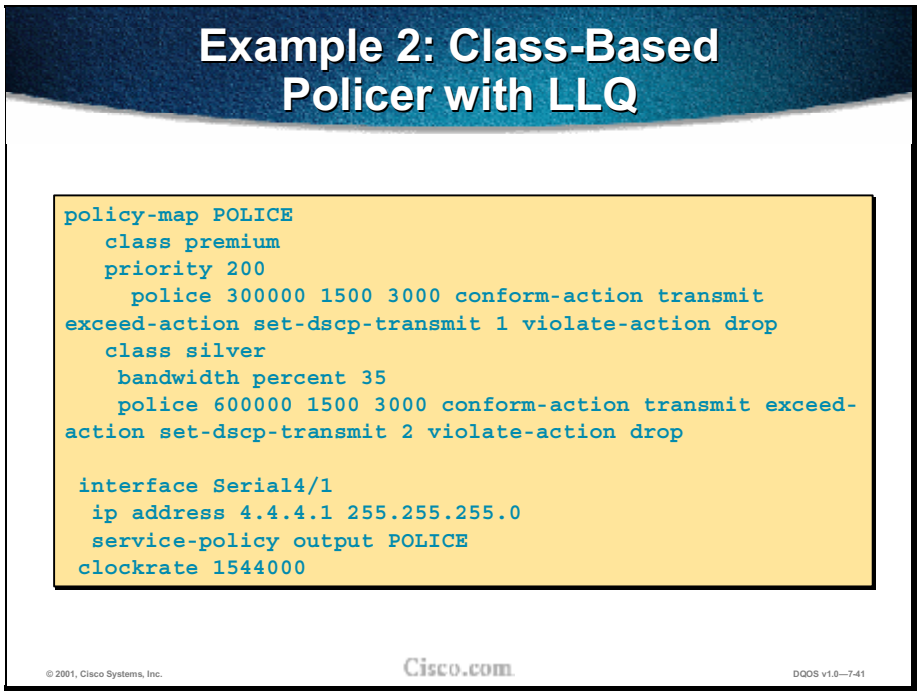
In the above configuration steps, the traffic-policing configuration is associated with the match criteria from the traffic class. The traffic-policing configuration is configured in the service policy (using the **policy-map** command). The **service-policy** command is then used to attach the service policy to the interface.



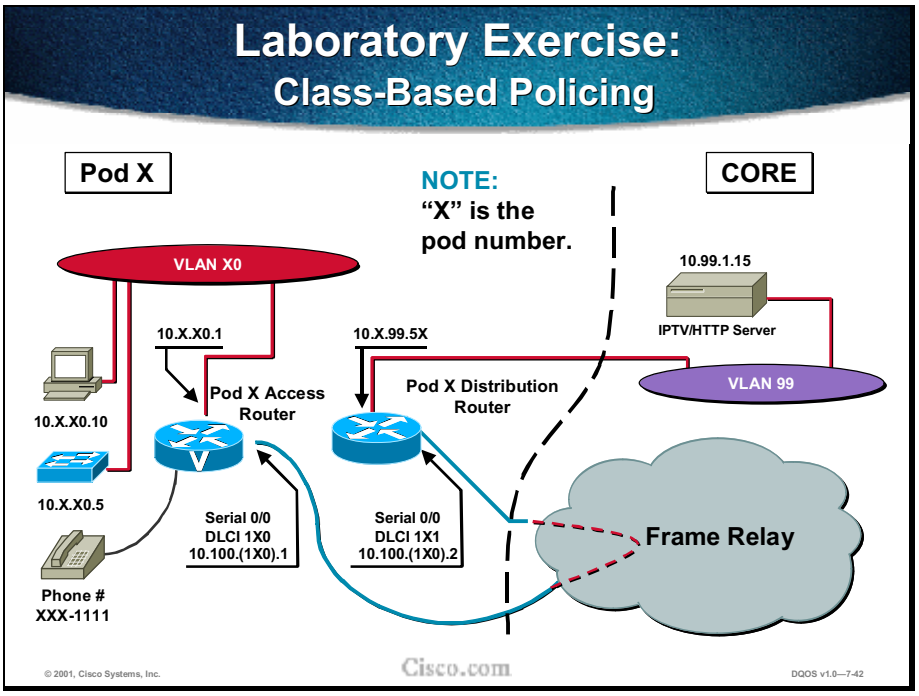
In this example, the conform action is to set the DSCP to 50 (110010—low drop precedence) and transmit. The exceed action is to set the DSCP to 52 (110100—high drop precedence) and transmit. Violate action is to drop.



In this example, for bronze class, guarantee minimum 15% with interface BW being 1544000, 173700 bps is guaranteed, yet traffic is policed for CIR of 300 bps.



In this example, priority queue is guaranteed 200 kbps during congestion. However, during a noncongestion period it cannot go over 300 kbps. Excess traffic will be reprioritized.




Traffic Shaping
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—7-43

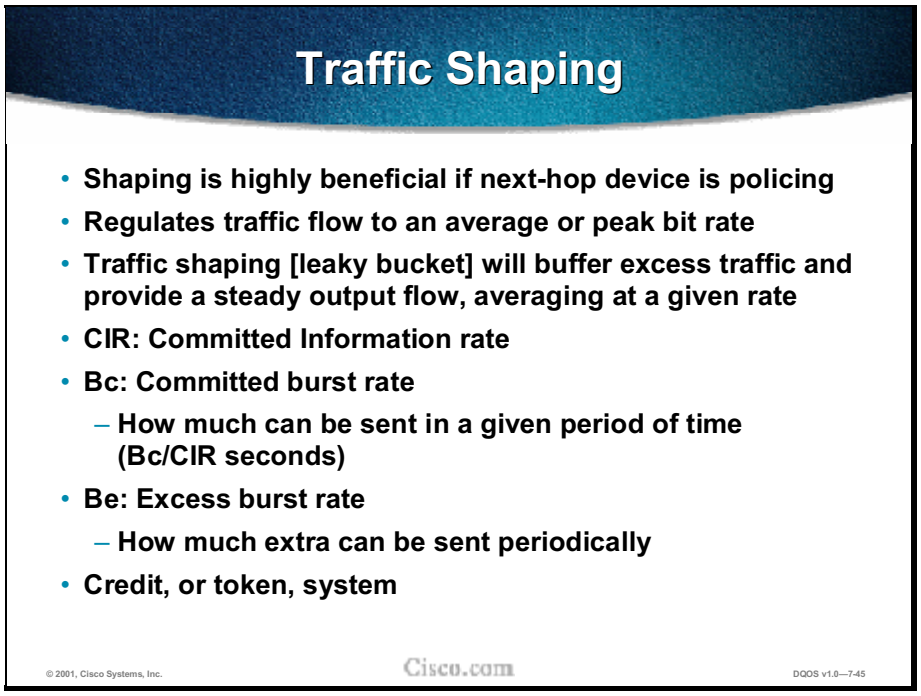
The primary reasons to use traffic shaping are to control access to available bandwidth, to ensure that traffic conforms to the policies established for it, and to regulate the flow of traffic in order to avoid congestion that can occur when the transmitted traffic exceeds the access speed of its remote, target interface. Here are some examples:

- Control access to bandwidth when, for example, policy dictates that the rate of a given interface should not on the average exceed a certain rate, even though the access rate exceeds the speed.

- Configure traffic shaping on an interface if you have a network with differing access rates. Suppose that one end of the link in a Frame Relay network runs at 256 kbps and the other end of the link runs at 128 kbps. Sending packets at 256 kbps could cause failure of the applications using the link.

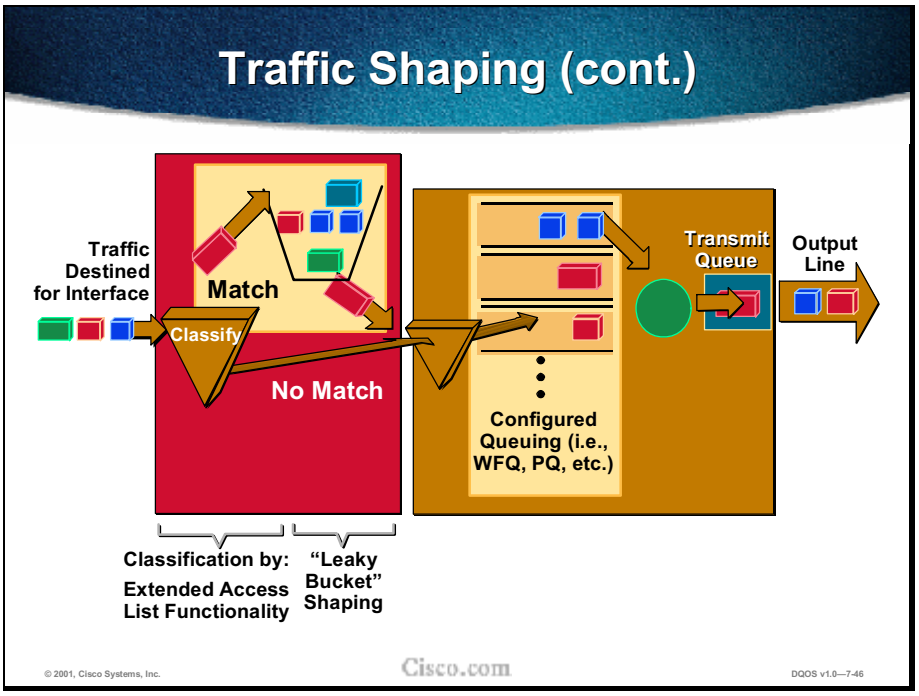
- A similar, more complicated case would be a link-layer network giving indications of congestion that has differing access rates on different attached data terminal equipment (DTE); the network may be able to deliver more transit speed to a given DTE at one time than another. (This scenario warrants that the token bucket be derived, and then its rate maintained.)

- Configure traffic shaping if you offer a subrate service. In this case, traffic shaping enables you to use the router to partition your T1 or T3 links into smaller channels.

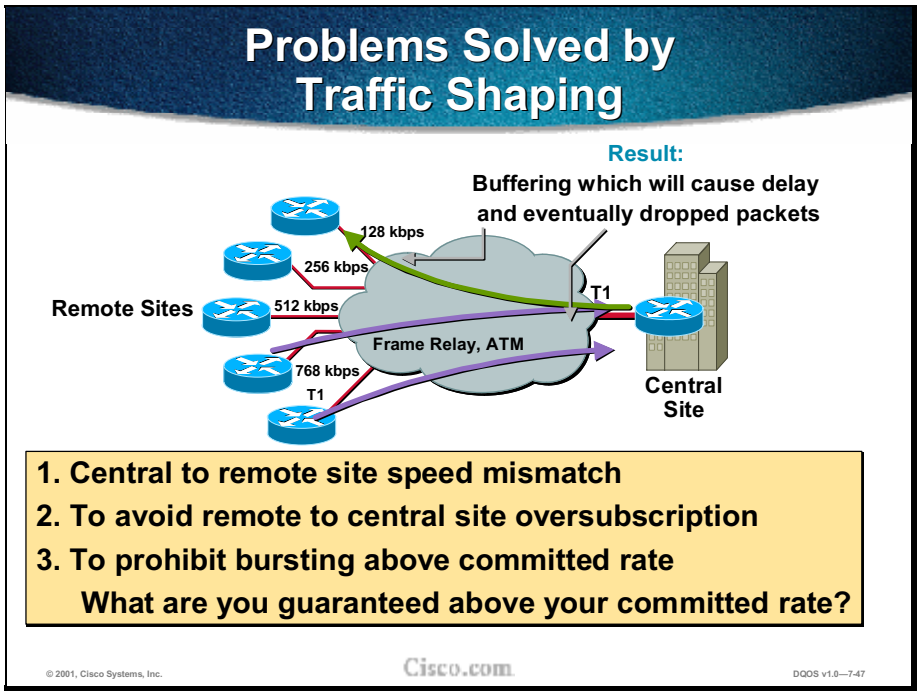


In addition to some form of priority queuing mechanism, Frame Relay traffic shaping (FRTS) allows better utilization of the contracted parameters by staying with the CIR and burst parameters.

In combination, priority queuing and Frame Relay traffic shaping generally provide an acceptable level of QoS for voice, even in single PVC environments.



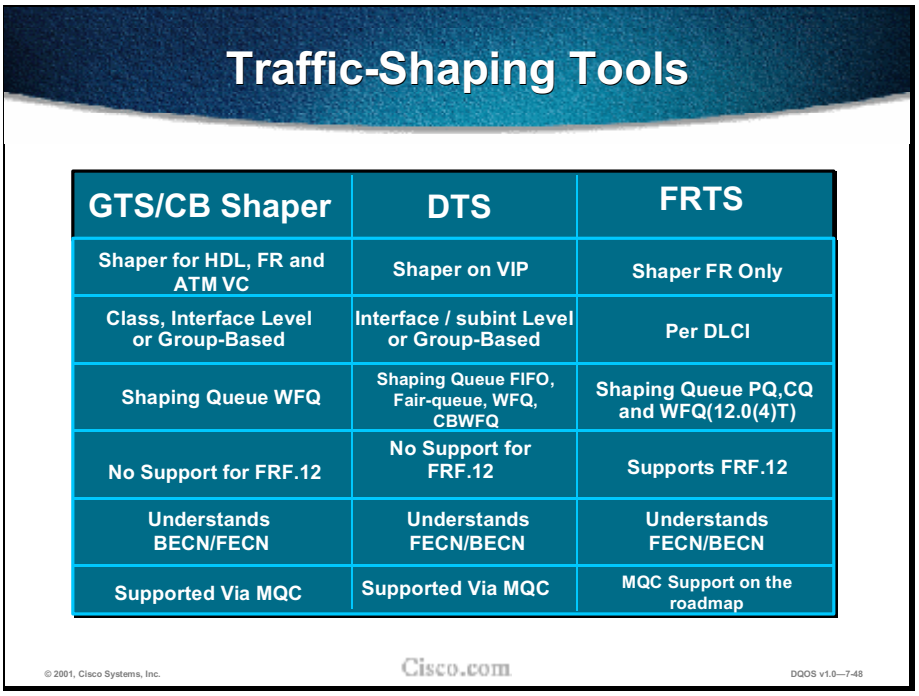
Traffic shaping allows control of the traffic going out an interface in order to match its flow to the speed of the remote, target interface and to ensure that the traffic conforms to policies contracted for it. Thus, traffic adhering to a particular profile can be shaped to meet downstream requirements, thereby eliminating bottlenecks in topologies with data-rate mismatches.



The figure shows a Frame Relay or ATM network. Pay close attention to the speeds of the access lines to the remote sites on the left. Suppose that each site has a CIR close to the access speed, with bursting up to the access bandwidth.

- What happens at the central site if the bottom two sites burst at the same time?
- What happens at the central site if a server rapidly transmits data for the top-left remote site?
- What happens if the two bottom-left sites try to send a large amount of data to the top-left site?

This section describes some of the QoS techniques for managing this type of issue.



Cisco has long provided support for forward explicit congestion notification (FECN) for DECnet and OSI, and BECN for Systems Network Architecture (SNA) traffic using LLC2 encapsulation via RFC 1490 and DE bit support.

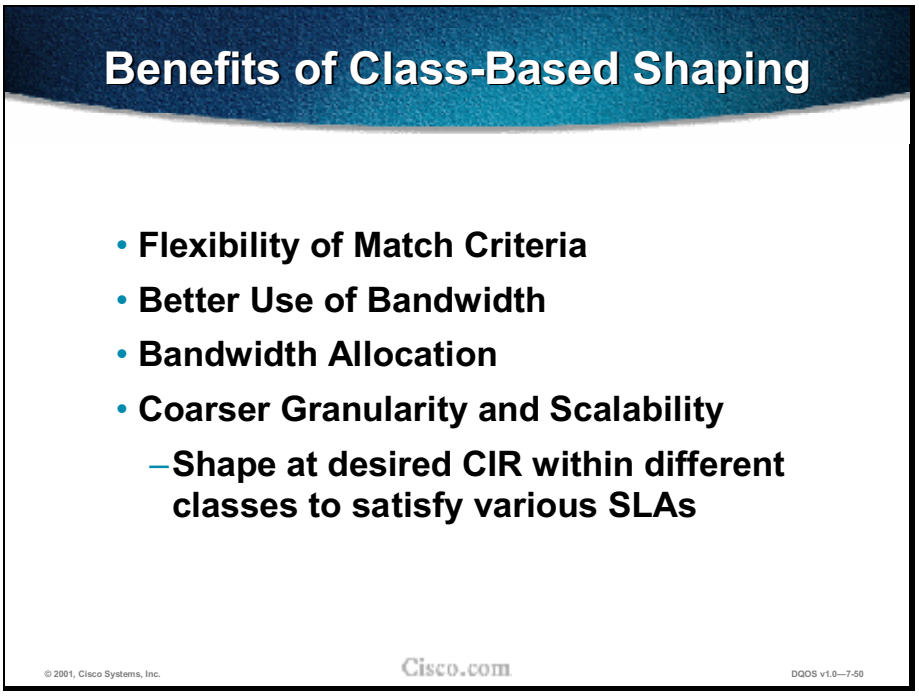
Frame Relay traffic shaping (FRTS) builds upon this existing Frame Relay support with additional capabilities that improve the scalability and performance of a Frame Relay network, increasing the density of virtual circuits and improving response time.

As is also true of GTS, FRTS can eliminate bottlenecks in Frame Relay networks that have high-speed connections at the central site and low-speed connections at branch sites. You can configure rate enforcement—a peak rate configured to limit outbound traffic—to limit the rate at which data is sent on the VC at the central site.

Using FRTS, you can configure rate enforcement to either the CIR or some other defined value, such as the excess information rate, on a per-VC basis. The ability to allow the transmission speed used by the router to be controlled by criteria other than line speed (that is, by the CIR or the excess information rate) provides a mechanism for sharing media by multiple VCs. You can preallocate bandwidth to each VC, creating a virtual time-division multiplexing network.

You can also define PQ, CQ, and WFQ at the VC or subinterface level. Using these queuing methods allows for finer granularity in the prioritization and queuing of traffic, providing more control over the traffic flow on an individual VC. If you combine CQ with the per-VC queuing and rate enforcement capabilities, you enable Frame Relay VCs to carry multiple traffic types such as IP, SNA, and Internetwork Packet Exchange (IPX) with bandwidth guaranteed for each traffic type.


Class-Based Shaping
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—7-49



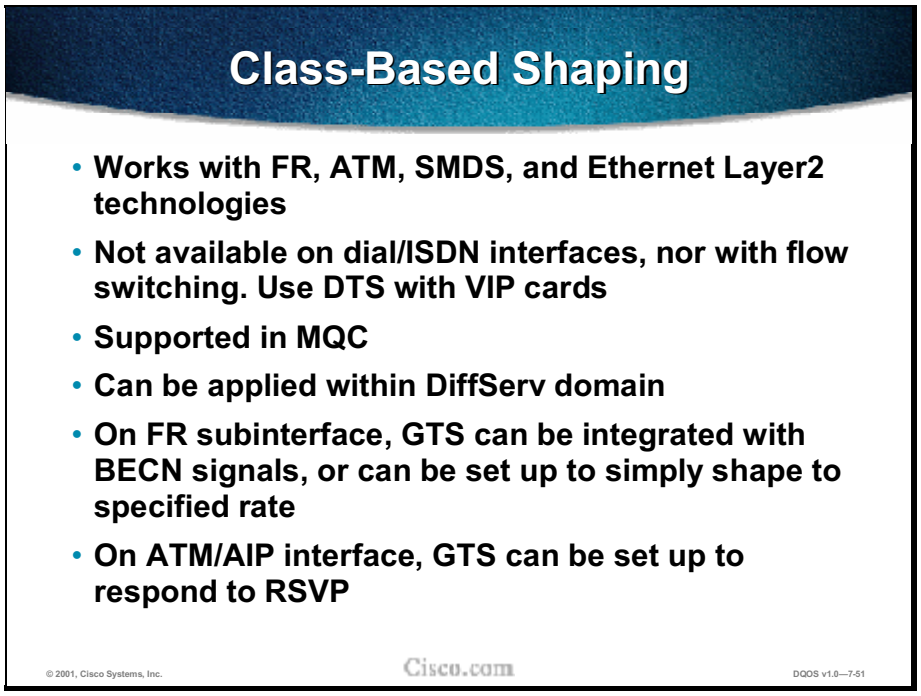
Applying generic traffic shaping (GTS) to classes provides greater flexibility for configuring traffic shaping. Previously this ability was limited to the use of ACLs.

Specifying peak rate shaping allows you to make better use of available bandwidth by allowing more data than the CIR to be sent if the bandwidth is available.

Class-based weighted fair queuing (CBWFQ) allows the administrator to specify the exact amount of bandwidth to be allocated for a specific class of traffic. When available bandwidth on the interface is taken into account, up to 64 classes can be configured and distribution can be controlled among them, which is not the case with flow-based weighted fair queuing (WFQ).

Flow-based WFQ applies weights to traffic to classify it into conversations and determine how much bandwidth each conversation is allowed relative to other conversations. These weights, and traffic classification, are dependent on and limited to the seven IP Precedence levels.

CBWFQ allows the network administrator to define what constitutes a class, based on criteria that exceed the confines of flow. CBWFQ allows the use of ACLs and protocols or input interface names to define how traffic will be classified, thereby providing coarser granularity. Traffic classification does not need to be maintained on a flow basis. Moreover, up to 64 discrete classes can be configured in a service policy.



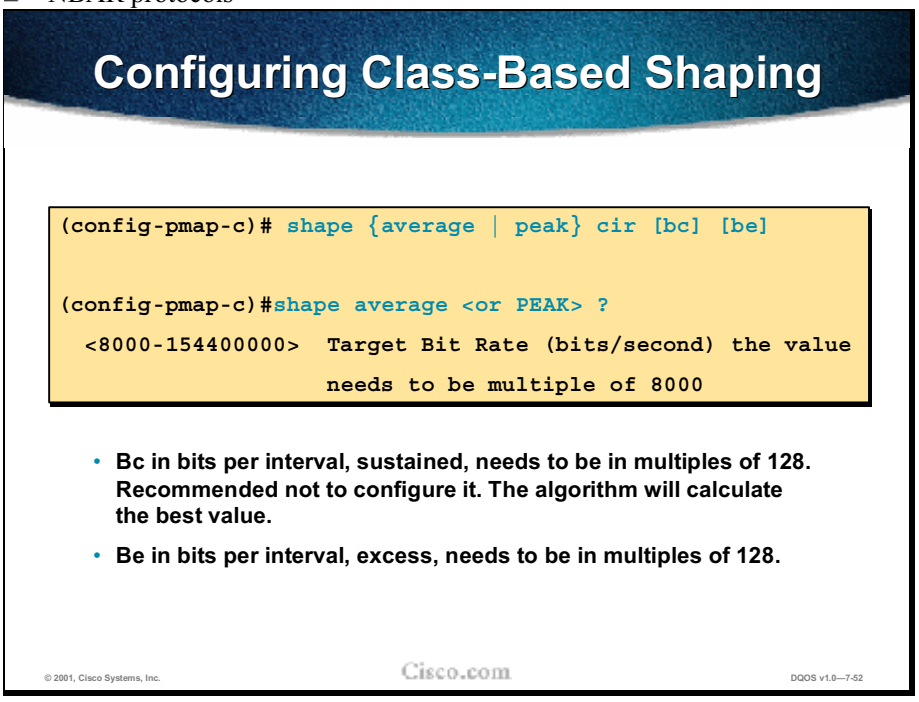
Key features of class-based traffic shaping are described in the figure.

Using the class-based shaping feature, class-based WFQ (CBWFQ) is supported for the queued packets. Using CBWFQ, it is possible to configure classes of queued traffic and provide relative or absolute bandwidth guarantees to those classes. Note that the relative or absolute bandwidth guarantees are with regard to the configured CIR.

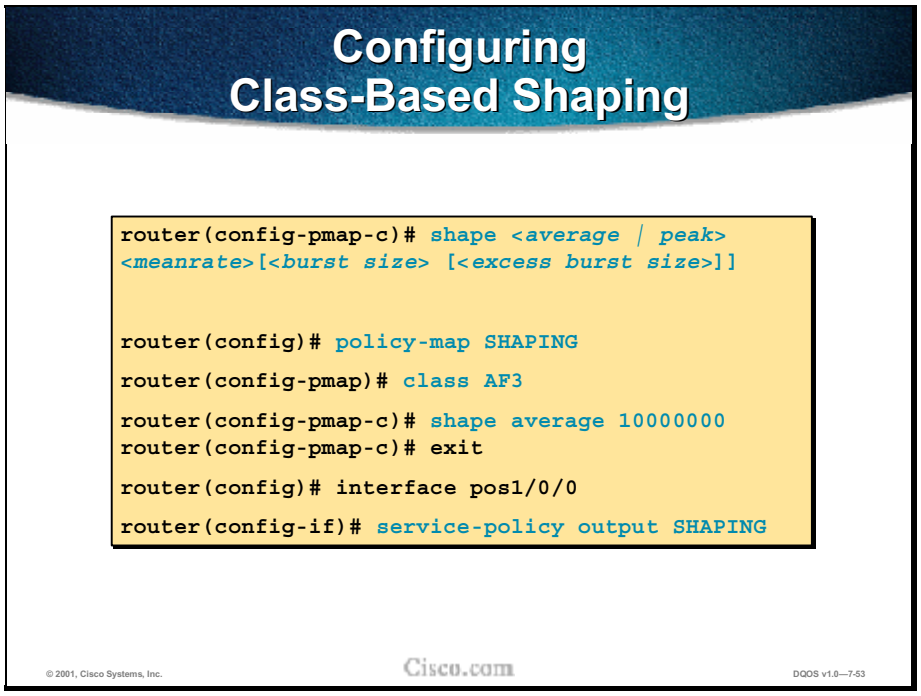
Traffic matching entails identification of traffic of interest for rate limiting, precedence setting, or both. Matching criteria for class-based shaping are:

- Matching criteria stated in MQC format
- Outgoing interface
- All/any IP traffic
- DSCP or IP Precedence value
- Standard or extended source/destination ACLs
- IP RTP ports
- 0-99 QoS group IDs
- CoS value
- Predefined class maps

- MPLS experimental value
- Source/destination MAC address
- NBAR protocols



Use the **shape** configuration command to shape traffic to the indicated bit rate according to the algorithm specified. It is configured in Policy map class configuration mode and was introduced in 12.0(5)XE—it became available for Cisco IOS Release 12.1 T in 12.1(5)T.



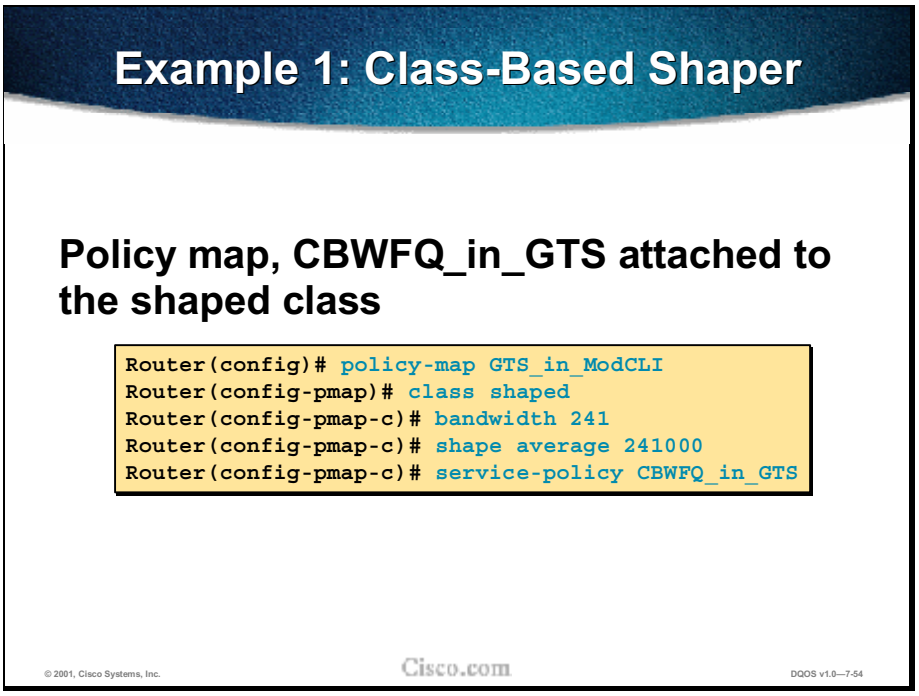
Usage Guidelines

The measurement interval is Bc/CIR. Bc cannot be set to 0. If it is too large (greater than 128 ms), the system subdivides it into smaller intervals.

If you do not specify Bc and Be, the algorithm decides the default values for the shape entity. The algorithm uses a 4-millisecond measurement interval, so Bc will be CIR * 4/1000.

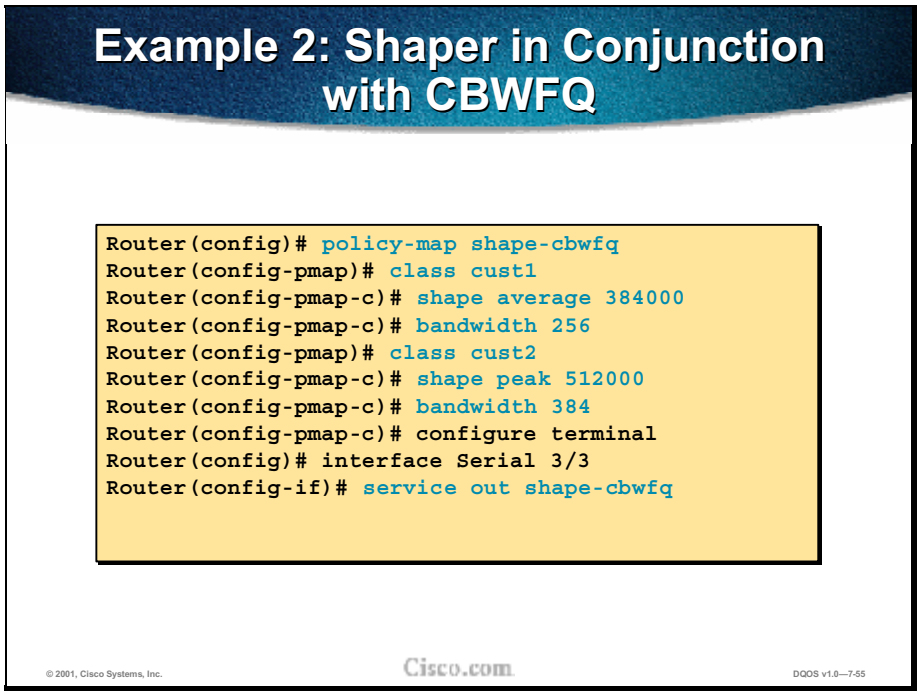
Burst sizes larger than the default Bc need to be explicitly specified. The larger the Bc, the longer the measurement interval. This may affect voice traffic latency, if applicable.

When Be is not configured, the default value is equal to Bc.



In this example, the class-based shaping feature is configured for the class named shaped in the policy map named GTS_in_ModCLI. The class shaped is shaped to an average rate of 241,000 bits per second (bps). CBWFQ is also enabled on the class, which guarantees a bandwidth of 241 kbps during times of congestion at the interface.

The shaped class is a congestion point for all the subclasses that make up that class. Therefore, the subclasses can be further differentiated in the shaped class. All these subclasses are part of the policy map, CBWFQ_in_GTS, which is attached to the shaped class.

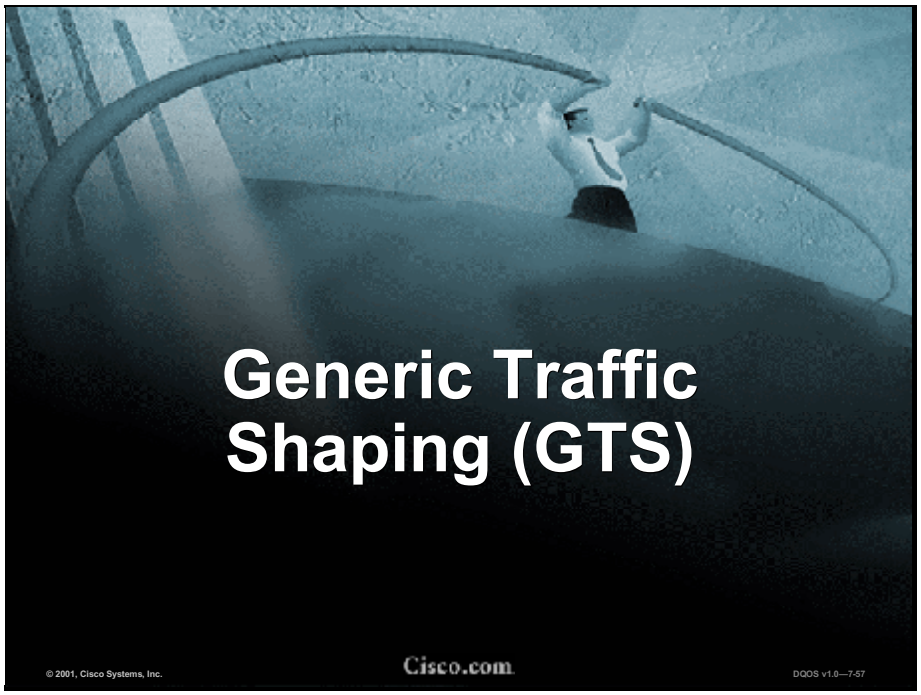


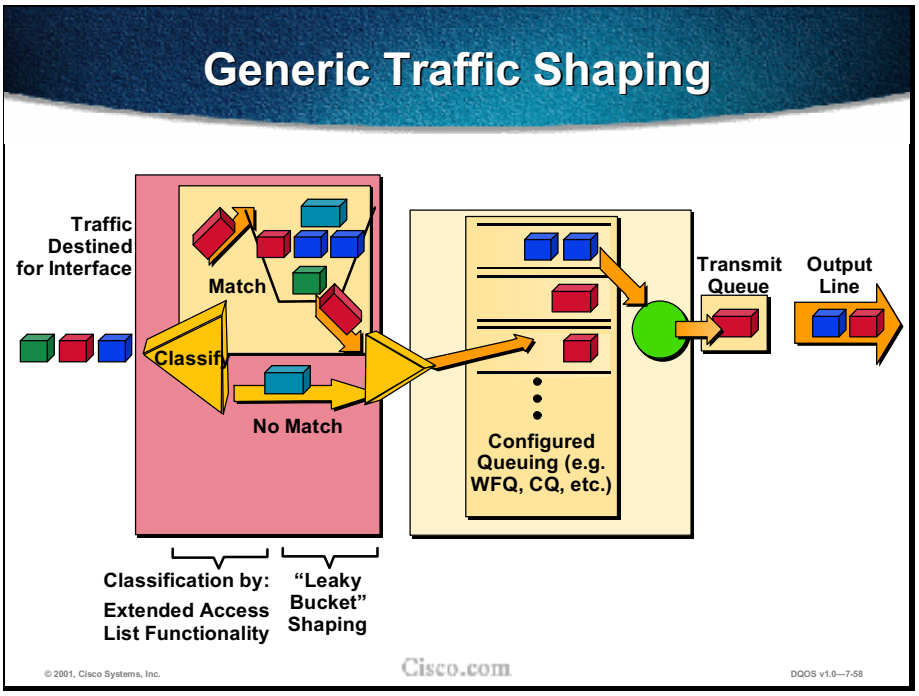
The above example uses CBWFQ at the interface and shapes the traffic before it is queued to CBWFQ.

Two classes are defined—cust1 and cust2. The class cust1 is ensured a bandwidth of 256 kbps, and the output is shaped to 384 kbps. The class cust2 is ensured a bandwidth of 384 kbps, but if enough bandwidth is available on the interface, the class can obtain throughput up to a peak of 512 kbps.

Monitoring Class-Based Shaper

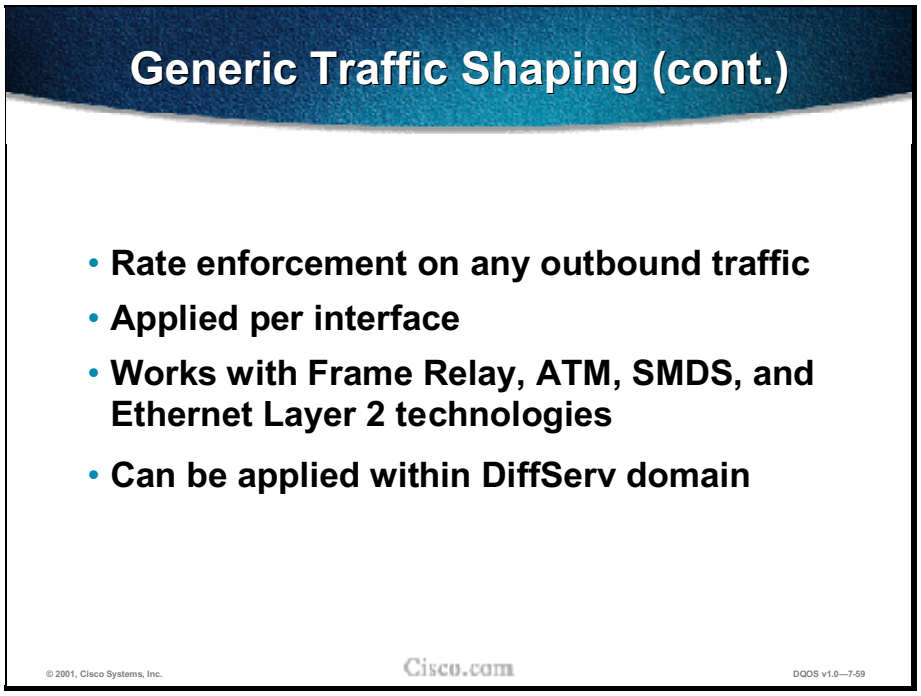
```
Router# show policy-map GTS_in_ModCLI
Policy Map GTS_in_ModCLI
Class shaped
Weighted Fair Queueing
Bandwidth 241 (kbps)  Max Threshold 64 (packets)
Traffic Shaping
Average Rate Traffic Shaping
CIR 241000 (bps) Max. Buffers Limit 1000 (Packets)
Policy Map CBWFQ_in_GTS
Class cust_A
Weighted Fair Queueing
Bandwidth 25 (%)    Max Threshold 64 (packets)
Class cust_B
Weighted Fair Queueing
Bandwidth 25 (%)    Max Threshold 64 (packets)
Class cust_C
Weighted Fair Queueing
Bandwidth 25 (%)    Max Threshold 64 (packets)
Class class-default
Weighted Fair Queueing
Flow based Fair Queueing
```





GTS shapes traffic by reducing outbound traffic flow to avoid congestion by constraining traffic to a particular bit rate using the token bucket mechanism. GTS applies on a per-interface basis and can use access lists to select the traffic to shape. It works with a variety of Layer 2 technologies, including Frame Relay, ATM, Switched Multimegabit Data Service (SMDS), and Ethernet.

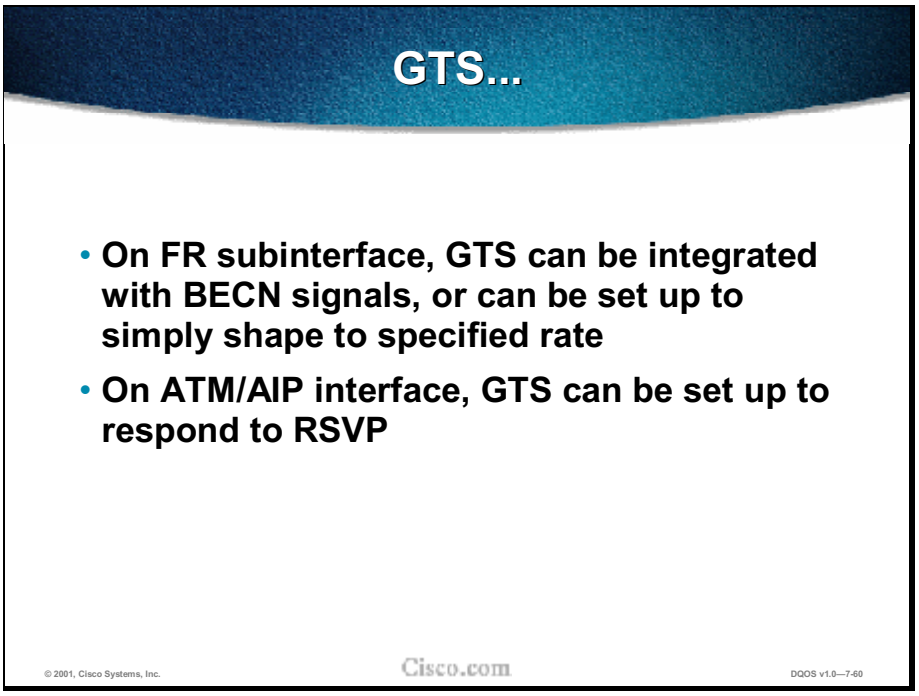
On a Frame Relay subinterface, GTS can be set up to adapt dynamically to available bandwidth by integrating BECN signals or set up simply to shape to a prespecified rate. GTS can also be configured on an ATM AIP model interface to respond to Resource Reservation Protocol (RSVP) signaled over statically configured ATM permanent virtual circuits (PVCs).



GTS provides a mechanism to control the traffic flow on a particular interface. It reduces outbound traffic flow to avoid congestion by constraining specified traffic to a particular bit rate (also known as the token bucket approach), while queuing bursts of the specified traffic. Thus, traffic adhering to a particular profile can be shaped to meet downstream requirements, eliminating bottlenecks in topologies with data-rate mismatches.

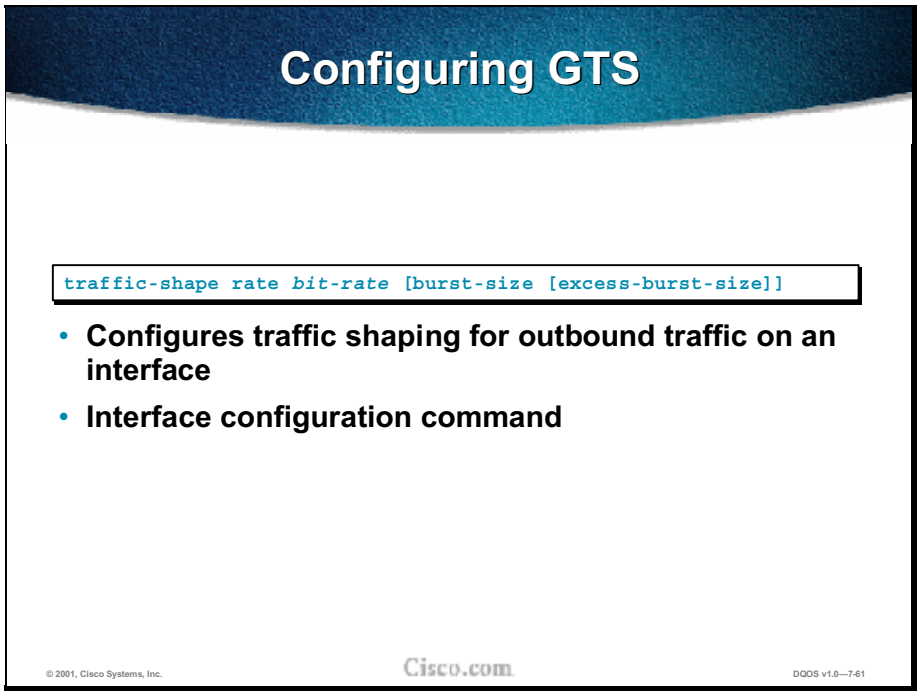
Traffic matching entails identification of traffic of interest for rate limiting, precedence setting, or both. The matching criteria for GTS are:

- Outgoing interface
- All/any IP traffic
- DSCP/IP Precedence
- Defined by rate-limit access list
- MAC address
- Standard or extended IP access list
- Source/destination MAC address

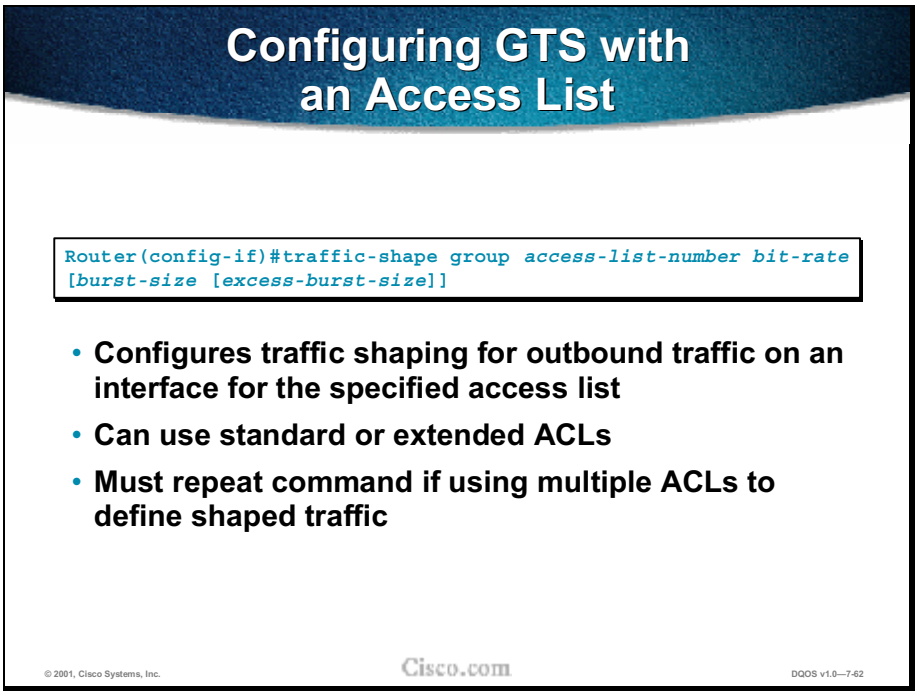


GTS applies on a per-interface basis, can use access lists to select the traffic to shape, and works with a variety of Layer 2 technologies, including Frame Relay, ATM, Switched Multimegabit Data Service (SMDS), and Ethernet.

On a Frame Relay subinterface, GTS can be set up to adapt dynamically to available bandwidth by integrating BECN signals or set up simply to shape to a prespecified rate. GTS can also be configured on an ATM/AIP interface card to respond to RSVP signaled over statically configured ATM permanent virtual circuits (PVCs).

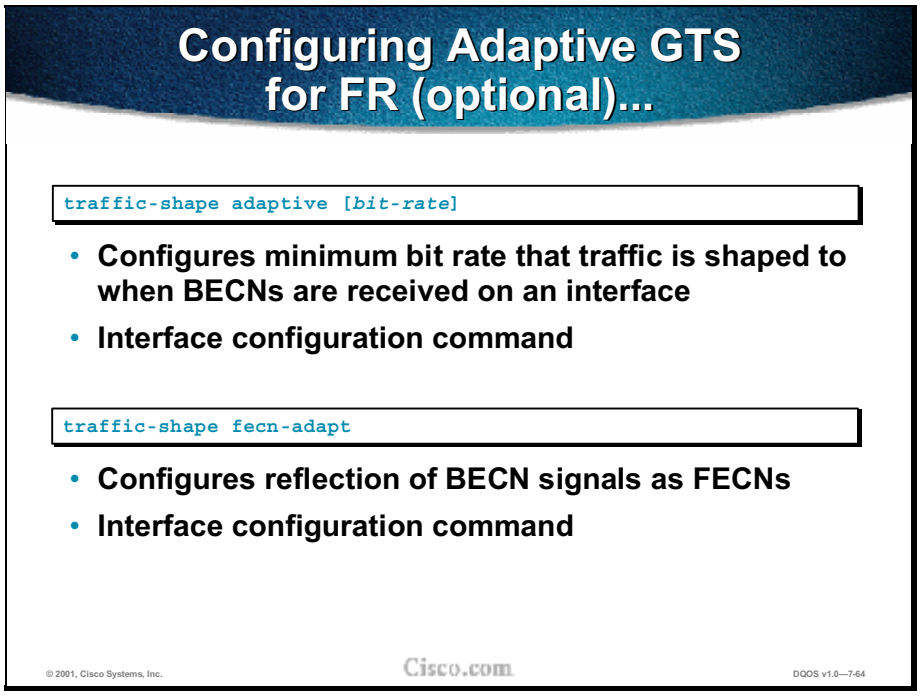


To enable traffic shaping for outbound traffic on an interface, use the **traffic-shape rate** interface configuration command. Of the parameters to be specified, *bit-rate* is the only mandatory one. The *burst-size* and *excess-burst-size* are optional.



To enable traffic shaping based on a specific access list for outbound traffic on an interface, use the **traffic-shape group** interface configuration command. The **traffic-shape group** command allows you to specify one or more previously defined access lists to shape traffic on the interface. You must specify one **traffic-shape group** command for each access list on the interface.

Associated access lists are configured using the standard **access-list** command.



With adaptive GTS the route uses BECNs to estimate available bandwidth and adjust the transmission rate accordingly. Actual maximum transmission rate is between the rate specified by **traffic-shape adaptive** command and that in **traffic-shape rate** command.

GTS Configuration Example 1

```
access-list 101 permit udp any any
interface Ethernet0
 traffic-shape group 101 1000000 125000 125000
!
interface Ethernet1
 traffic-shape rate 5000000 625000 625000
```

GTS Configuration Example 2

GTS on Serial Interface (T1):

```
interface serial 4/1:0
 traffic-shape rate 64000 6400 6400
```

GTS Configuration Example 3

Applying Access Lists

```
access-list 101 permit 10.10.10.10
access-list 102 permit 10.10.10.20
access-list 103 permit 10.10.10.30
!
interface serial 0
 traffic-shape group 101 64000
 traffic-shape group 102 64000
 traffic-shape group 103 256000
```

GTS Configuration Example 4

Using BECN

```
interface serial 2
 traffic-shape rate 1544000
 traffic-shape adaptive 64000
 traffic-shape fecn-adapt
```

GTS Configuration Example 5

Mismatched line speeds

```
interface serial 3
 traffic-shape rate 128000
 traffic-shape adaptive 64000
```

Monitoring GTS

```
show traffic-shape [interface-name]
```

- Shows the current traffic-shaping configuration

```
show traffic-shape statistics [interface-name]
```

- Shows the current traffic-shaping statistics

Sample Outputs

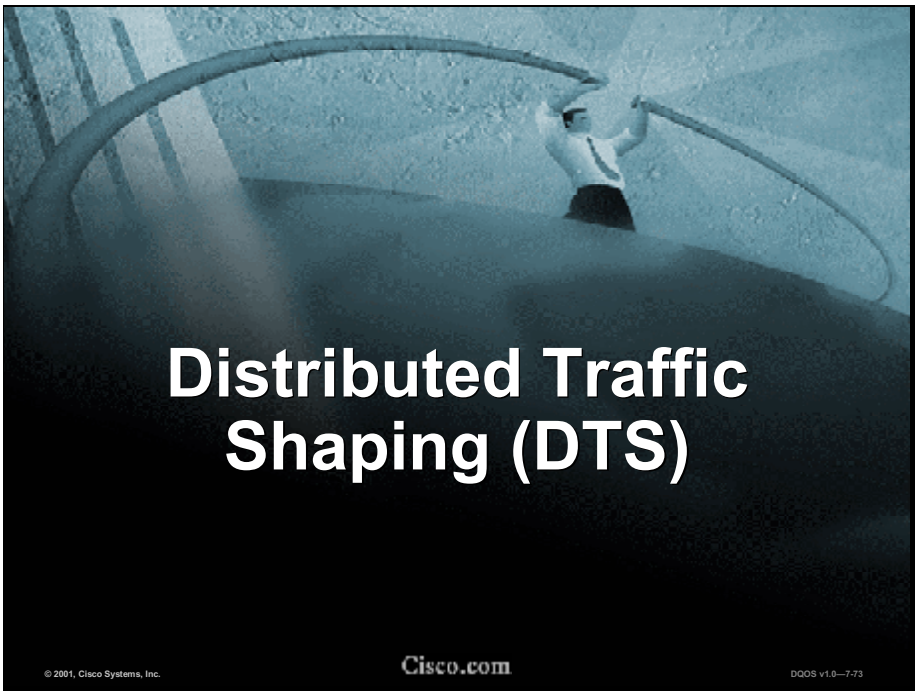
Ethernet 0 is configured to limit UDP traffic to 1 Mbps. Ethernet 1 is configured to limit all output to 5 Mbps

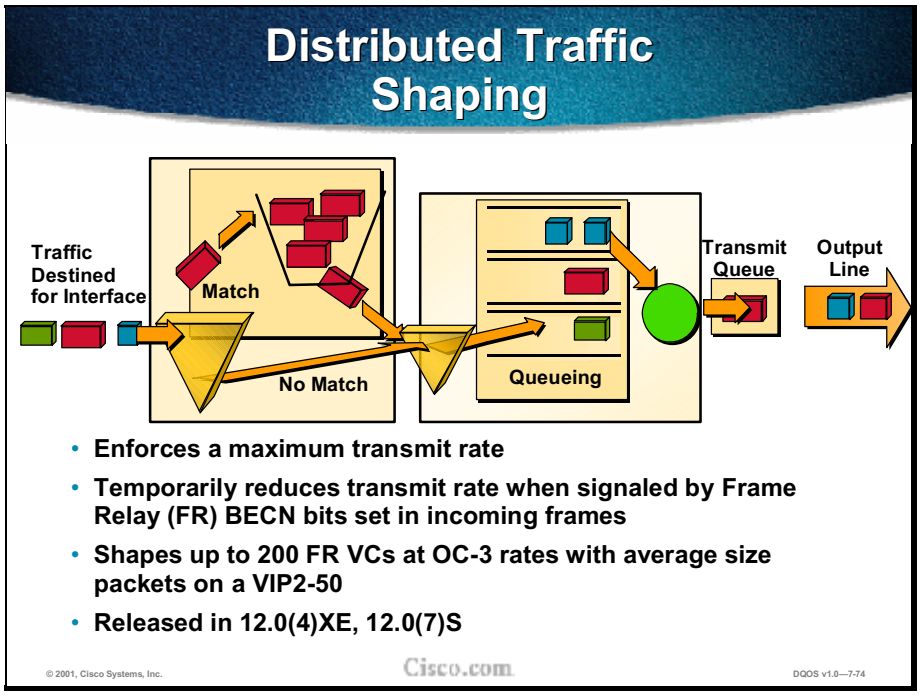
```
Router# show traffic-shape

     access Target  Byte  Sustain  Excess   Interval Increment Adapt
I/F  list   Rate    Limit bits/int bits/int (ms)     (bytes)   Active
Et0  101    1000000 23437 125000   125000    63      7813      -
Et1         5000000 87889 625000   625000    16      9766      -
```

Sample Outputs (cont.)

Ethernet 0 is configured to limit UDP traffic to 1 Mbps. Ethernet 1 is configured to limit all output to 5 Mbps

```
Router# show traffic-shape statistics

      Access Queue   Packets  Bytes  Packets  Bytes     Shaping
I/F   List   Depth                  Delayed  Delayed   Active
Et0   101    0       2        180    0         0        no
Et1          0       0        0      0         0        no
```



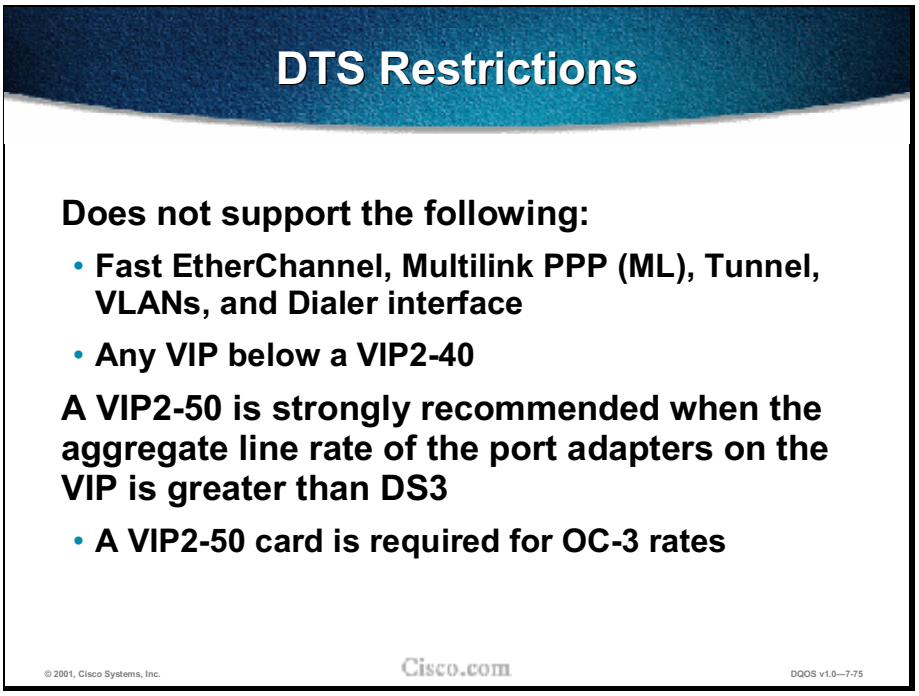


Distributed traffic shaping (DTS) benefits:

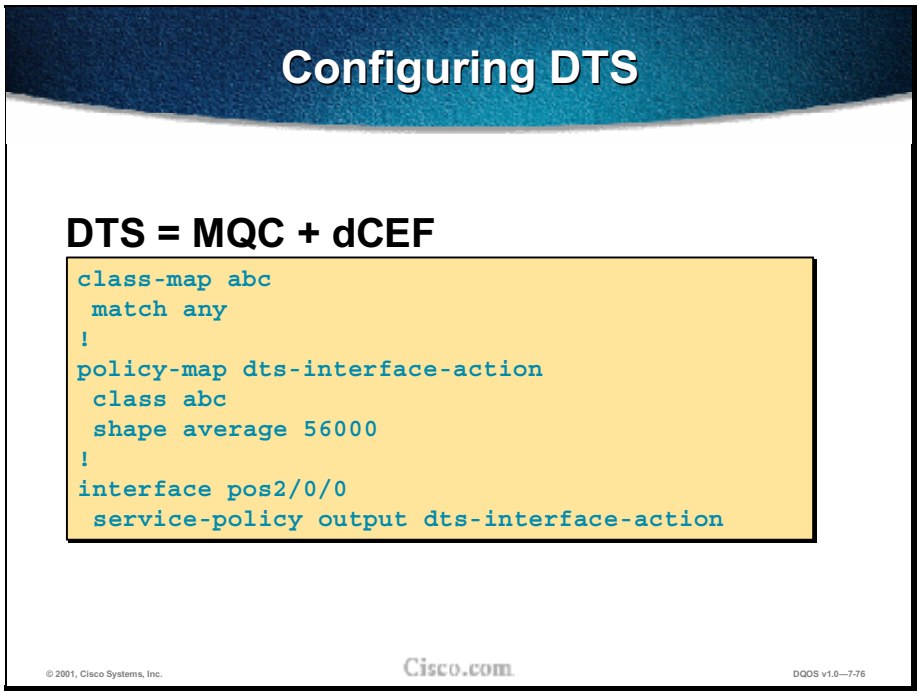
- It offloads traffic shaping from the Route/Switch Processor (RSP) to the Versatile Interface Processor (VIP).
- It supports up to 200 shape queues per VIP, supporting up to OC-3 rates when the average packet size is 250 bytes or greater and when using a VIP2-50 or better with 8 MB of SRAM. Line rates below T3 are supported with a VIP2-40.

Limitations:

- Only IP traffic can be shaped.
- dCEF must be enabled.
- FastEtherChannel, Tunnel, VLAN, and ISDN/Dialer interfaces are not supported.



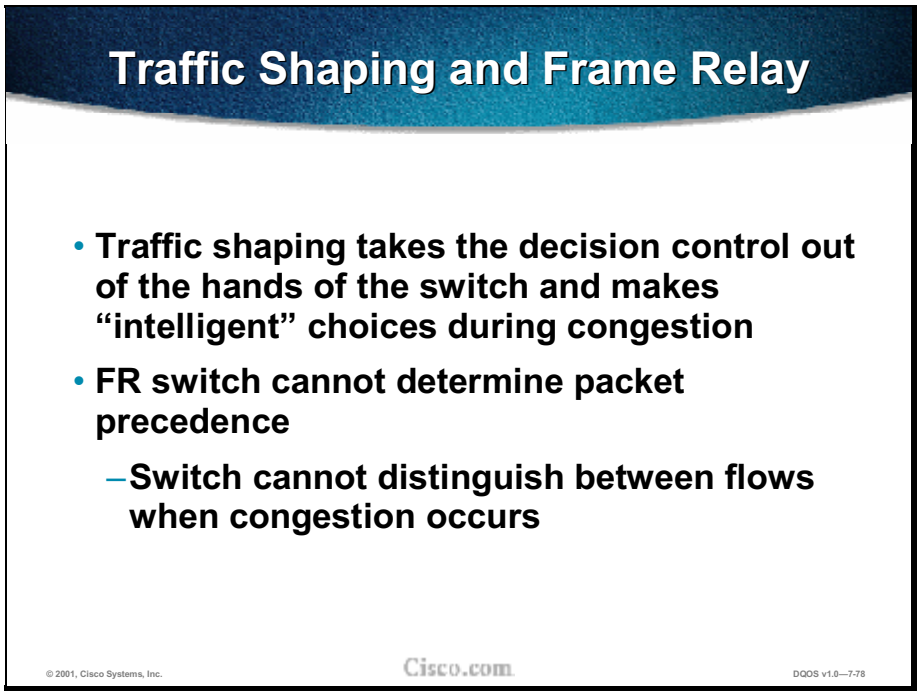
The above restrictions define the interfaces not supported for DTS. The design recommendation is to use the VIP2-50 for high-speed operation.



To enable DTS, you must create a policy map. You can configure class policies for as many classes as are defined on the router, up to the maximum of 256.

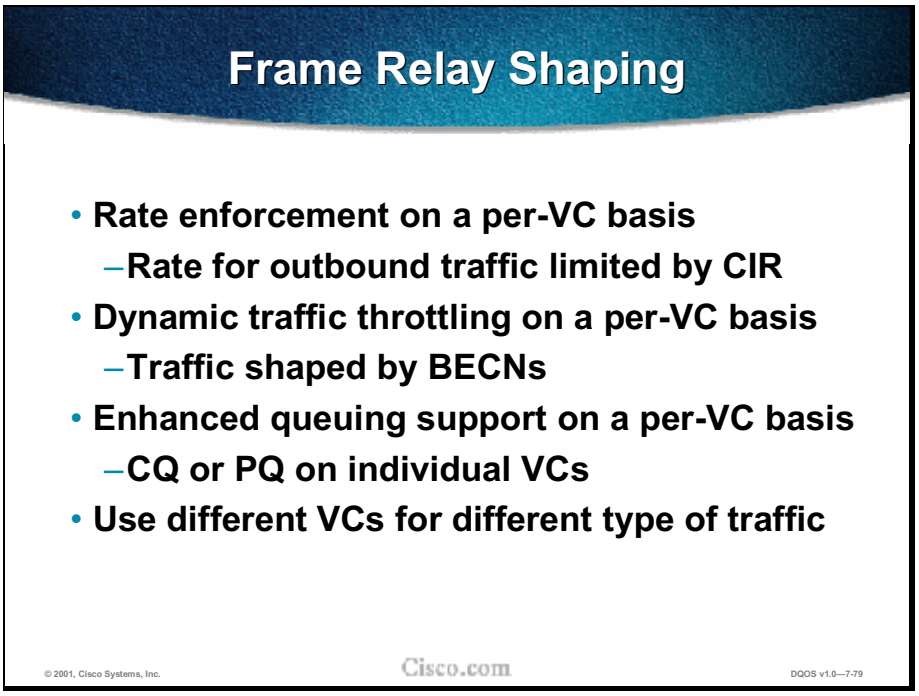
To configure a policy map, use the **policy-map** command to specify the policy map name, then use the following configuration commands to configure class name, traffic shaping, and class policy.

Traffic is directed to the policy map default class if it does not satisfy the match criteria of any other classes whose policies are defined in the policy map.



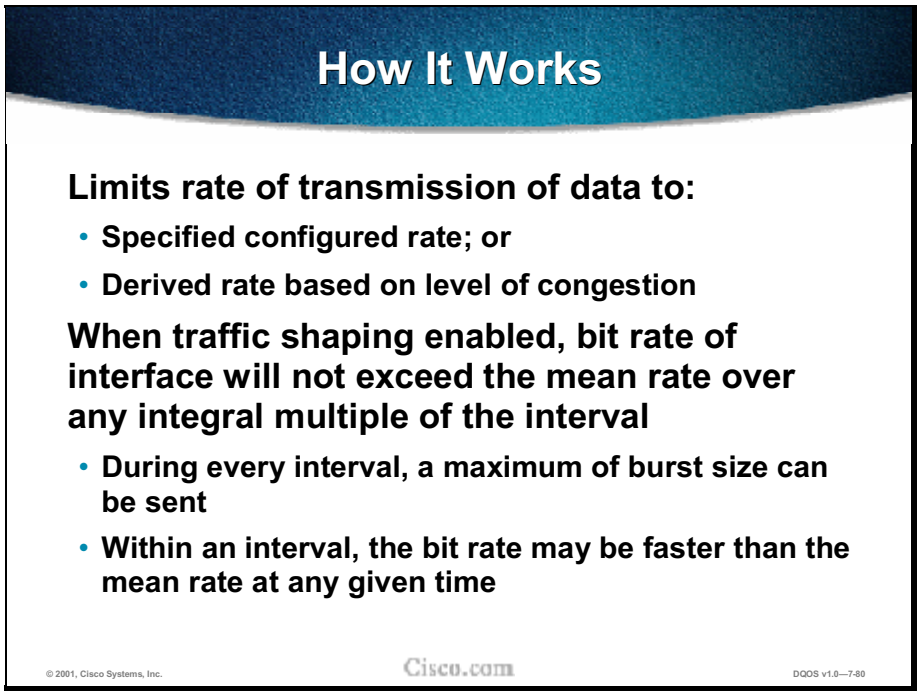
Frame Relay traffic shaping (FRTS) provides parameters that are useful for managing network traffic congestion. These include committed information rate (CIR), forward and backward explicit congestion notification (FECN/BECN), and the discard eligibility (DE) bit.

Without the use of FRTS, the switch at the service provider would make decisions to discard packets that exceed CIR during congestion. Packets within the PVC are considered equal and precedence is not given to one flow over the other. FRTS enables the edge device to set DE based on traffic flow and to ensure that high precedence packets do not get discarded during congestion.



You can configure rate enforcement—a peak rate configured to limit outbound traffic—to either the CIR or some other defined value, such as the excess information rate (EIR), on a per-virtual-circuit (VC) basis.

You can also define priority and custom queuing at the VC or subinterface level. This allows for finer granularity in the prioritization and queuing of traffic and provides more control over the traffic flow on an individual VC. If you combine CQ with the per-VC queuing and rate enforcement capabilities, you enable Frame Relay VCs to carry multiple traffic types, with bandwidth guaranteed for each traffic type.



Mean rate:

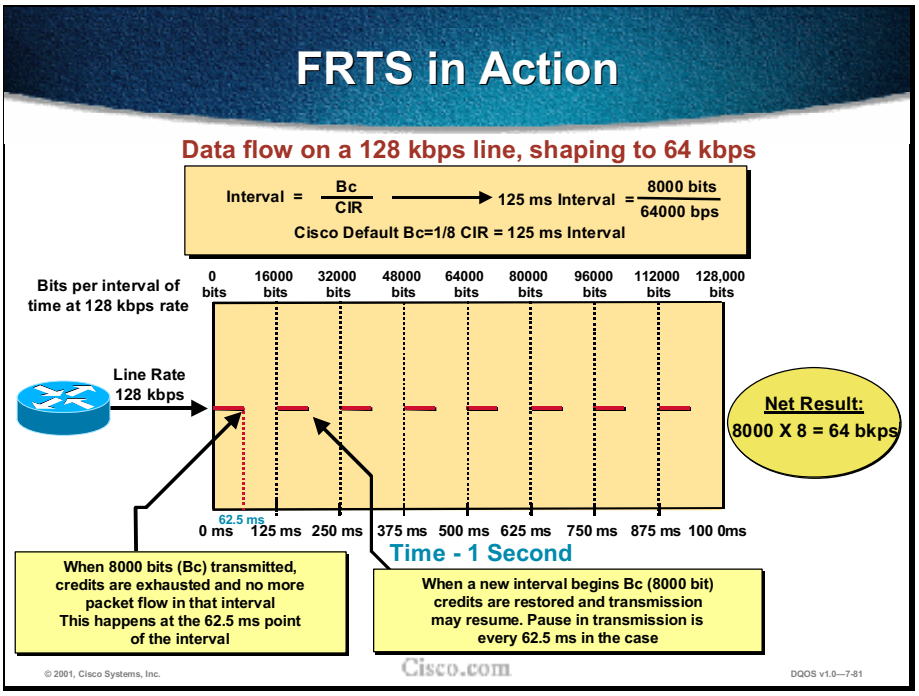
- Committed information rate (CIR)
- Specifies average data sent or forwarded per unit time

Burst size:

- Committed burst size (Bc)
- Specified in bits (or bytes) per burst
- Indicates how much data can be sent within a given unit of time without creating scheduling concerns

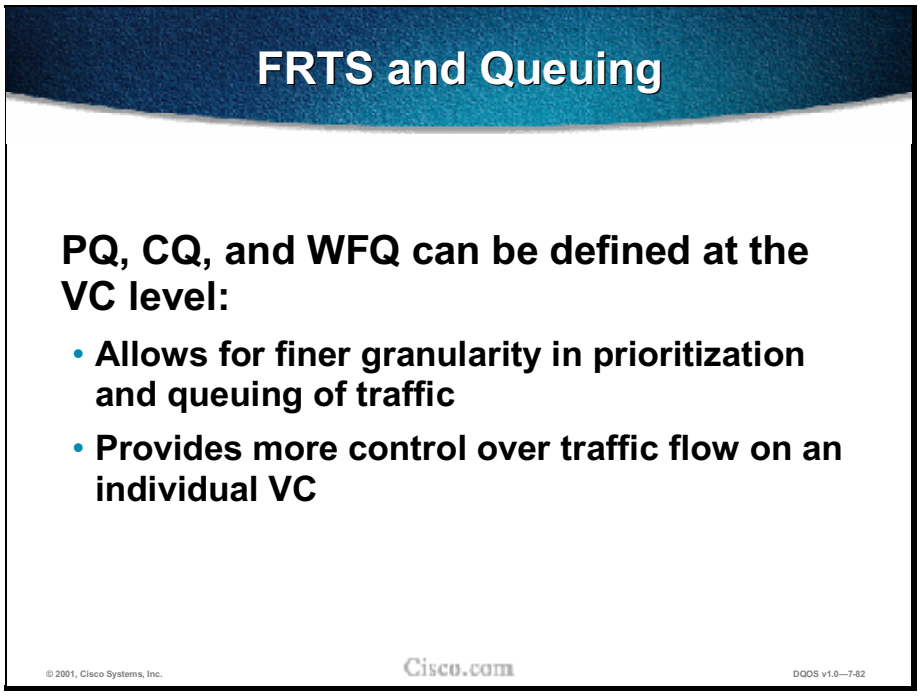
Time interval:

- Measurement interval
- Time quantum in seconds per burst

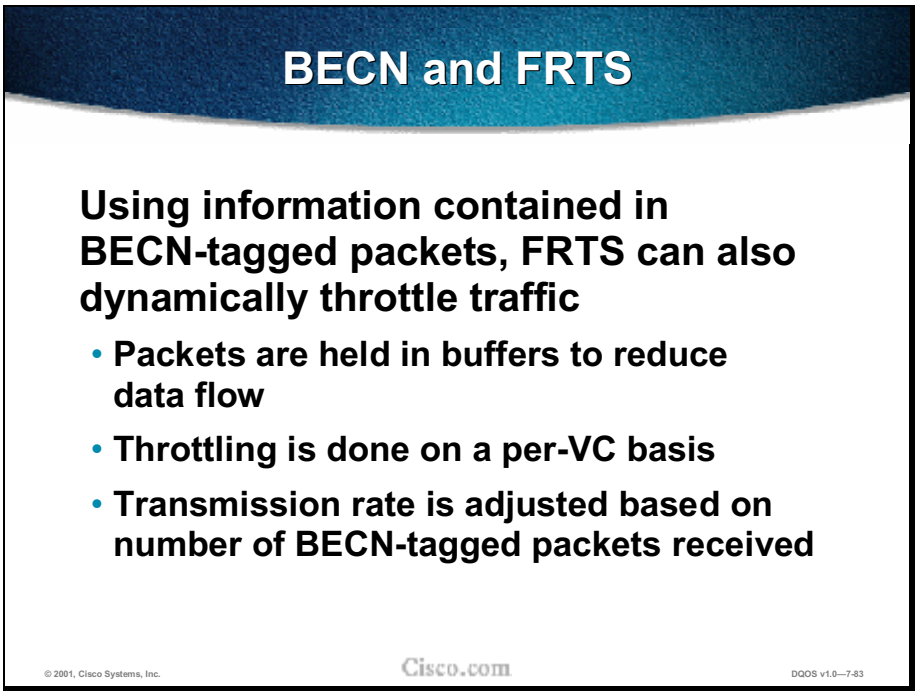


Be (excess burst size) corresponds to the number of bits outside the CIR that are still accepted by the FR switch but are marked as discard eligible. Under certain situations, it allows more than the burst size to be sent during a time interval. Be size values are:

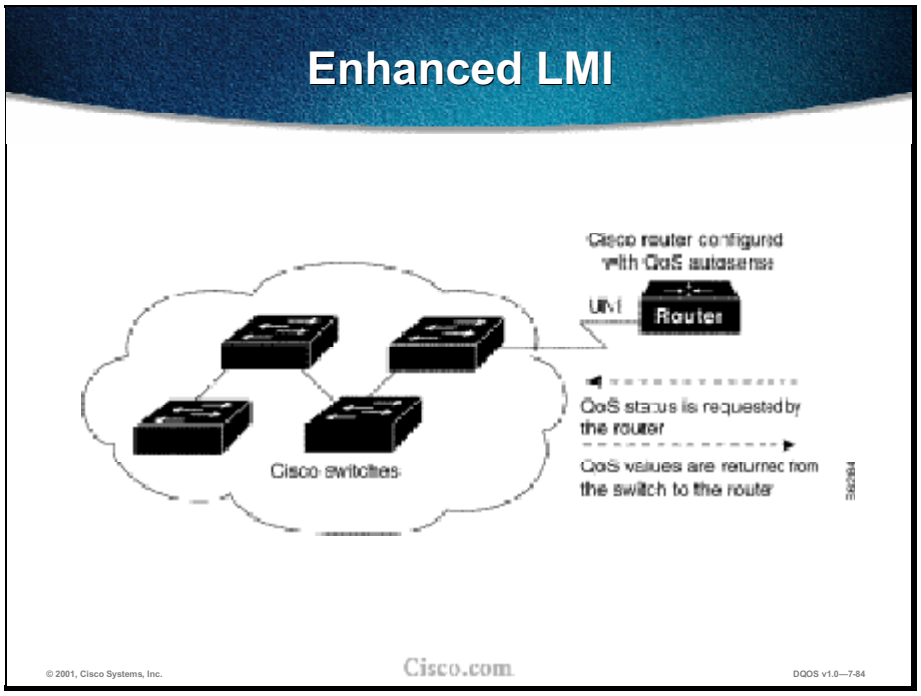
- Equal to 0
  - Interface allows no more than burst size every interval
    - Average rate no higher than mean rate
- Greater than 0
  - Interface can send as many as Bc+Be bits in a burst if maximum amount was not sent in previous time interval
    - Whenever less than Bc is sent in interval, remaining number of bits (up to Be) can be used to send more than Bc in later interval



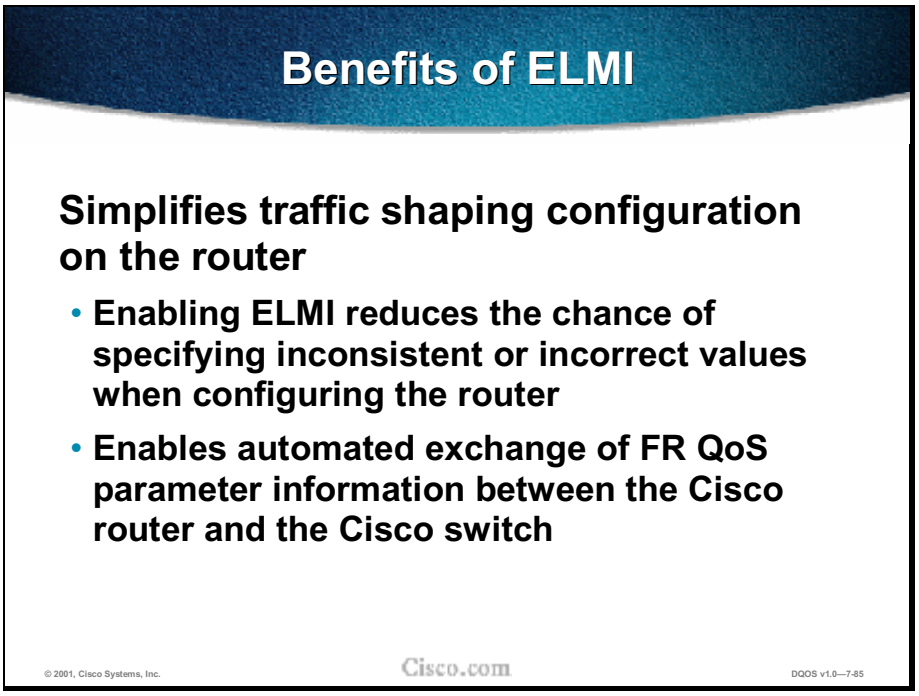
For example, if you combine CQ with the per-VC queuing and rate-enforcement capabilities, you can enable FR VCs to carry IP, SNA, and IPX traffic with bandwidth guaranteed for each.



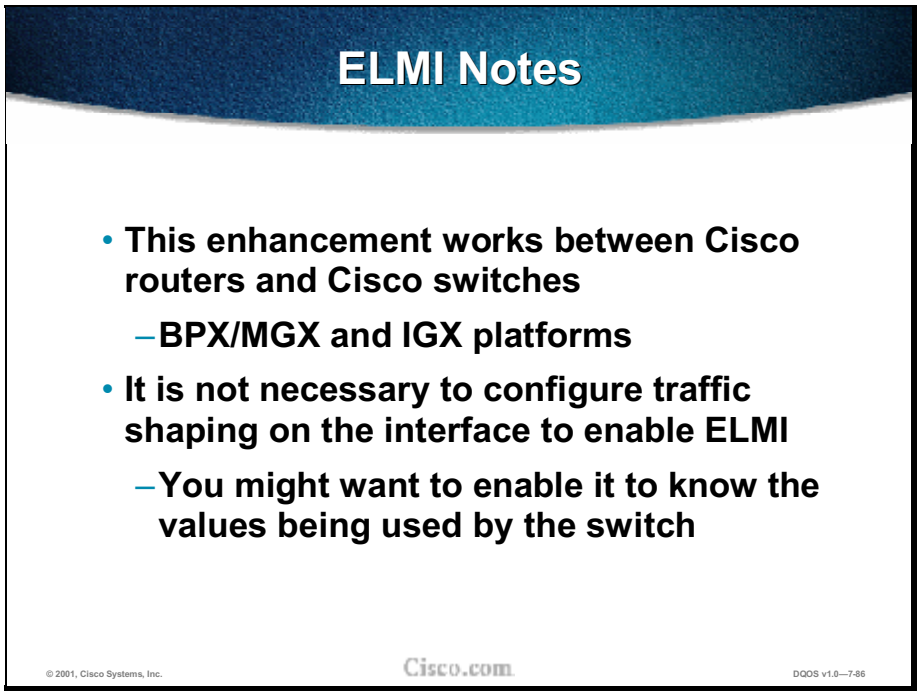
Using information contained in BECN-tagged packets received from the network, FRTS can also dynamically throttle traffic. With BECN-based throttling, packets are held in the router's buffers to reduce the data flow from the router into the Frame Relay network. The throttling is done on a per-VC basis, and the transmission rate is adjusted based on the number of BECN-tagged packets received.



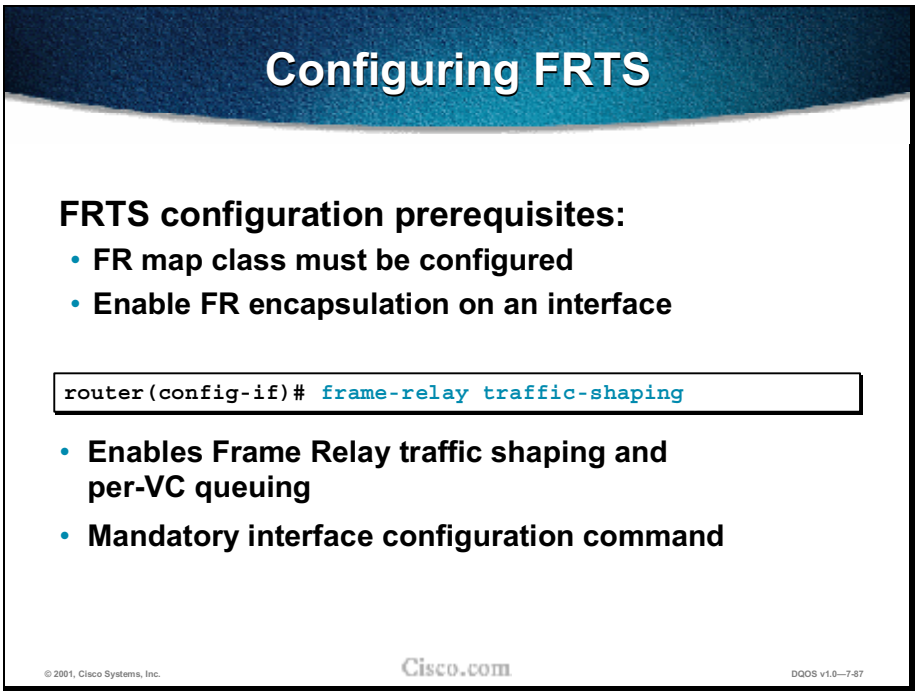
When used in conjunction with traffic shaping, the router can respond to changes in the network dynamically. This optional feature allows the router to learn QoS parameters from the Cisco switch and use them for traffic shaping, configuration, or management purposes.



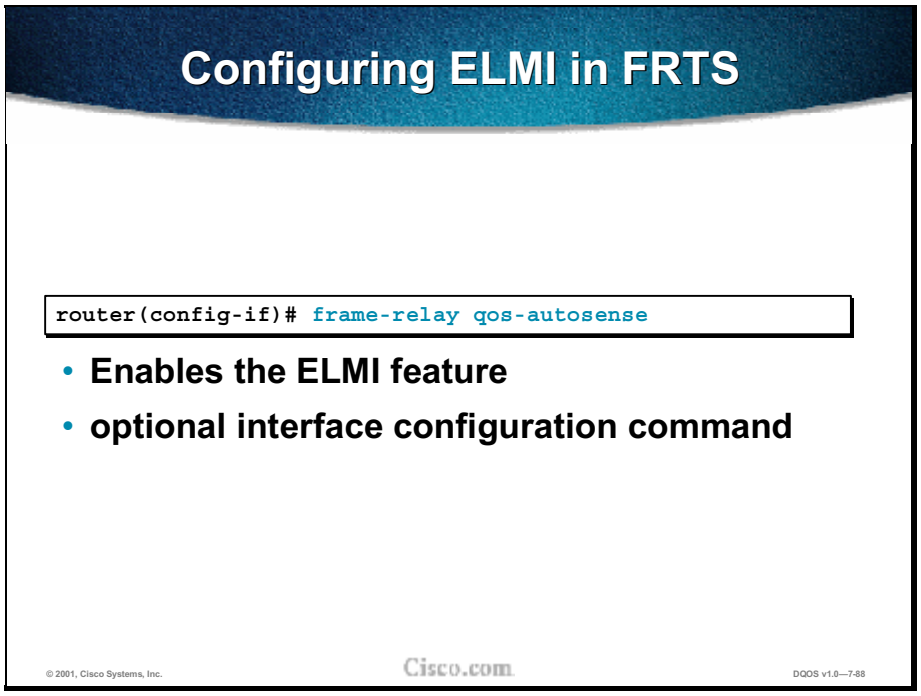
Enhanced local management interface (ELMI) simplifies traffic shaping configuration on the router. Previously, users needed to configure traffic shaping rate enforcement values, possibly for every VC. Enabling ELMI reduces the chance of specifying inconsistent or incorrect values when configuring the router.



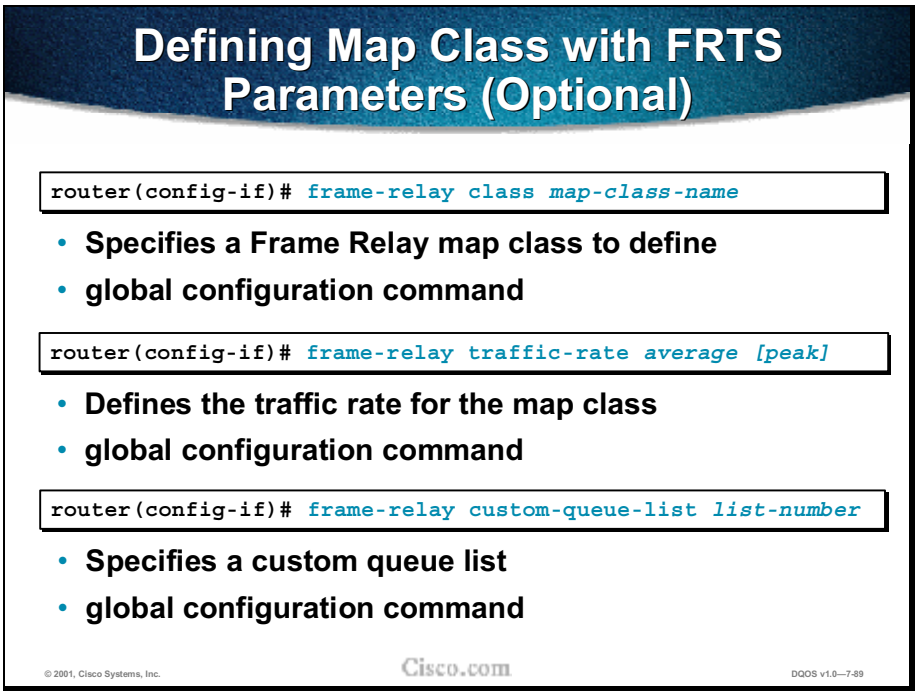
ELMI enables automated exchange of Frame Relay QoS parameter information between the Cisco router and the Cisco switch. Routers can base congestion-management and prioritization decisions on known QoS values, such as the CIR, Bc, and Be. The router senses QoS values from the switch and can be configured to use those values in traffic shaping. This enhancement works between Cisco routers and Cisco switches.



Enabling FRTS on an interface enables both traffic shaping and per-VC queuing on all the interface's PVCs and SVCs. Traffic shaping enables the router to control the circuit's output rate and react to congestion notification information if also configured.

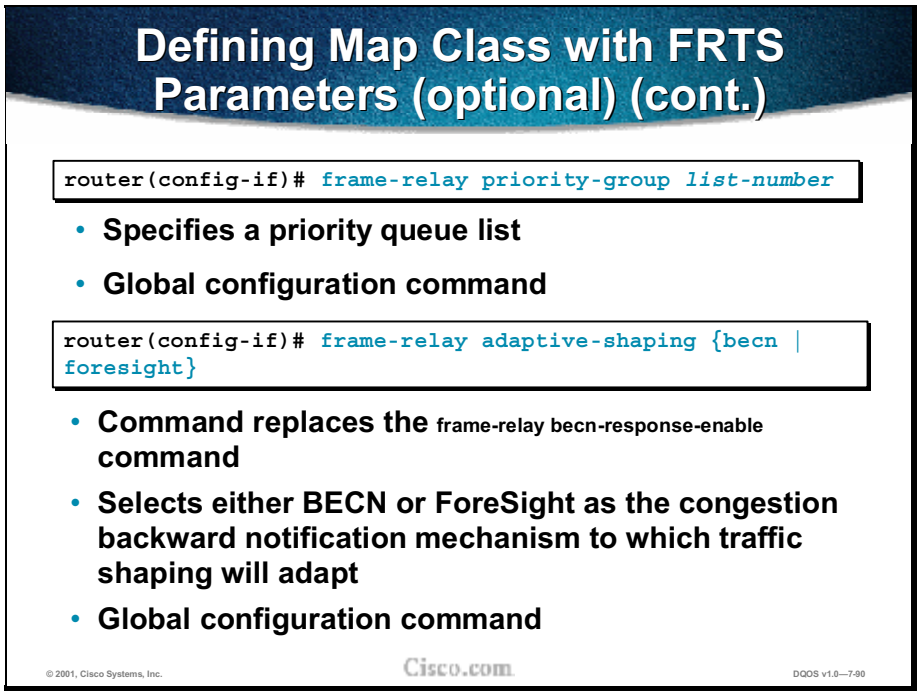


To enable ELMI, you must configure it on the main interface. It is not necessary to configure traffic shaping on the interface to enable ELMI. You might want to enable it to know the values being used by the switch.

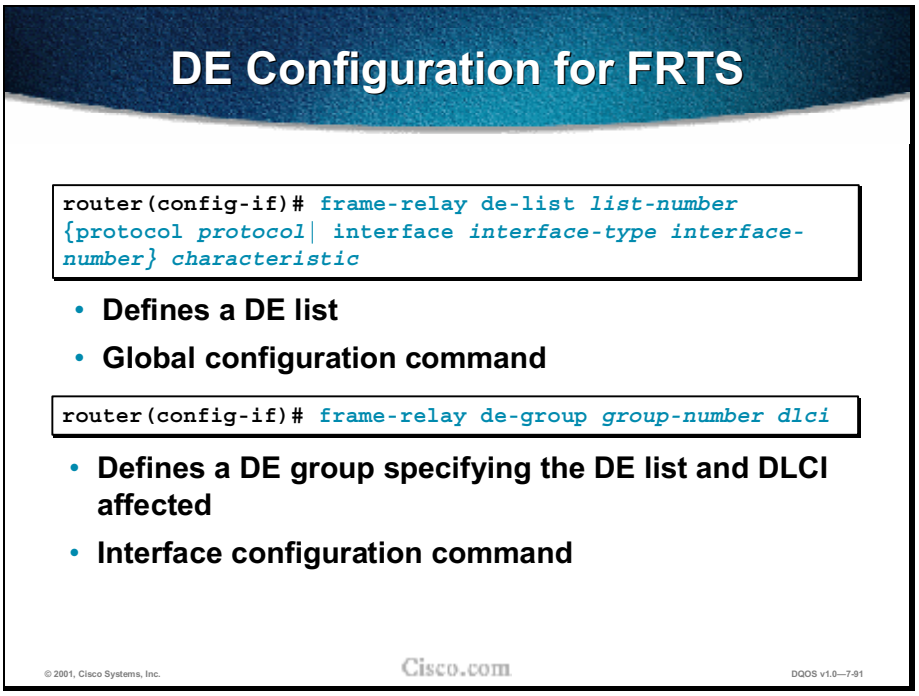


When you specify a Frame Relay map class for a main interface, all the VCs on its subinterfaces inherit all the traffic shaping parameters defined for the class. You can override the default for a specific DLCI on a specific subinterface by using the **frame-relay class** VC configuration command to assign the DLCI explicitly to a different class.

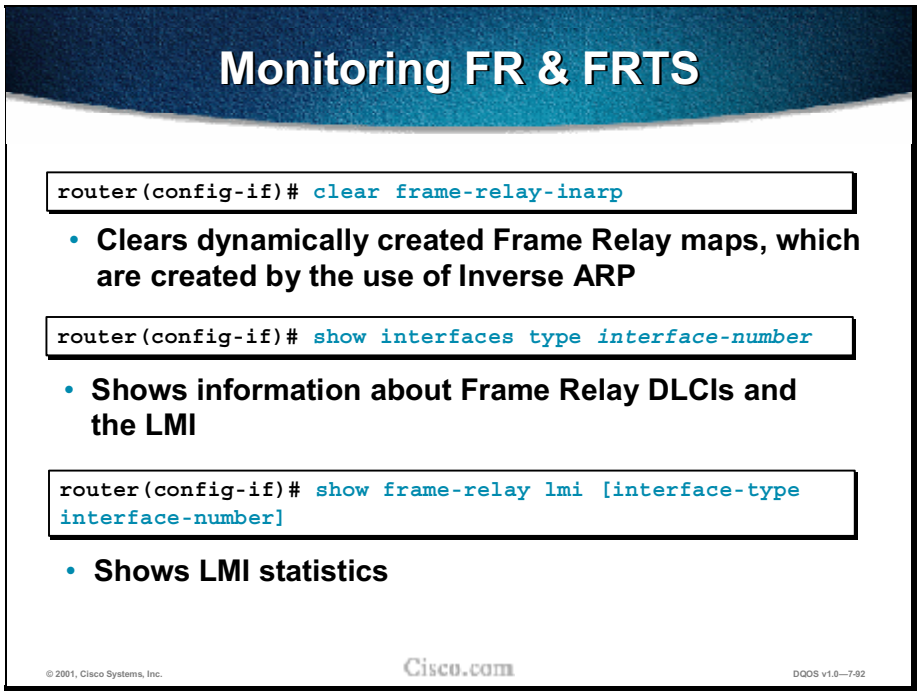
When you define a map class for Frame Relay, you can define the average and peak rates (in bits per second) allowed on VCs associated with the map class. You can also specify either a custom queue list or a priority queue group to use on VCs associated with the map class.



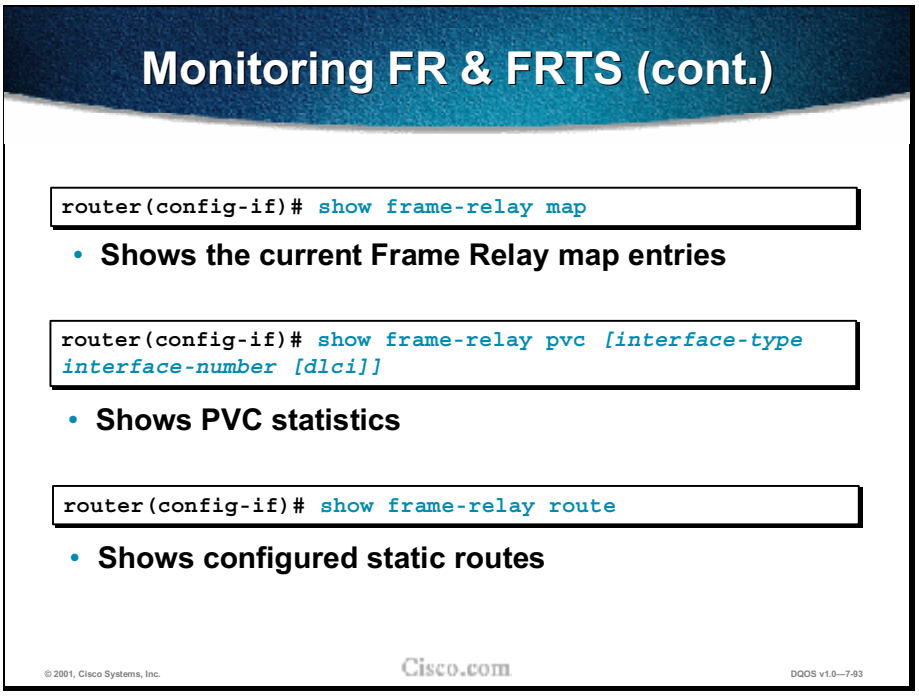
As part of the map class definition, select either BECN or ForeSight as the congestion backward-notification mechanism to which traffic shaping will adapt.



You can specify which Frame Relay packets have low priority or low time sensitivity and should be the first to be dropped when a Frame Relay switch is congested. The mechanism that allows a Frame Relay switch to identify such packets is the discard eligible (DE) bit. This feature requires that the Frame Relay network be able to interpret the DE bit. Some networks take no action when the DE bit is set. Other networks use the DE bit to determine which packets to discard. The most desirable interpretation is to use the DE bit to determine which packets should be dropped first and also which packets have lower time sensitivity. You can define DE lists that identify the characteristics of packets to be eligible for discarding, and you can also specify DE groups to identify the DLCI that is affected. You can define a DE group specifying the DE list and DLCI affected.



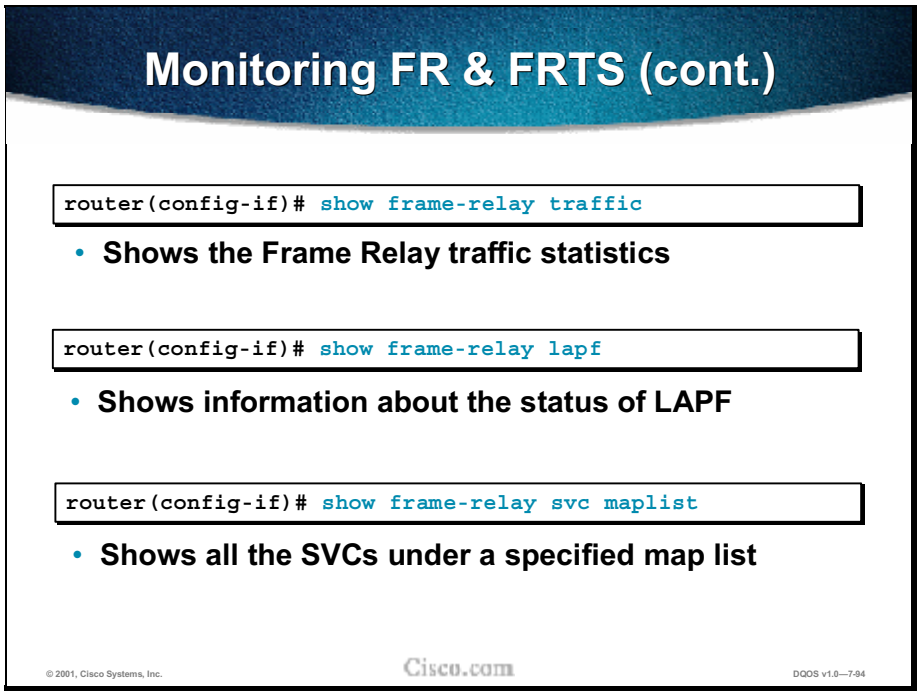
LMI stats include LMI specification (CISCO, ANSI, or ITU-T), number of received LMI messages with invalid unnumbered information field, number of received LMI messages with invalid protocol discriminator, number of received LMI messages with invalid dummy call references, number of received LMI messages with invalid message type, number of received LMI messages with invalid status message, number of received LMI messages with invalid lock shift type, number of received LMI messages with invalid information identifier, number of received LMI messages with invalid report information element (IE) length, number of received LMI messages with invalid report request, number of received LMI messages with invalid keep IE length, number of LMI status inquiry messages sent, number of LMI status messages received, number of LMI asynchronous update-status messages received, number of times the status message was not received within the keepalive time value, number of LMI status enquiry messages received, number of LMI status messages sent, number of times the status enquiry message was not received within the T392 DCE timer value, and number of LMI asynchronous update status messages sent.



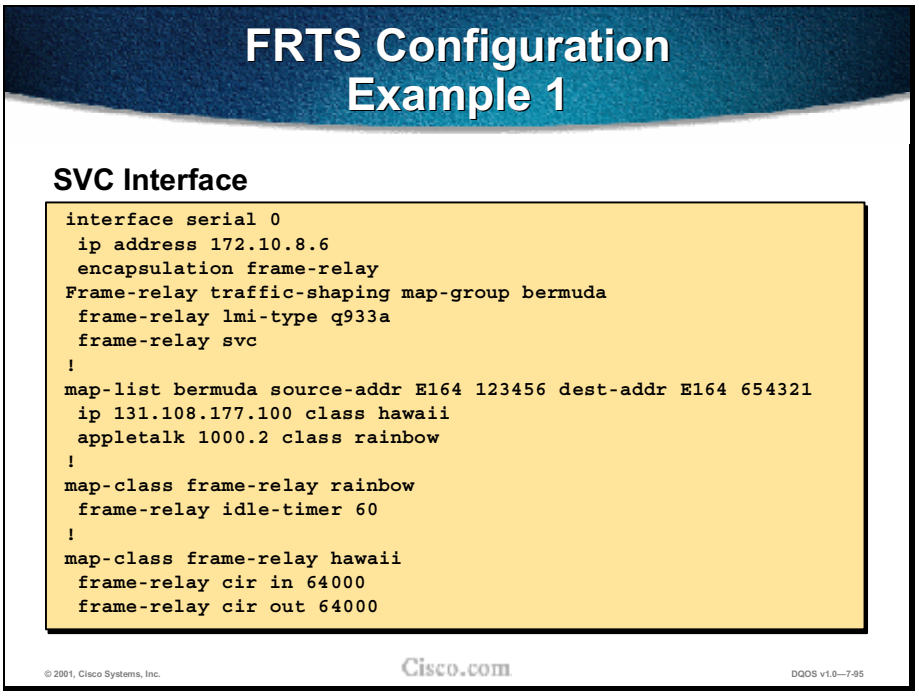
show frame-relay map: The output identifies a Frame Relay interface and its status, destination IP address, and DLCI that identifies the logical connection being used to reach this interface; indicates whether this is a static or dynamic entry; indicates the encapsulation type for this map (either CISCO or IETF); and indicates whether the TCP/IP header compression characteristics were inherited from the interface or were explicitly configured for the IP map.

show frame-relay pvc: The output identifies current committed information rate (CIR), in bits per second; current committed burst size, in bits; current excess burst size, in bits; maximum number of bytes transmitted per internal interval (excess plus sustained); interval being used internally (may be smaller than the interval derived from Bc/CIR); minimum committed information rate (CIR) for the PVC; number of bytes that will be sustained per internal interval; whether Frame Relay has been BECN Adaptation configured; and shows identifier and parameter values for a custom queue list defined for the PVC. These identifiers and values correspond to the command **queue-list 1 queue 4 byte-count 100**; and output queues used for the PVC, with the current size, the maximum size, and the number of dropped frames shown for each queue.

Use the **show frame-relay route** EXEC command to display all configured Frame Relay routes, along with their status.



To display the global Frame Relay statistics since the last reload, use the **show frame-relay traffic** EXEC command. To display information about the status of the internals of Frame Relay Layer 2 (LAPF) if SVCs are configured, use the **show frame-relay lapf** EXEC command. To display all the SVCs under a specified map list, use the **show frame-relay svc maplist** EXEC command.



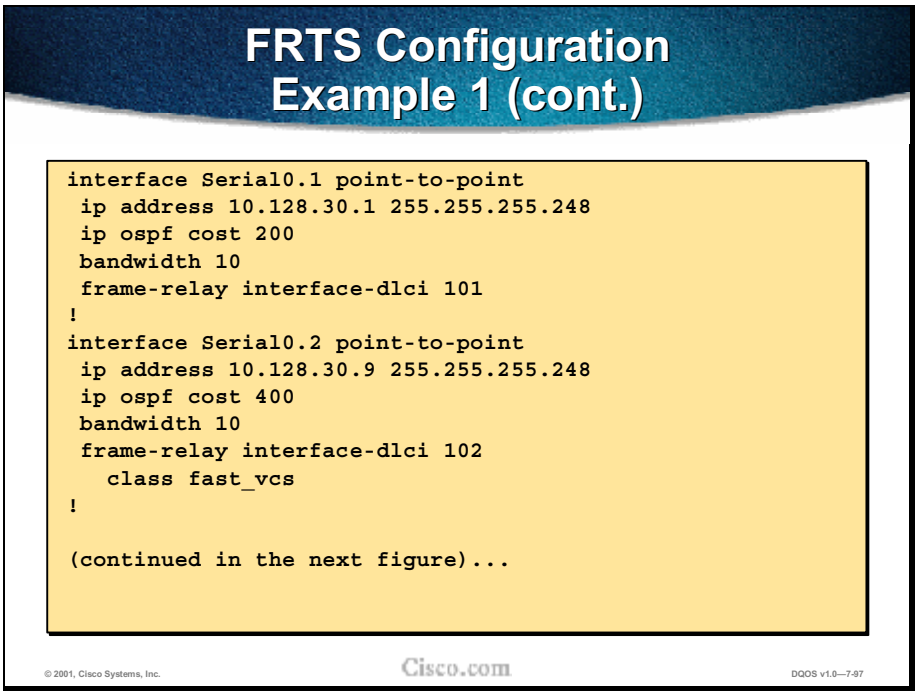
SVC operation can be enabled at the interface level only. Once it is enabled at the interface level, it is enabled on all subinterfaces on the interface. One signaling channel, DLCI 0, is set up for the interface, and all SVCs are controlled from the physical interface.

The first use of this command on the router starts all SVC-related processes on the router. If they are already up and running because SVCs are enabled on another interface, no additional action is taken. These processes are not removed once they are created.

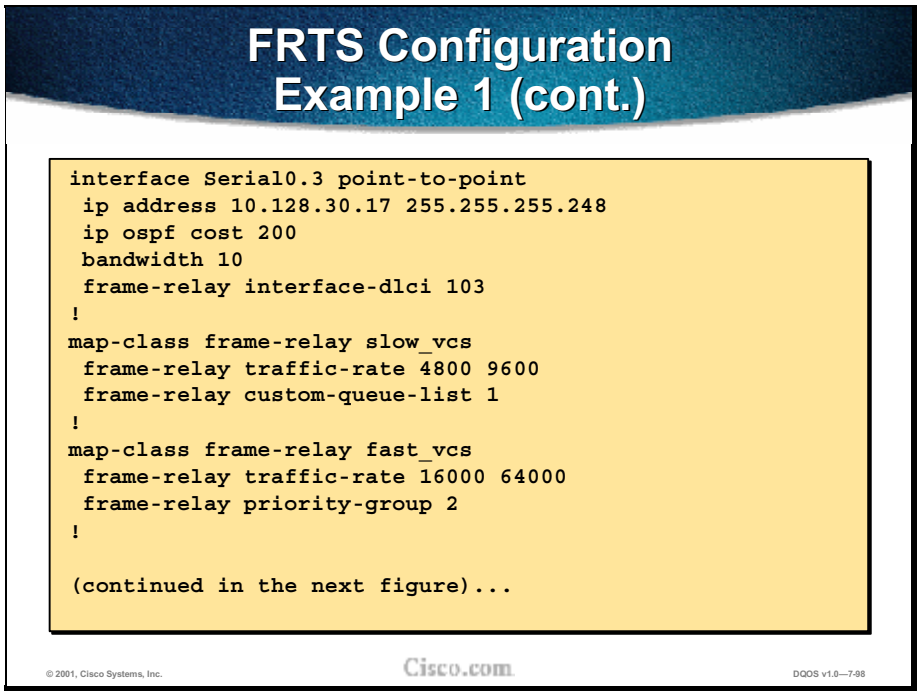
FRTS Configuration Example 1 (cont.)

```
interface Serial0
 no ip address
 encapsulation frame-relay
 frame-relay lmi-type ansi
 frame-relay traffic-shaping
 frame-relay class slow_vcs

(continued in the next figure)...
```

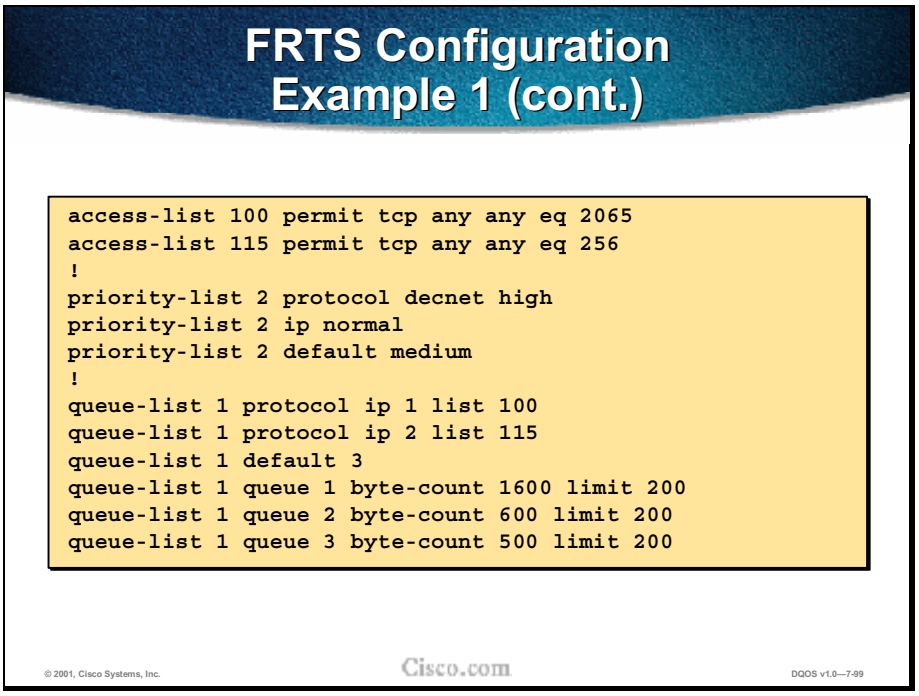

Map class fast_vcs uses a peak rate of 64,000 and average rate of 16,000 bps, configured for priority queuing using priority-group 2.



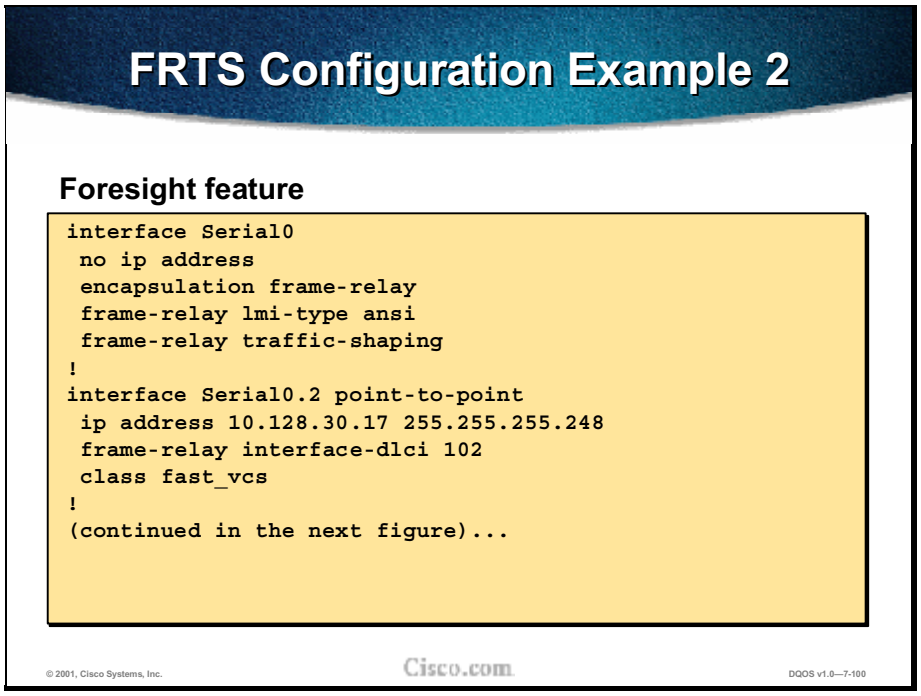
In this example, the VCs on subinterfaces Serial0.1 and Serial0.2 inherit class parameters from the main interface (defined in slow_vcs); VC defined on Serial0.2/DLCI 102 is configured to use map class fast_vcs.



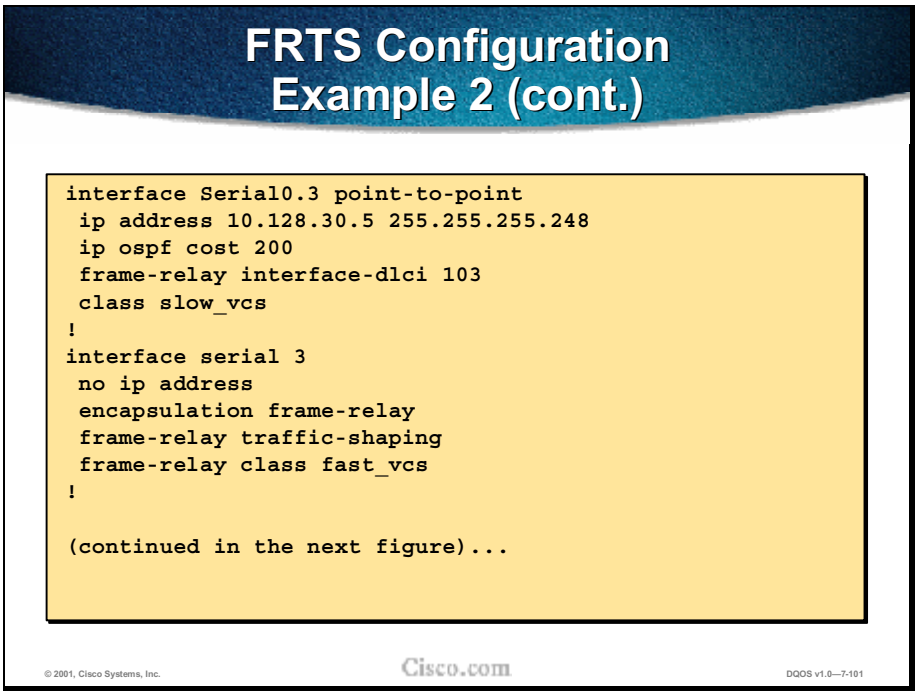
Map class slow_vcs uses a peak rate of 9600 and an average rate of 4800 bps; because BECN feedback is enabled by default, the output rate will be cut back as low as 4800 bps in response to received BECNs; slow_vcs is configured to use CQ defined as queue-list 1.



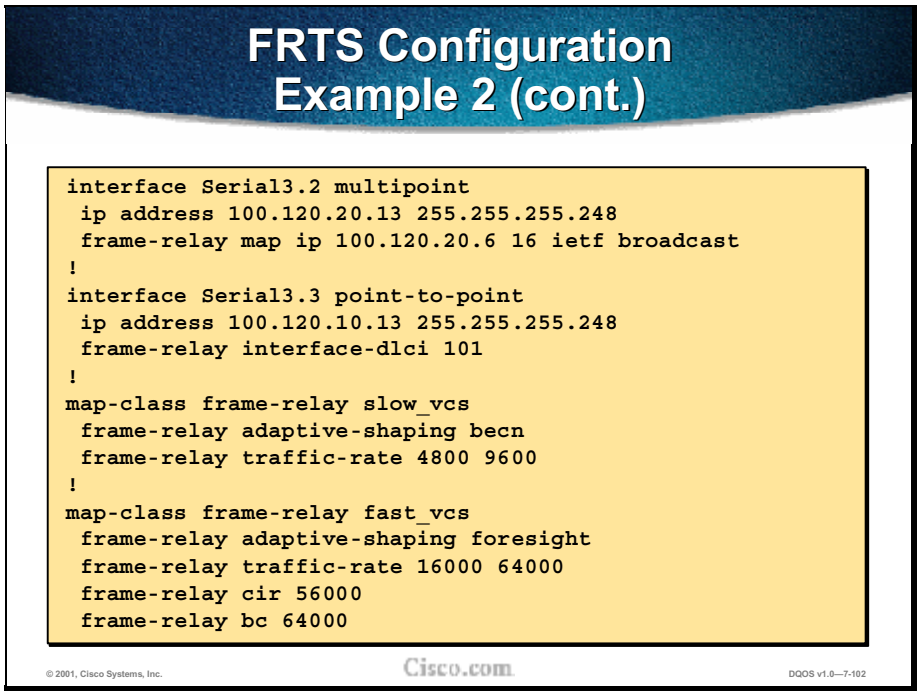
Queue-list 1 has three queues—with the first two being controlled by access lists 100 and 115.



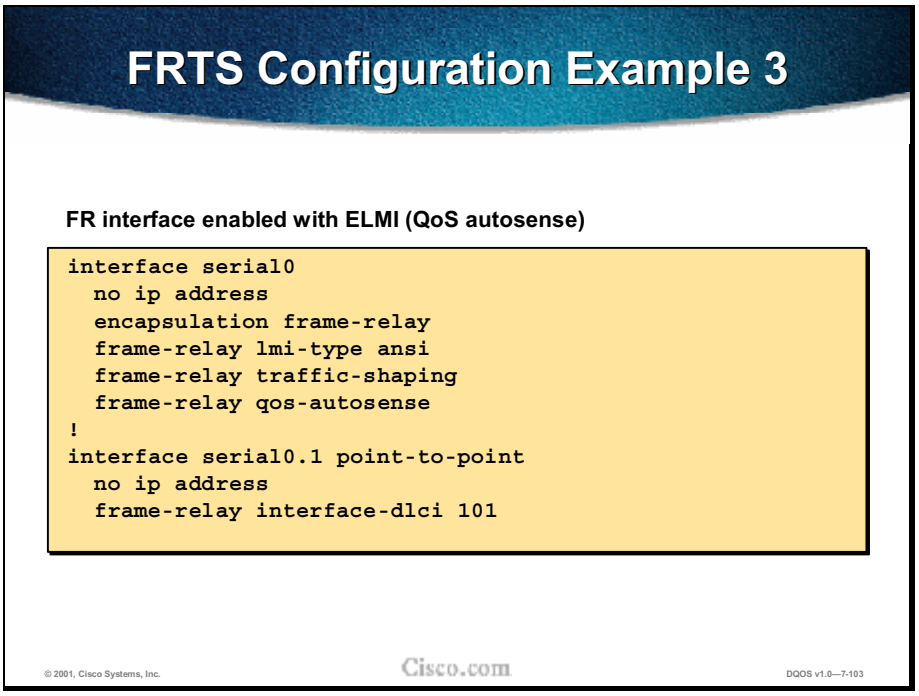
DLCIs 100 and 101 on subinterfaces Serial3.2 and Serial3.3 inherit class parameters from the main interface; traffic shaping for these two VCs will be adaptive to the ForeSight notification. For serial interface 0, the output rate for DLCI 103 will not be affected by the ForeSight function.



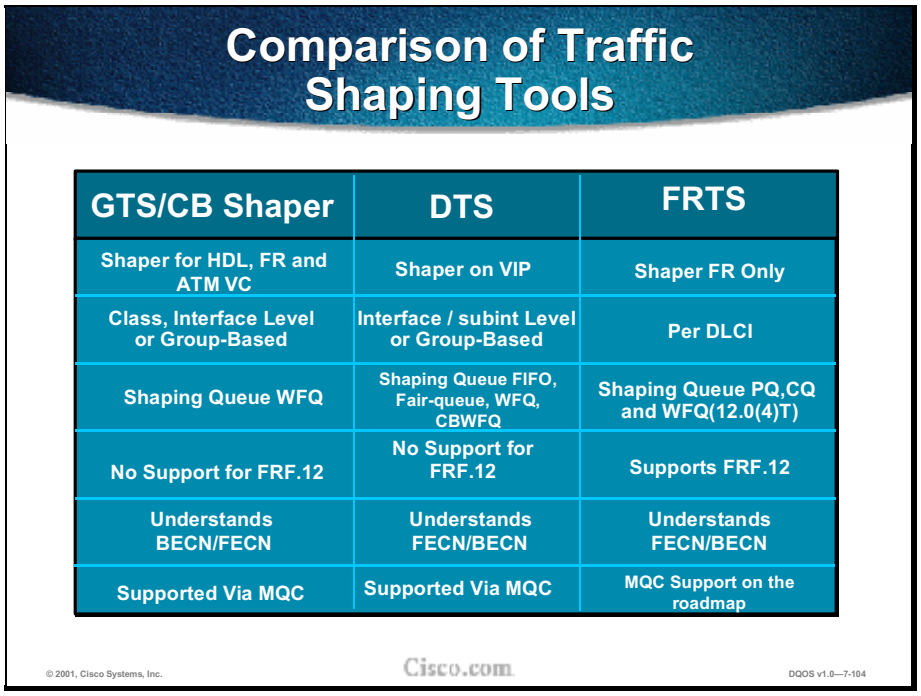
This figure shows the additional configuration of the example being discussed.



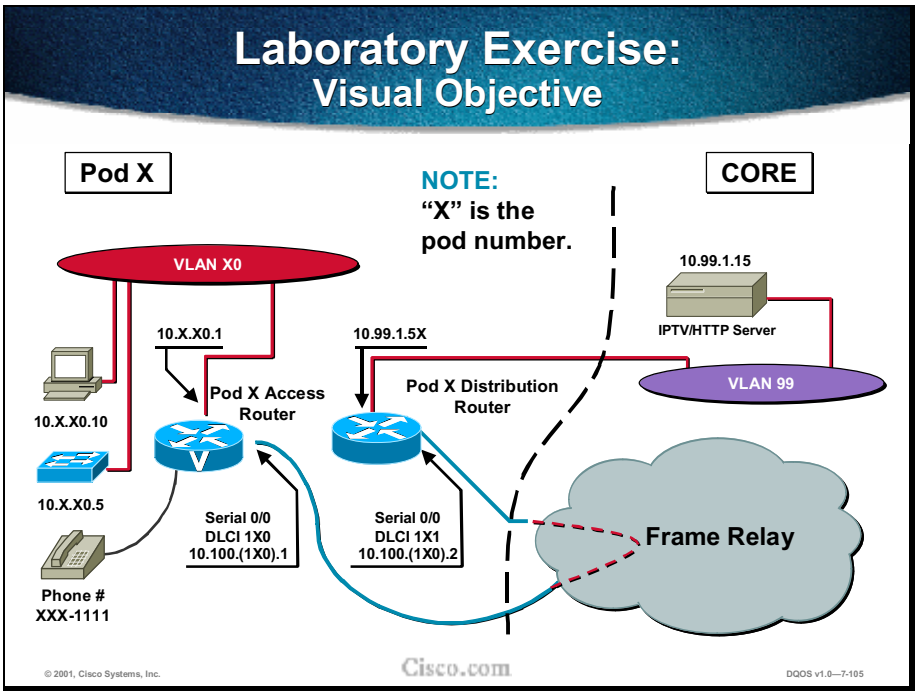
This figure shows the additional configuration of the example being discussed.



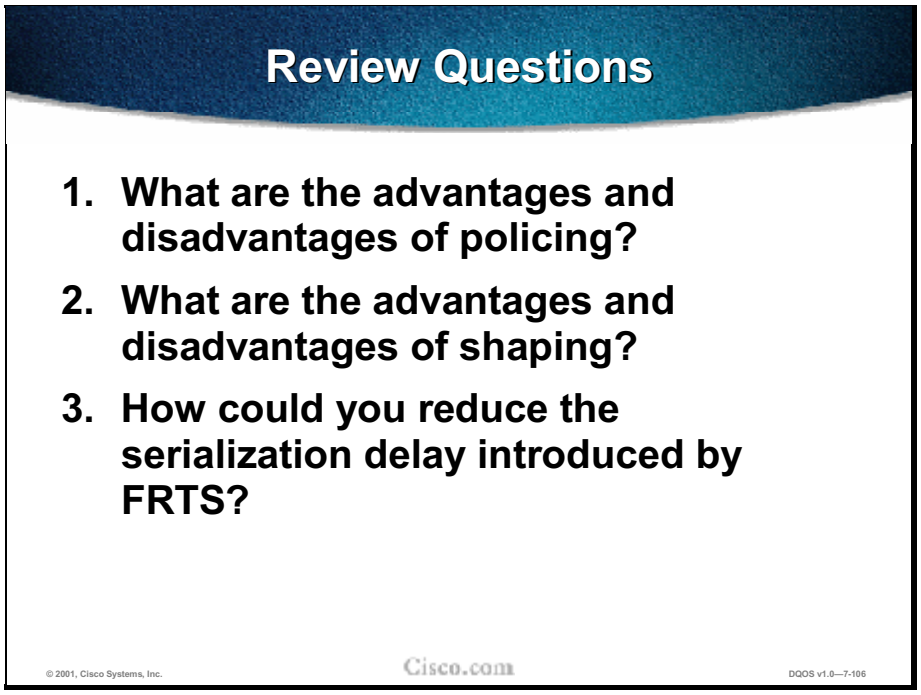
The router receives messages from the Cisco switch, which is also configured with QoS Autosense enabled.



The figure above reviews the differences and similarities between the traffic shaping tools.



Review Questions

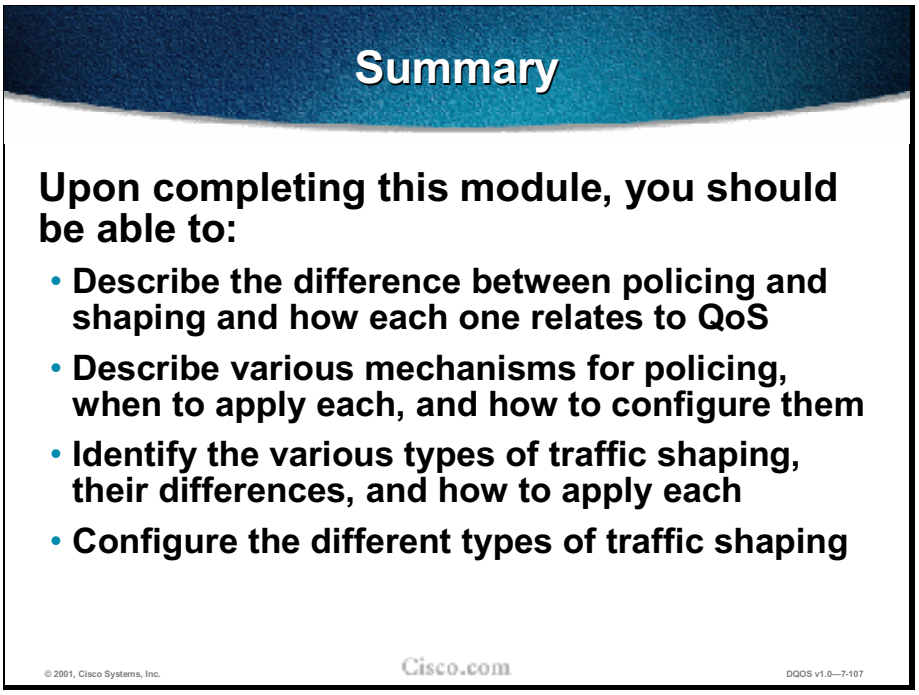


Q1) What are the advantages and disadvantages of policing?

Q2) What are the advantages and disadvantages of shaping?

Q3) How could you reduce the serialization delay introduced by FRTS?

Summary



8

Call Admission Control

Overview

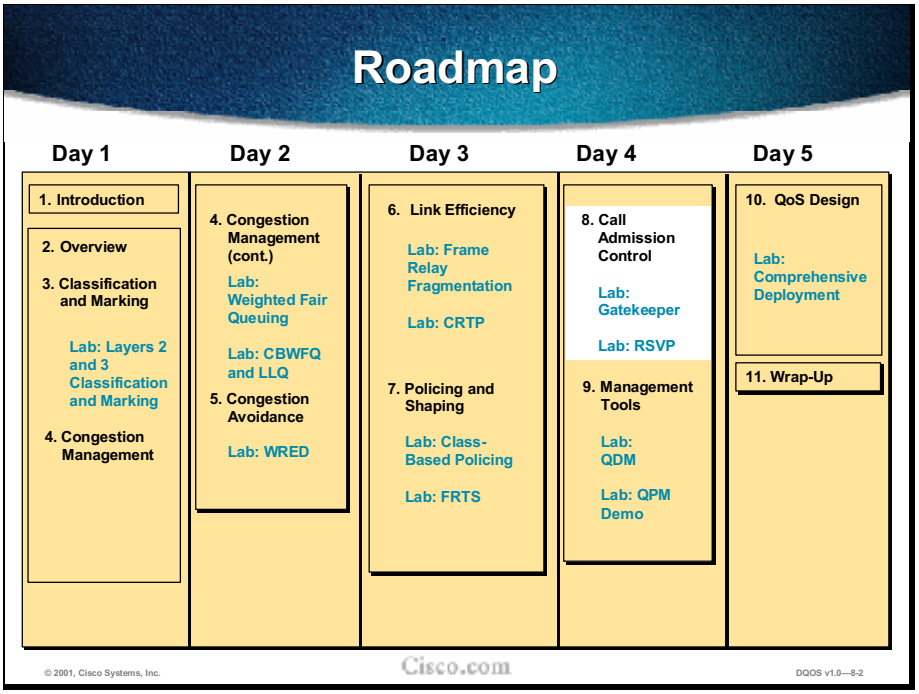
This chapter explains the attributes of call admission control (CAC) and the different categories of CAC: local, measurement-based, and resource-based. The chapter ends with a recommendation of when to use which CAC methods.

Objectives

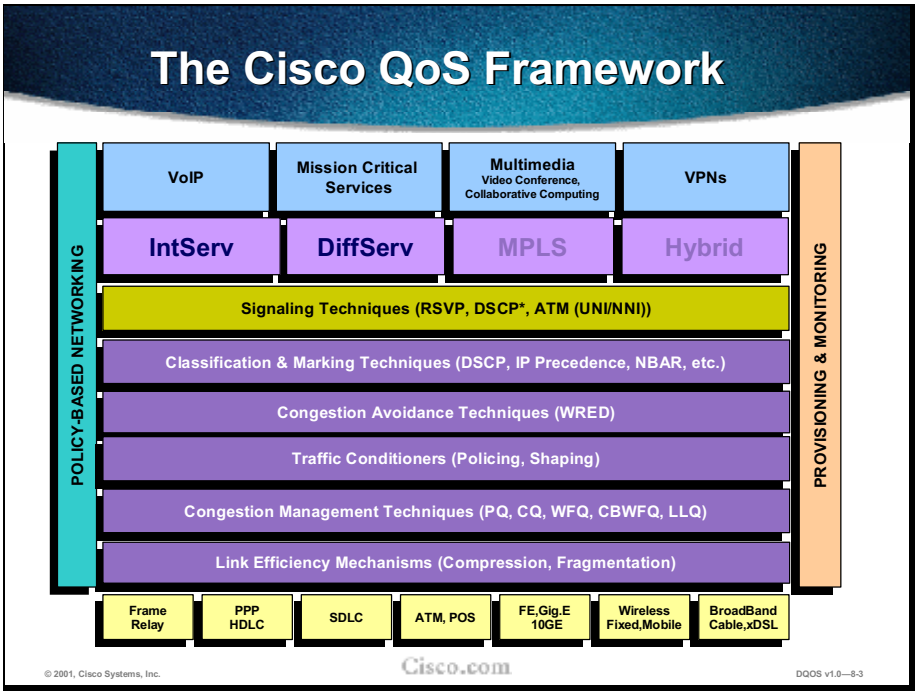
Upon completing this chapter, you will be able to:

- Correctly list five local CAC methods and their primary functions
- Correctly list two measurement-based CAC methods and their primary functions
- Correctly describe IntServ/RSVP and its main function
- Given an enterprise network scenario, correctly determine which method(s) of achieving call admission control best meets the customer requirements

Outline

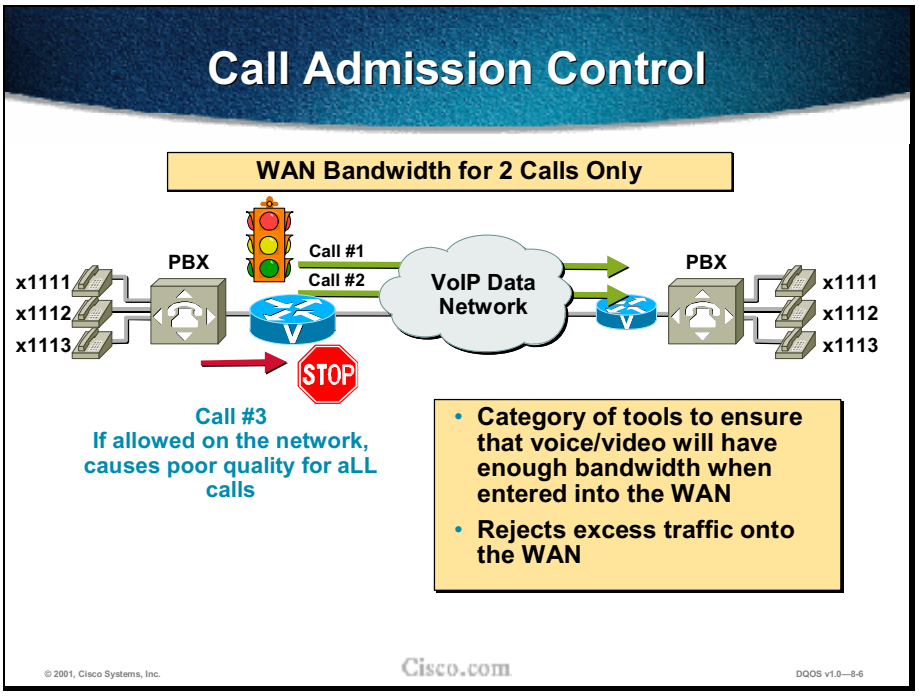


The figure above shows the plan for the week. This chapter covers CAC and has two labs: one for GateKeeper and one for RSVP.



There is a debate about whether CAC belongs in the category of quality of service mechanisms. It is not represented in the quality of service (QoS) framework above.


Agenda
• Overview
• Local Call Admission Control
• Measurement-Based CAC
• Resource-Based CAC
• Combinations, Interactions, and Sequencing
• When to Use What
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—8-5

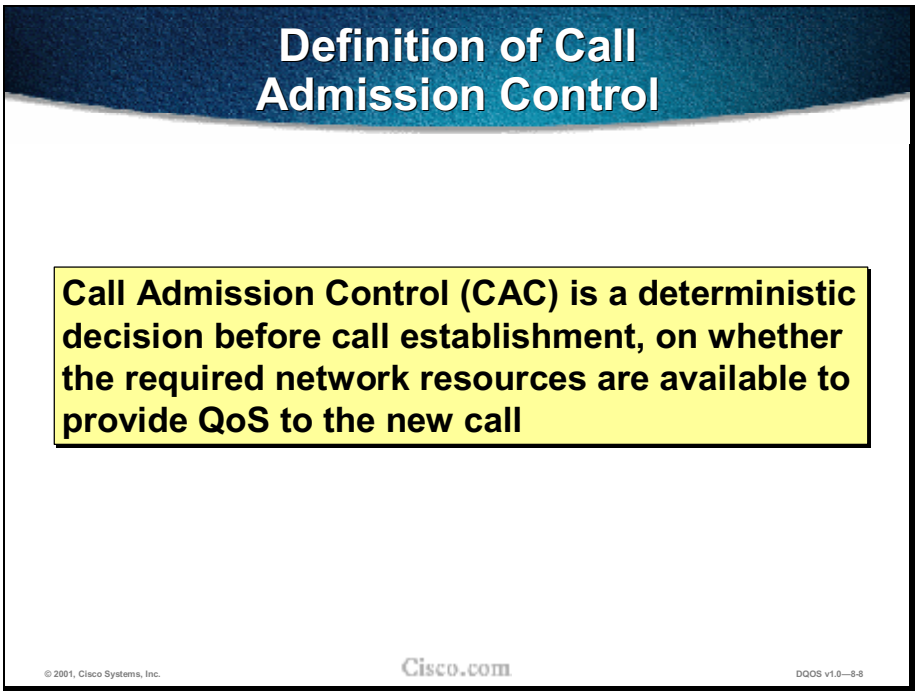


CAC is a category of tools for ensuring that there is sufficient bandwidth for voice and video before traffic enters the network. This effectively prevents too much voice or video (which typically have fixed bandwidth requirements) from being sent down a wide-area link that does not have enough bandwidth to handle it. In the case of voice, for example, if there is enough bandwidth to carry two voice calls between locations A and B, and a third call is placed, this call should not be admitted across the link because the quality of all calls could suffer, regardless of the QoS techniques used. If this call is not admitted over the IP WAN, it may be automatically rerouted via some other path, such as the PSTN links that are available between sites.

The tools that will be taught include local configuration options, measurement-based options, and resource-based options. The local configuration options are physical DS0 limitations, max connections, voice bandwidth for FR, trunk conditioning, and local voice busyout (LVBO). The measurement-based options are advanced voice busyout (AVBO), and PSTN fallback. Resource-based options are resource availability indicator (RAI) , gatekeeper (GK) zone bandwidth, and Resource Reservation Protocol (RSVP).

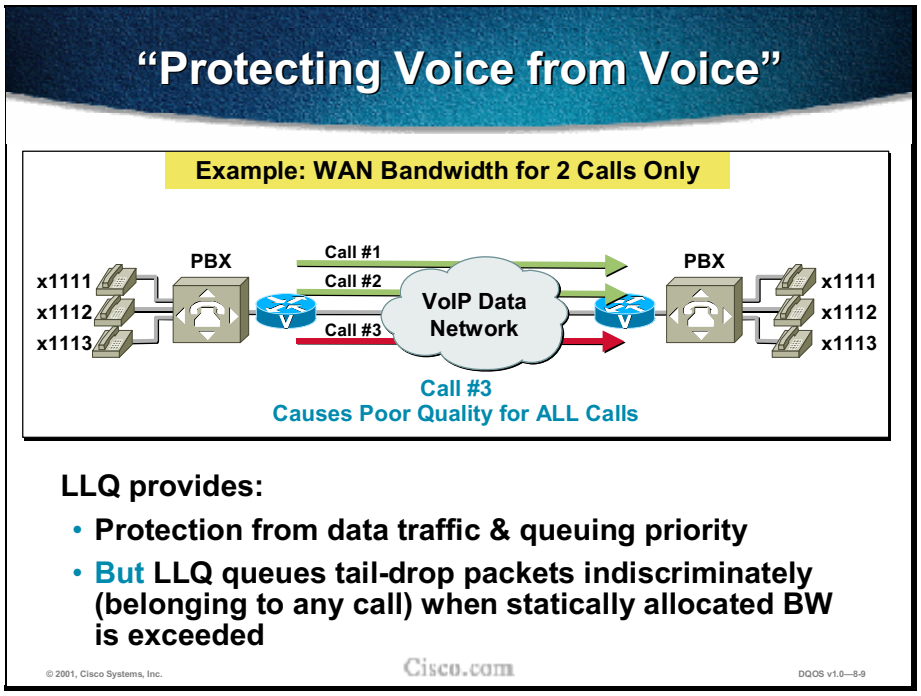
While call admission control is not represented in the QoS framework, it is a necessary component in a voice or video-over-data network in order to ensure quality.


Overview
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—8-7



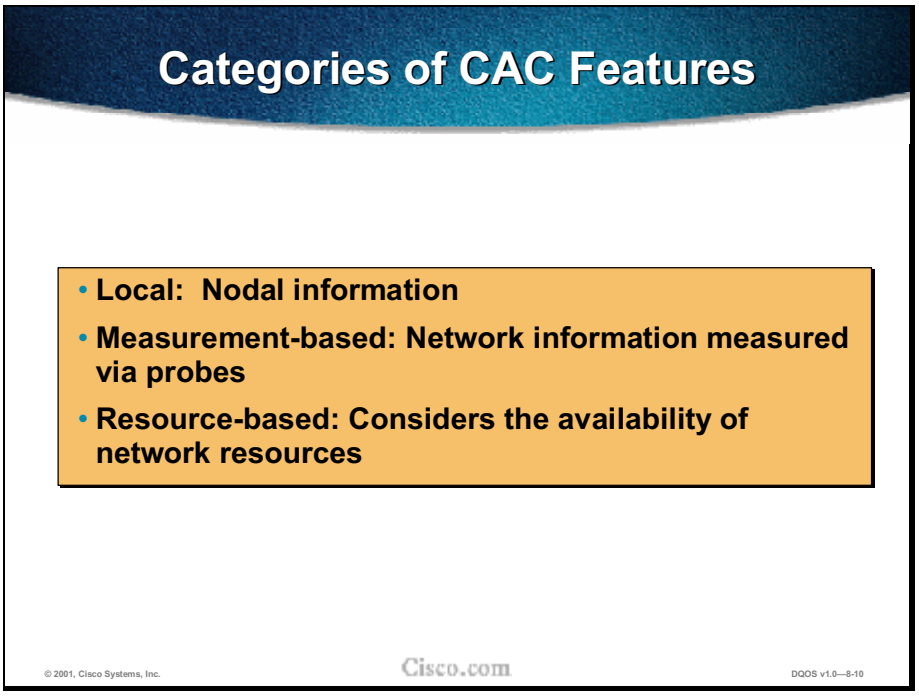
Protecting real-time traffic such as voice or video from oversubscribing the network bandwidth is accomplished by mechanisms collectively described as call admission control. The need for call admission control in IP telephony networks is amplified greatly by the fact that all IP phones have an open IP path to the WAN, whereas toll-bypass networks, in contrast, can limit the number of physical trunks eligible to initiate calls across the WAN. For real-time-sensitive traffic like voice, it is better to deny network access (under congestion) than to allow traffic and drop/delay it.

CAC features allow voice gateways and CallManager to make an informed decision before admitting a new call, based on the condition of the network. Other options include reordering the tone, trying another VoX route, or redirecting the call through the PSTN.



CAC is required to ensure that network resources are not oversubscribed. When more calls attempt to travel over the data network than the network can handle, all calls suffer poor quality.

Low latency queuing (LLQ) is not sufficient as a CAC mechanism. Though LLQ does provide protection from data traffic and queuing priority, it queues tail-drop packets indiscriminately (belonging to any call) when bandwidth statically allocated is exceeded.

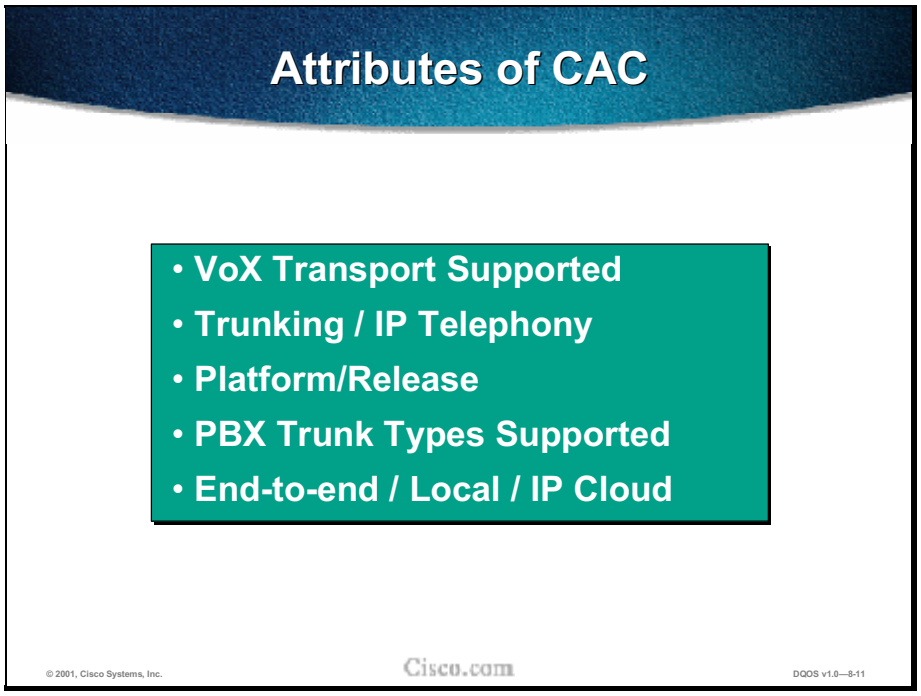


CAC is accomplished through either a decision that is made at the local level, through measurement of the network, or through resource reservation in the network.

Local CAC mechanisms function on the originating gateway (OGW). The CAC decision is based on nodal information such as the state of the outgoing LAN/WAN link. Clearly if the local packet network link is down, there is no point in executing complex decision logic based on the state of the rest of the network, since that network is unreachable. Other local mechanisms include configuration items to disallow more than a fixed number of calls.

The **measurement-based CAC** techniques look ahead into the packet network to gauge the state of the network in order to determine whether to allow a new call. This implies sending probes to the destination IP address (likely the terminating GW, or TGW) that will return to the OGW with some measured information on the conditions it found while traversing the network to the destination. Typically, loss and delay characteristics are the interesting elements of information for voice.

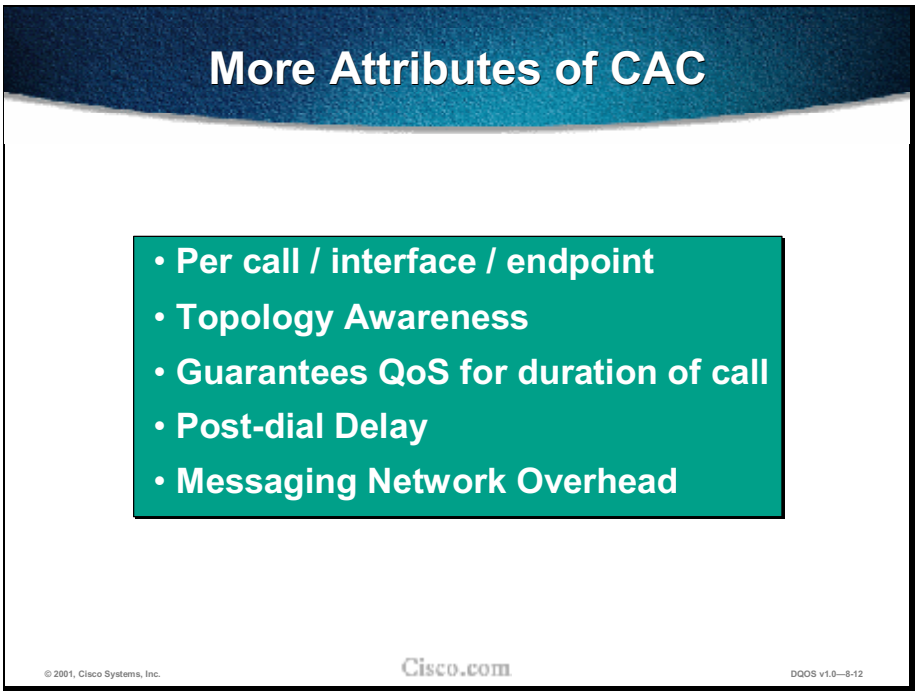
There are two types of **resource-based mechanisms:** those that calculate resources needed and/or available and those reserving resources for the call. Resources of interest include link bandwidth, DSPs, and DS0 timeslots on the connecting TDM trunks, CPU power, and memory. Several of these resources could be constrained at any of the nodes, or multiple nodes, the call will traverse to its destination.



As each CAC method is described in this chapter, a comparison of the features based on various factors and criteria will help determine which is the best or the most appropriate method for the network design under consideration.

As an introduction, there are ten criteria of interest when weighing different CAC tools:

- **VoX transport supported:** Which voice technologies are supported by the method? Some methods apply to a single technology, while other methods apply across the board.
- **Trunking/IP telephony:** Is the method usable only between voice GWs connected to the PSTN or a PBX, or can this method also be used with IP phone endpoints?
- **Platform/release:** Which IOS platforms are this feature available on, and in which software release was it introduced?
- **PSTN/PBX trunk types supported:** Some CAC features have a dependency on the PSTN/PBX trunk type used in the connection or act differently with CCS trunks versus CAS trunks.
- **End-to-end/local/IP cloud:** This is the scope of visibility of the CAC feature. Some mechanisms work locally on the OGW only, others consider the cloud between the source and destination nodes, some consider the destination POTS interface, and some work end-to-end.



- **Per call/interface/endpoint:** Different mechanisms involve different elements of the network. Several CAC methods work per call, but some work per interface, and some work per endpoint or IP destination.

- **Topology awareness:** Does the CAC mechanism take into account the topology of the network, and therefore provide protection for the links and nodes in the topology?

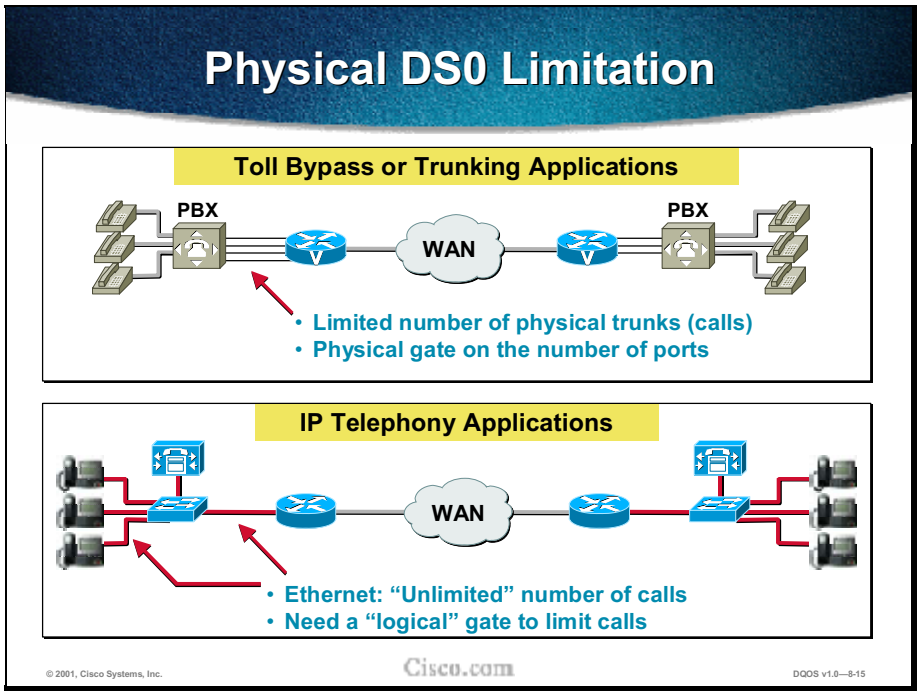
- **Guarantees QoS for duration of call:** Does the mechanism make a one-time decision before allowing the call, or does it also protect the QoS of the call for the duration of the call by reserving the required resources?

- **Postdial delay:** Does the mechanism impose an additional postdial delay because it requires extra messaging or processing during call setup?

- **Messaging network overhead:** Does the method use additional messaging that has to be provisioned in the network to gather the information necessary for the CAC decision?


Local Call
Admission Control
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—8-13

In this section, you will learn about five different methods for applying CAC on the OGW. These are physical DSO limitation, max connections, voice bandwidth for Frame Relay, trunk conditioning, and local voice busyout.



The simplest of all CAC features is not really a software feature at all but a design methodology based on the physical limitations of the interfaces. Although ridiculously simple when compared to some of the other features, it is nevertheless a key building block to many existing customer networks.

For example, if the desire is to limit to five the number of calls from the originating PBX to the OGW, only configure or enable five timeslots on the T1/E1 trunk between the switch and the OGW. This works well for many toll-bypass applications. It is the predominant CAC mechanism deployed in toll-bypass networks today. It protects the bandwidth on the WAN link of the local site.

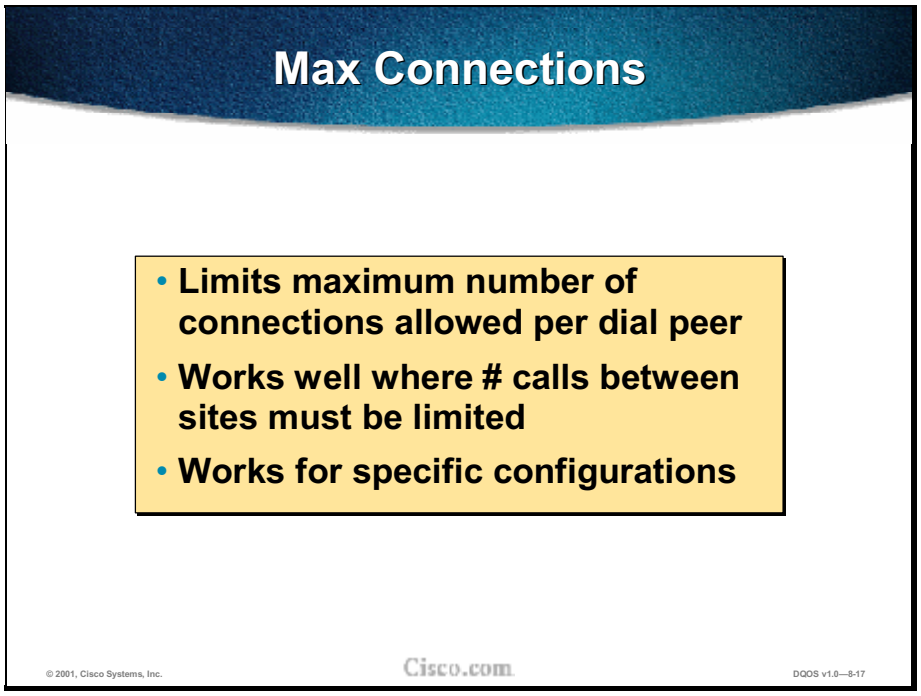
This design method, because it is local, provides adequate protection for the egress WAN link from the OGW. Its shortcoming, as that of all local mechanisms, is that it provides no protection against the availability of bandwidth on any other link in the network. It works well in simple hub-and-spoke topologies and reasonably well in more complex multilayer hierarchical networks. That is because the maximum number of possible calls (worst case) on any backbone link can be reasonably and accurately estimated by a calculation based on the known number of calls that can come in from each edge location and the busy-hour traffic patterns of who calls whom between locations.

However, this method is limited to simple topologies, does not work for IP telephony applications, and does not react to link failures or changing network conditions.

Incoming DS0 Limitations Attributes

VoX Supported:	All
Trunking/IP Telephony	Trunking only
Platform/Release	All Voice GWs; all releases of SW
PBX Trunk Types Supported	All
Per call/interface/endpoint	Per call

Note Other attributes are not present.



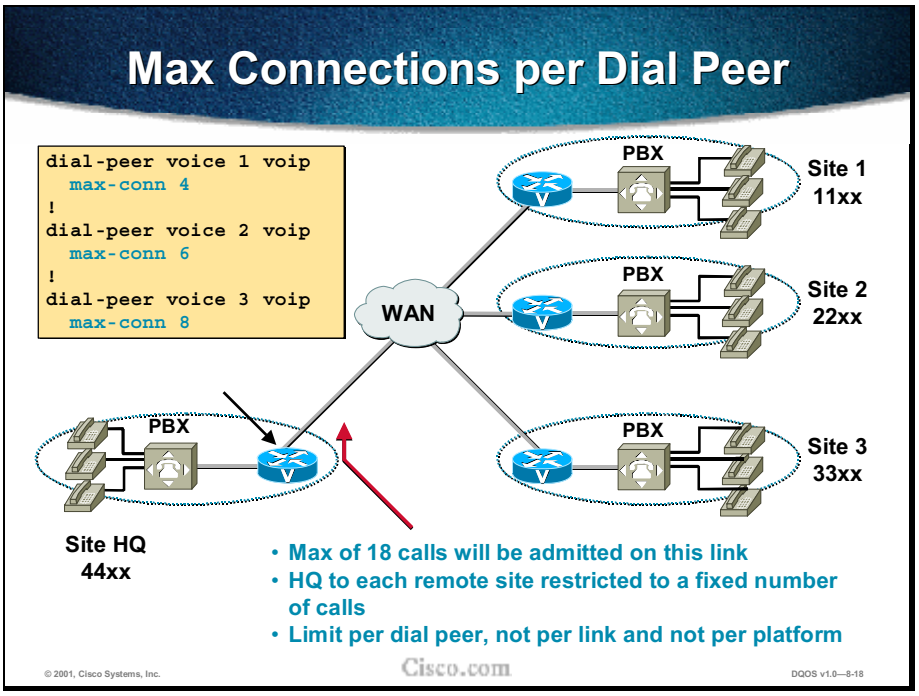
A simple and easy-to-use feature, the max connections CLI can be entered on the VoX dial peer of the OGW. It restricts the number of concurrent connections (calls) that can be active by that dial peer at one time.

This feature provides a viable CAC method in at least two scenarios:

- When the sum of the individual max-connections dial-peer statements will provide the maximum number of calls that can be simultaneously active across the WAN link
- If the design objective is to limit the maximum number of calls between sites (rather than protecting the bandwidth of egress WAN link)

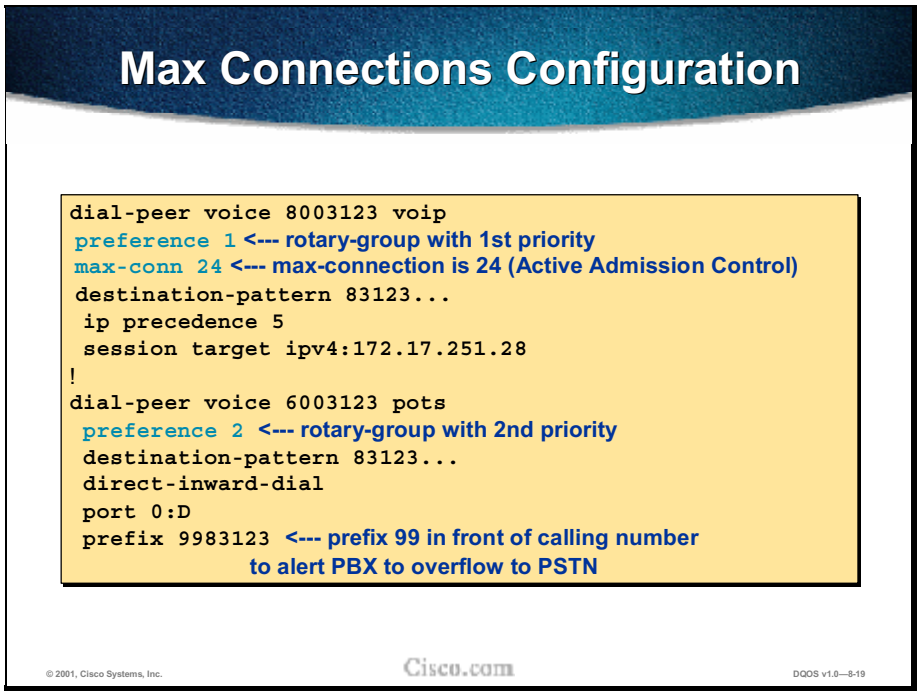
Although useful in many scenarios, the drawbacks of this feature include:

- When it is applied per dial peer, there is no way to limit the maximum number of calls the OGW can have active simultaneously. It is limited in the scope of the network design problems it can solve.
- It provides some protection for Voice GW egress WAN link, but little to no protection for links in the network backbone.
- It does not work for IP telephony applications that do not use dial peers.
- It is limited to simple topologies.
- It does not react to link failures or changing network conditions.

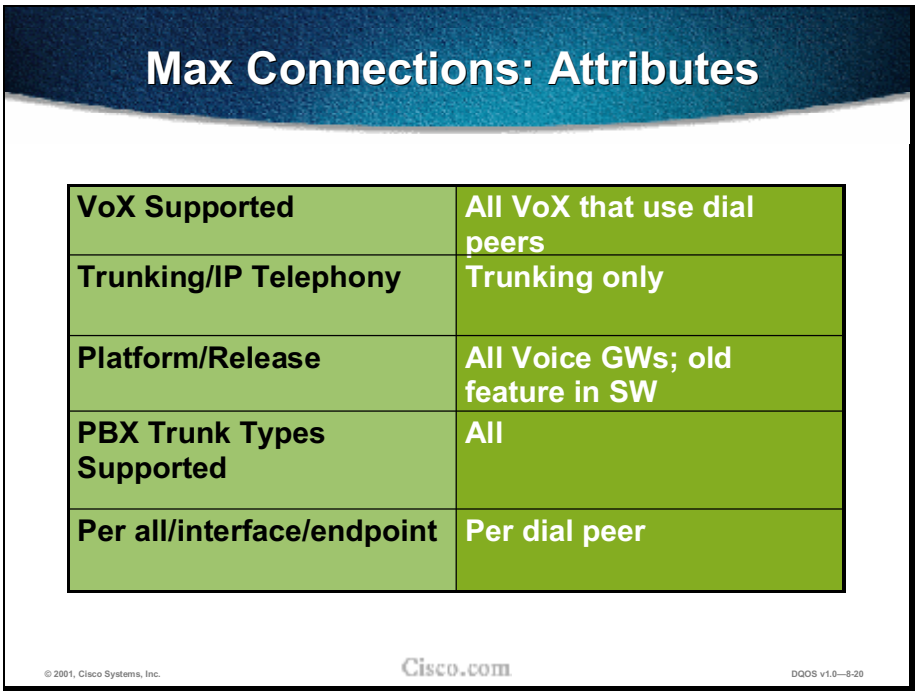


The max connections CLI can be entered on the VoX dial peer of the OGW. It restricts the number of concurrent connections (calls) that can be active by that dial peer at any one time.

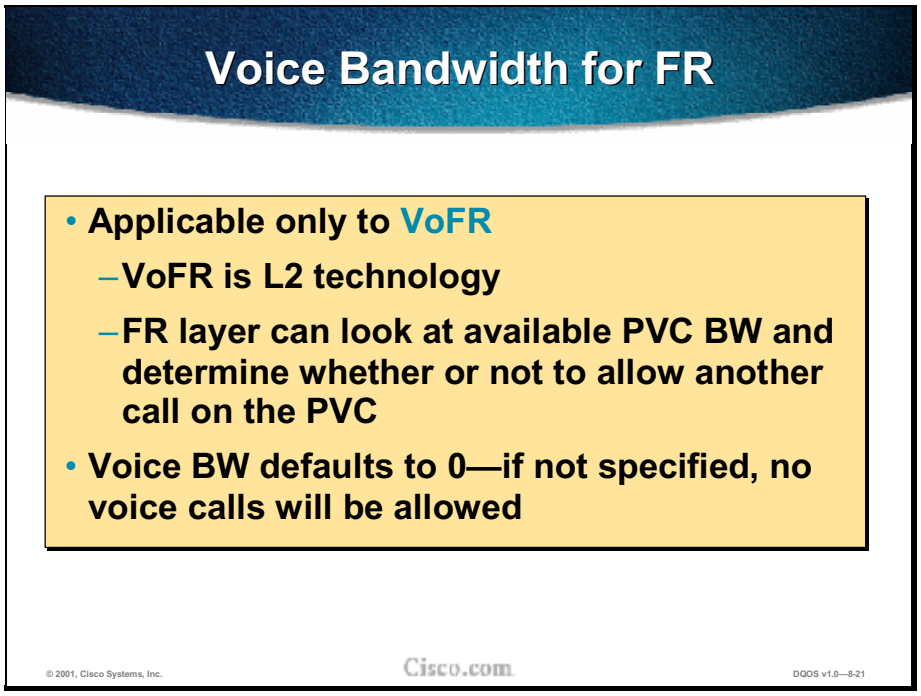
The slide shows an example of this type of network: There are three remote sites, each with recognizable first digits in the dialing plan. The outgoing VoX dial peers at the headquarters (HQ) site therefore matches the remote sites one for one. The numbers of calls to remotes sites 1, 2, and 3 will be limited to 4, 6, and 8 respectively. The egress WAN link can therefore have no more than 18 calls active at any one time, and in this configuration it is prudent to provision the bandwidth of this link for that number of calls.



The CLI segment above shows an example where no more than 24 VoIP calls will be allowed (dial peer 800) over the egress WAN link. The 25th call will be hairpinned back to the PBX to be redirected to the PSTN (suitable digits are prepended to the dial string to direct the routing logic of the PBX).



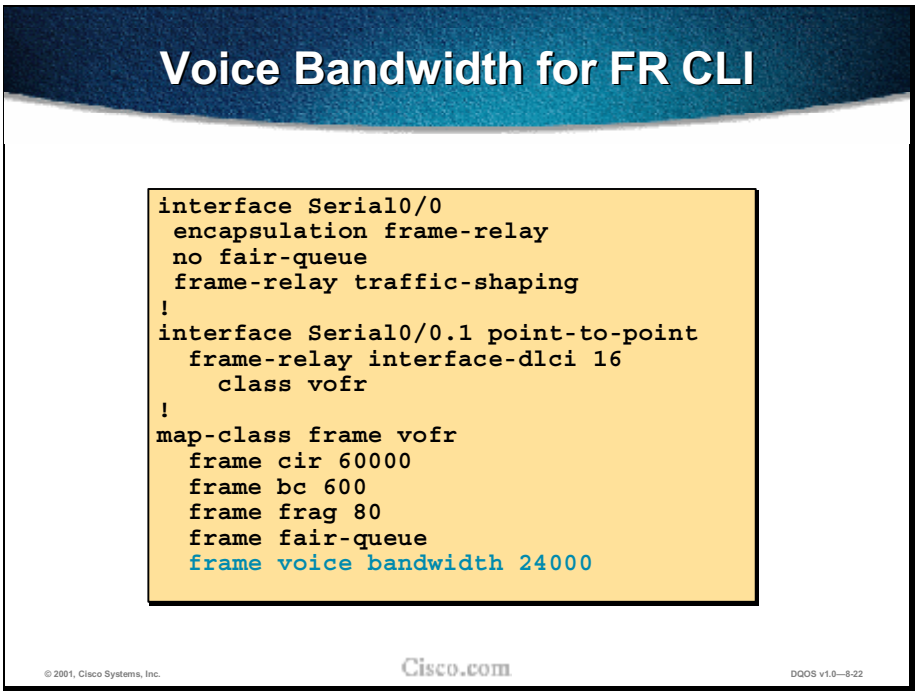
Note	Other attributes are not present.



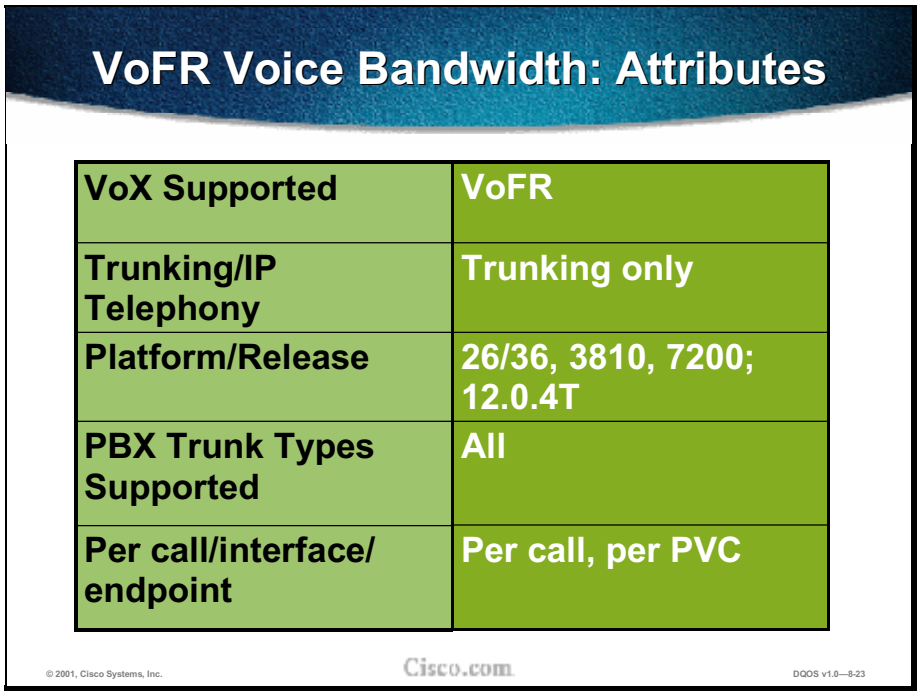
In VoFR configurations, a **voice-bandwidth** command is used in the FR map class. This command sets aside bandwidth for VoFR calls similar to the way in which the IP RTP Priority and LLQ features do for general traffic flows. However, it differs from these other features in one important respect and that is the voice bandwidth for VoFR not only manages bandwidth, but also provides CAC, which the general queuing features do not.

Since VoFR is a Layer 2 technology, voice bandwidth is able to provide CAC. The FR software implicitly knows what frames are voice frames and which are data—from the FRF.11 (voice) and FRF.3.1 (data) headers. Subsequent fields in the header carry channel identification (CID) and payload information, identifying for FR which frame belongs to which voice call. When this command sets aside bandwidth for voice, it can also deny the next call that exceeds the bandwidth allocated to voice.

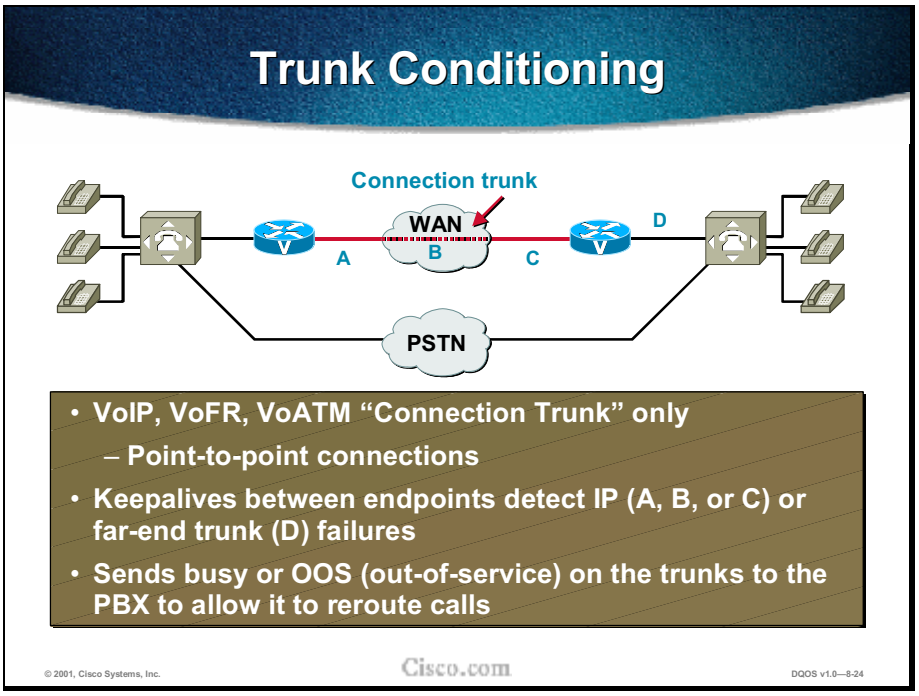
This feature may be of limited use as it applies only to VoFR networks, but it is nevertheless handy for customers for whom VoFR is a viable technology. It should also be noted that the voice-bandwidth defaults to 0, so if not specified at all, no voice calls will be allowed over the WAN link. Signaling traffic should not be included in the bandwidth specified in this command, just voice payload traffic.



The CLI segment above shows an example where no more than 24 VoIP calls will be allowed (dial peer 800) over the egress WAN link. The 25th call will be hairpinned back to the PBX to be redirected to the PSTN (suitable digits are prepended to the dial string to direct the routing logic of the PBX). Note that 24 K is enough for 2 G.729 calls at 10.4 K each.



Note Other attributes are not present.



Trunk conditioning provides for connection-trunk networks, for example, networks with nailed-up voice connections across the VoX portion of the network. Trunk conditioning monitors the state of the VoX connection and busy-backs the trunk to the originating PBX if the VoX connection should fail.

This feature is limited in scope since it applies to connection-trunk networks only. On the other hand, most of the other CAC features apply only to switched networks. Doing CAC on a connection-trunk configuration is a slightly different problem than doing it for switched networks. This is because the VoX connections between the two GWs are nailed up, the bandwidth is therefore already established and allocated and must be available or the connection-trunk connections will not establish properly.

The unique attribute of this feature, as compared to other CAC features, is that it has visibility not only into the condition of the WAN (end-to-end) but also into the condition of the POTS connection on the terminating side of the network. In the figure above, if any one of legs A, B, C, or D should fail, the OGW will know this and can busy-back the trunk to the originating PBX to trigger rerouting capability at the source. The information is carried as part of the keepalives that are done on connection-trunk configurations anyway.

The precise bit pattern that will be generated to the originating PBX can be tuned, and it is from this functionality that the feature derives its name. The ABCD bits can be conditioned to specific busy or out-of-service (OOS) indications that the originating PBX will recognize and act upon.

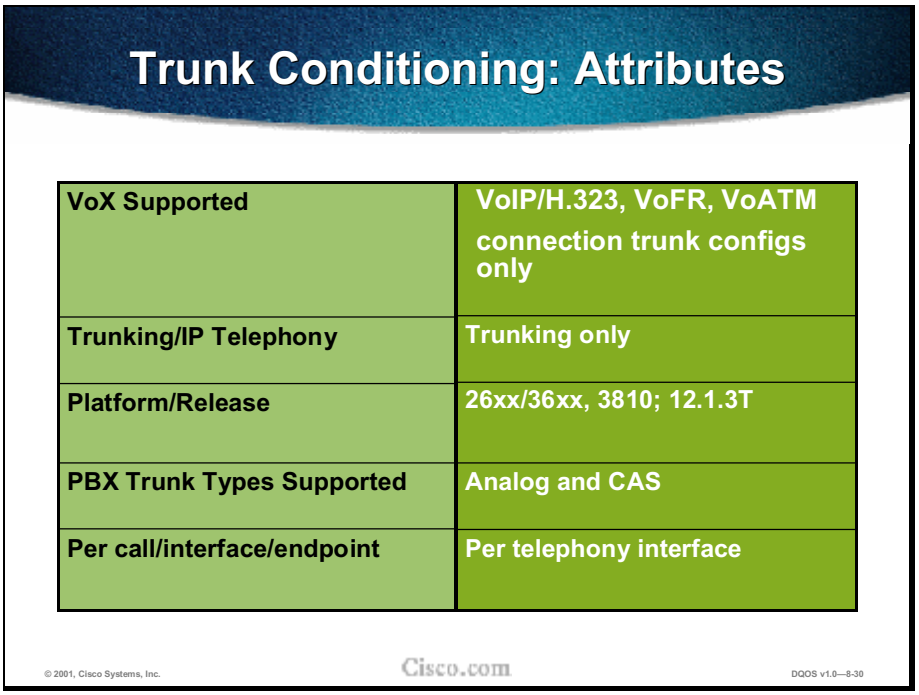
Trunk Conditioning Configuration Example

1) Create a voice class and define trunk-conditioning attributes

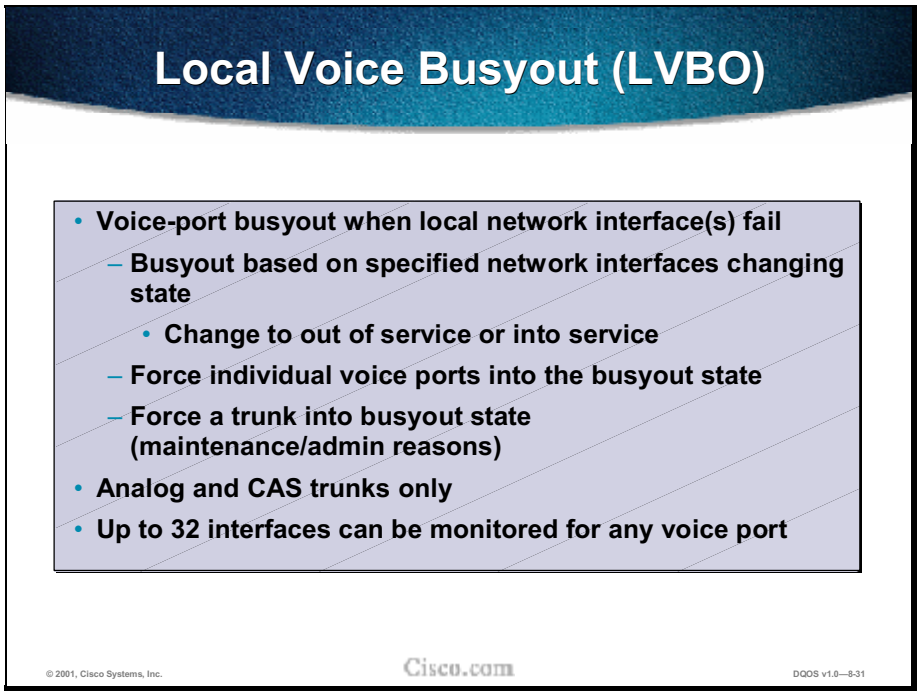
```
router(config)# voice class permanent 10
router(config-class)# signal keepalive 10
router(config-class)# signal pattern idle receive 0101
router(config-class)# signal pattern idle transmit 0101
router(config-class)# signal timing idle suppress-voice 5
router(config-class)# signal pattern oos receive 0001
router(config-class)# signal pattern oos transmit 0001
router(config-class)# signal timing oos timeout 60
router(config-class)# signal timing oos restart 120
router(config-class)# signal timing oos suppress-voice 30
```

2) Apply voice class to dial peers

```
router(config)# dial peer voice vofr 10
router(config-dial-peer)# voice-class permanent 10
router(config)# dial peer voice voatm 20
router(config-dial-peer)# voice-class permanent 10
```

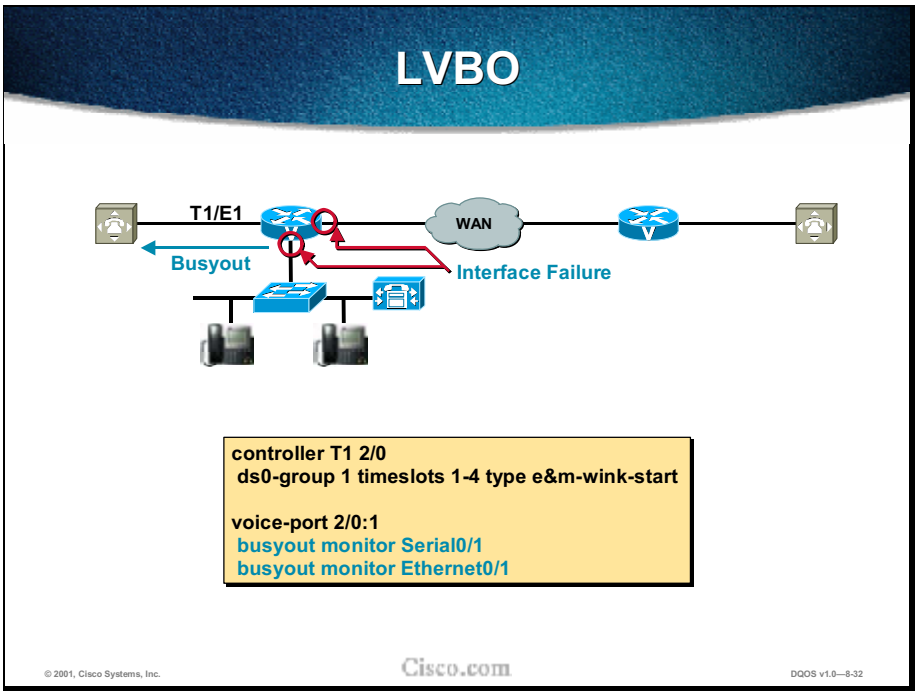


Note Other attributes are not present.

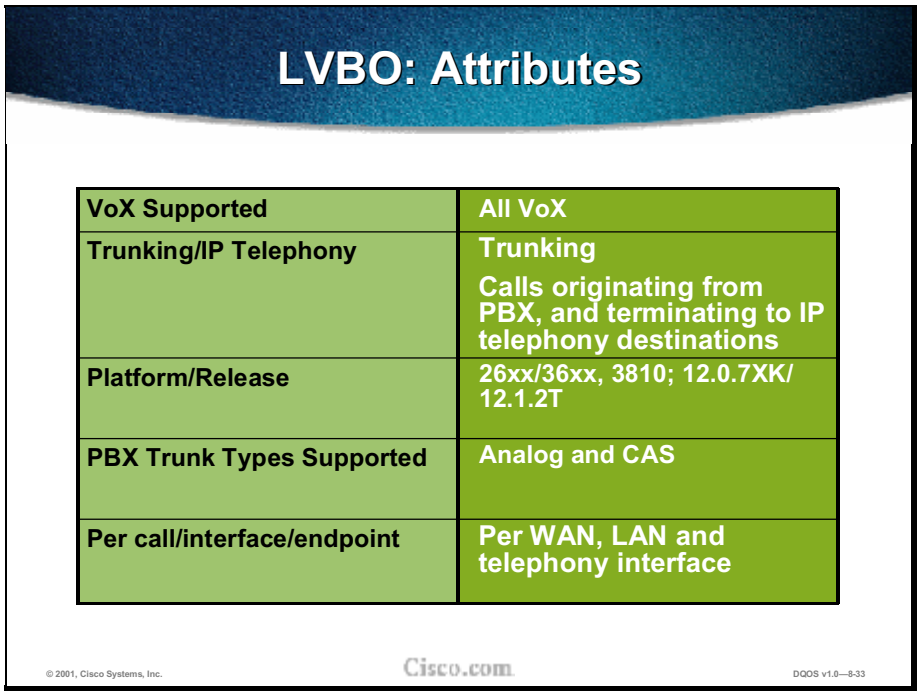


LVBO provides the ability, in the OGW, to monitor the state of various network interfaces, both LAN and WAN, and busy-back the trunk to the PBX when any of the monitored links should fail. Up to 32 interfaces can be monitored, and if either one of, or all of them, change state, the GW can be configured to busy-back the trunk to the PBX. The “local” in the name of the feature refers to the fact that only local links can be monitored; this is not a feature that has any visibility into the network beyond the links of the local GW.

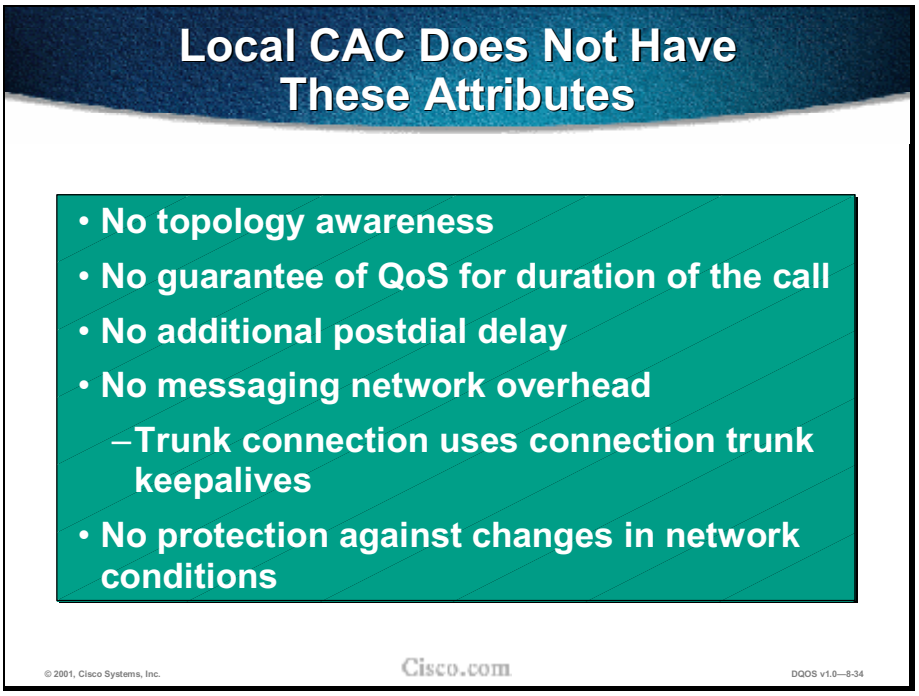
LVBO in current software works on CAS and analog PBX/PSTN trunks only. On CCS trunks, the cause code functionality can be used to inform the PBX/CO switch to redirect a rejected call. LVBO can be configured to force individual voice ports into a busyout state or force an entire T1/E1 trunk into busyout state.



The figure above illustrates the operation of the LVBO feature, including a CLI segment to show its configuration. In the example the OGW is monitoring two interfaces, Ethernet interface e0/1 and WAN interface s0/1 on behalf of voice port 2/0:1, a T1 CAS trunk to a PBX. As shown in the example, this feature is applicable only if the origination device is a PBX/PSTN interface, although the destination device can be anything, including an IP-capable voice device.

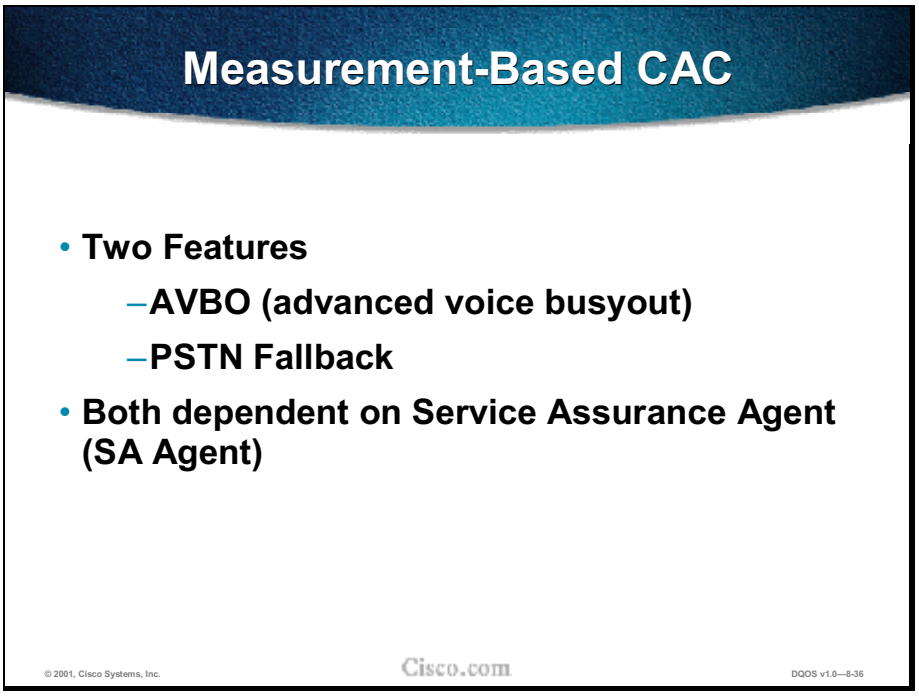


Note Other attributes are not present.



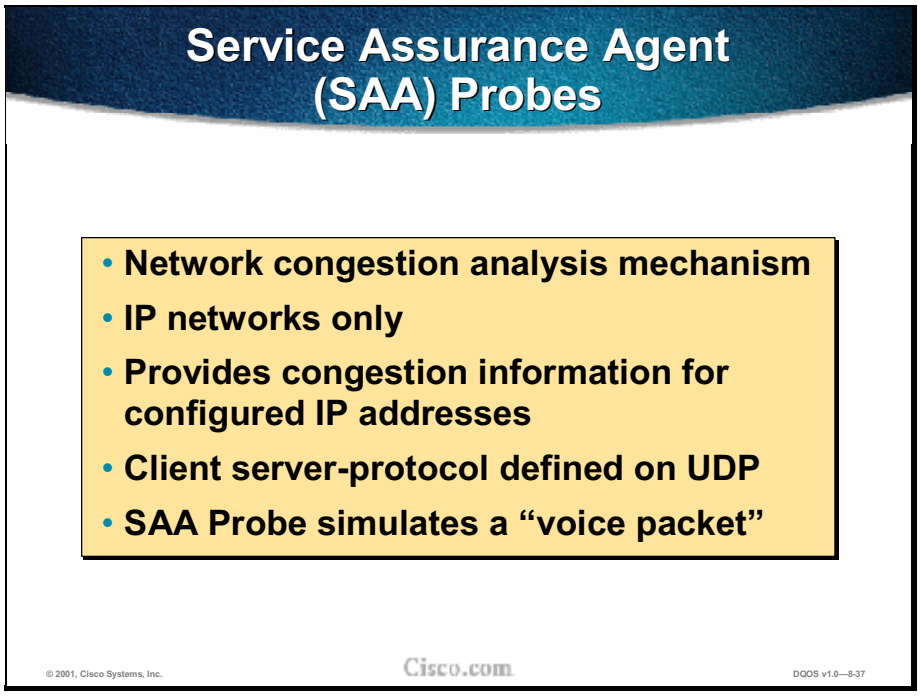
There are attributes that some CAC mechanisms have that do not apply to the local CAC mechanisms as indicated in the figure above. These attributes—topology awareness, guarantee of QoS for duration of the call, postdial delay, messaging network overhead, and protection against changes in the network—require visibility to the network. These are by definition the limitations of local CAC.


Measurement-Based CAC
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—8-35



The two types of measurement-based call admission control are advanced voice busyout(AVBO) and PSTN fallback.

Both are dependent on Service Assurance Agent (SA Agent), which will be taught, in greater detail, in the Management Tools chapter of this course.



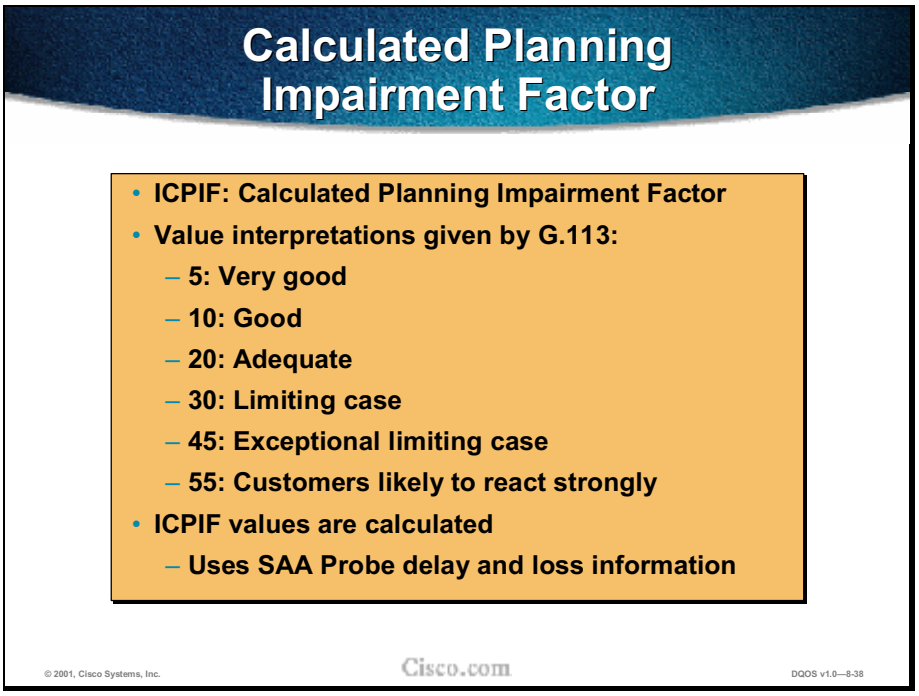
SAA is a generic network management IOS feature that provides a mechanism for network congestion analysis, and it underlies a multitude of other features. It was not implemented for CAC, nor is it part of the CAC suite itself. SAA probes traverse the network to a given IP destination and measure the loss and delay characteristics of the network along the path traveled. These values are returned to the OGW to use in making a decision on the condition of the network and its ability to carry a voice call.

The SAA feature was called RTR (Response Time Reporter) in earlier releases of software and older Cisco documentation.

The SAA feature is a client-server protocol defined on UDP. The client builds and sends the probe, and the server (previously the RTR Responder) returns the probe to the sender. The SAA probes used for CAC go out randomly on ports selected from within the top end of audio UDP-defined port range (16384 to 32767); they use a packet size based on the codec the call will use. IP Precedence can be set if desired, and a full RTP/UDP/IP header is used as per the header a real voice packet would carry. By default the SAA probe uses the RTCP port (the odd RTP port number), but can also be configured to use the RTP media port (the even RTP port number) if desired.

SAA is a Cisco proprietary protocol, first introduced on selected platforms in 12.0.7T

- Supported on 1600, 2500, 2600, 3600, 4000, 5400, 7200, 7500, 7700, VG200.
- Lower-end IOS platforms tend not to (for example, 1750).
- CM does not support SAA probes.
- IP phones do not support SAA probes.

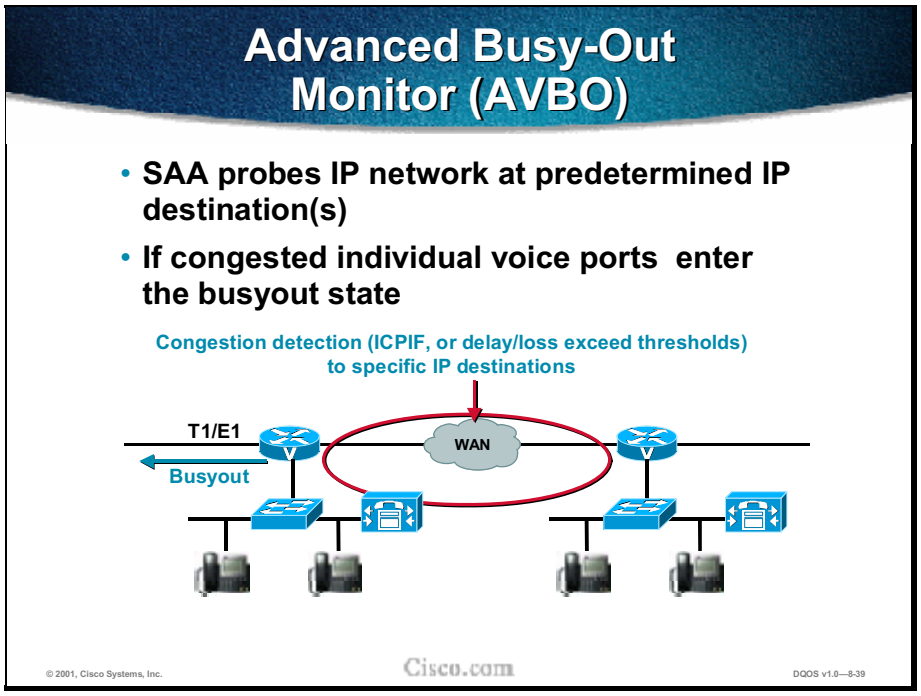


The ITU standardizes network transmission impairments in ITU G.113. This standard defines the term ICPIF (Calculated Planning Impairment Factor), which is a calculation, based on network delay and a packet loss figure, that yields a single value that can then be used as a gauge of network impairment.

ITU G.113 provides the following interpretations of specific ICPIF Values:

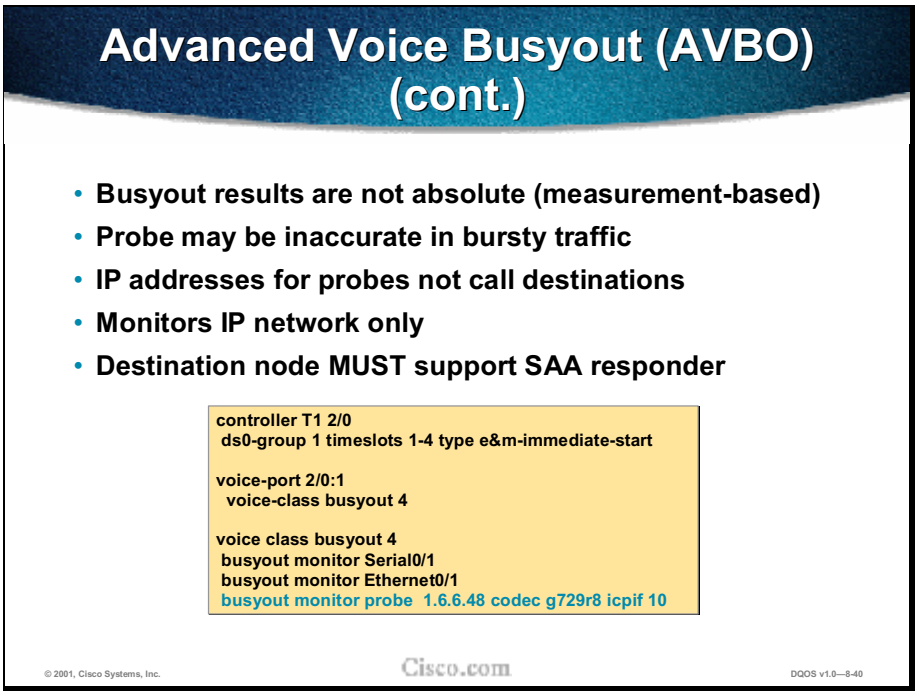
- 5: Very good
- 10: Good
- 20: Adequate
- 30: Limiting case
- 45: Exceptional limiting case
- 55: Customers likely to react strongly

SAA probe delay and loss information is used in calculating an ICPIF value that is then used as a threshold for CAC decisions, based either on the ITU's interpretation (above) or on the requirements of an individual customer network.



AVBO, as the name suggests, is an enhancement to LVBO. While LVBO provides for busyout based on local conditions of the OGW, AVBO adds the capability to trigger an SAA probe to one or more given (configured) IP destinations. The information returned by the probe, either the explicit loss or delay values, or the ICPIF congestion threshold, can be used to trigger a busyout of the connection to the PBX.

AVBO therefore introduces the ability to do PBX trunk, or individual voice ports, busyout based on the current conditions of the IP network (the “cloud”). This is illustrated in the figure above.



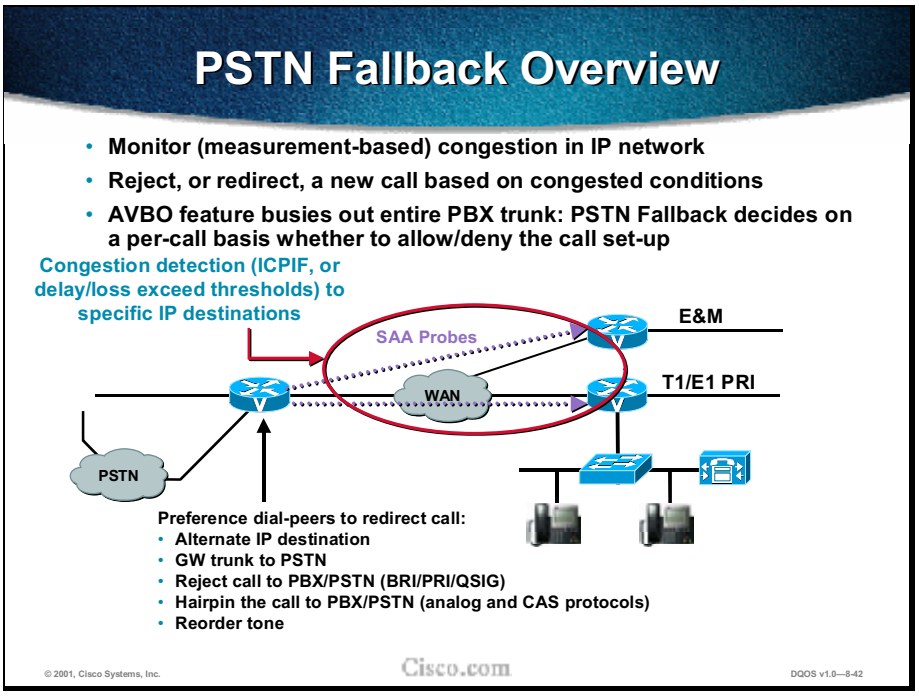
The CLI segment above shows an example of configuring AVBO on a T1 CAS trunk to a PBX. Some things to keep in mind about this feature include:

- Busyout results based on probes (measurement-based) are not absolute—there are therefore conditions where a false positive will happen.
- The IP addresses monitored by the probes are statically configured (as shown above). It is necessary to ensure, manually, that these IP addresses are indeed the destinations to which calls are being made. There is no automatic coordination between the probe configuration and the actual IP destinations to which the VoIP dial peers or a gatekeeper (GK) may direct calls.
- The destination node must support an SAA responder (the device that owns the IP address to which the probe is sent).
- This feature cannot busy-back the local PBX trunk based on the state of the telephony trunk on the remote node—it monitors the IP network only.
- SAA probe-based features do not work well in networks where traffic load fluctuates dramatically in a short period of time.
- As with LVBO, this feature can be applied only to analog and CAS trunks; CCS trunks are not yet supported.

AVBO: Attributes

VoX Supported	VoIP only
Trunking/IP Telephony	Trunking Calls originating from PBX, and terminating to IP telephony destinations
Platform/Release	26xx/36xx, 3810; 12.1.3T
PBX Trunk Types Supported	Analog and CAS
End-to-end/Local/IP Cloud	IP cloud
Per call/interface/endpoint	Per IP destination
Messaging Network Overhead	Periodic SAA probes

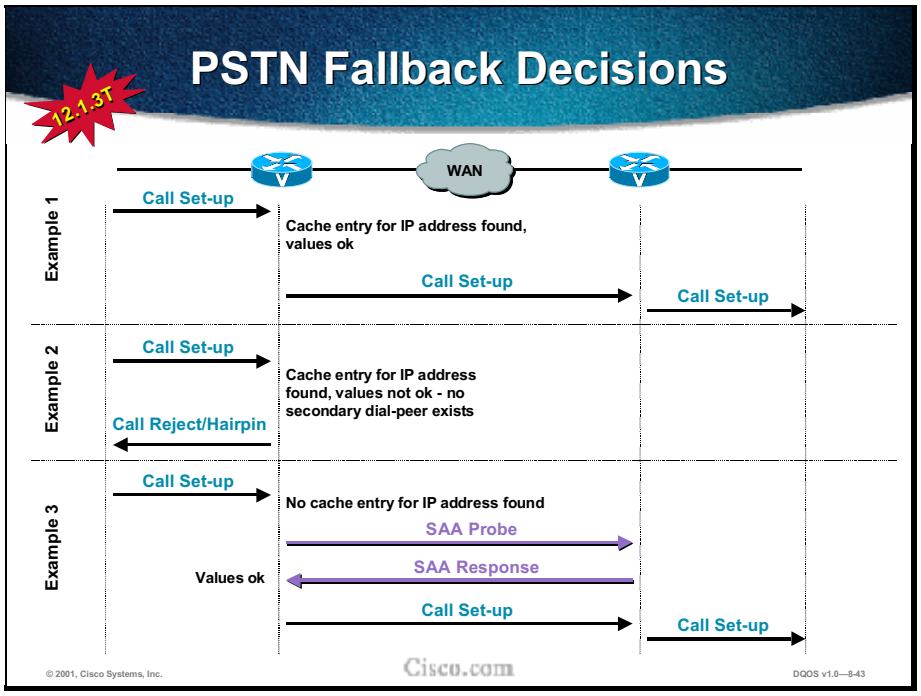
No topology awareness, no guarantee of QoS for duration of call,
No protection against changes in network conditions



PSTN fallback is a per-call CAC mechanism, that is, it does not busyout any trunks or provide any general indication to the attached PBX that the IP cloud is not capable of taking calls. The CAC decision is triggered only when a call setup is attempted.

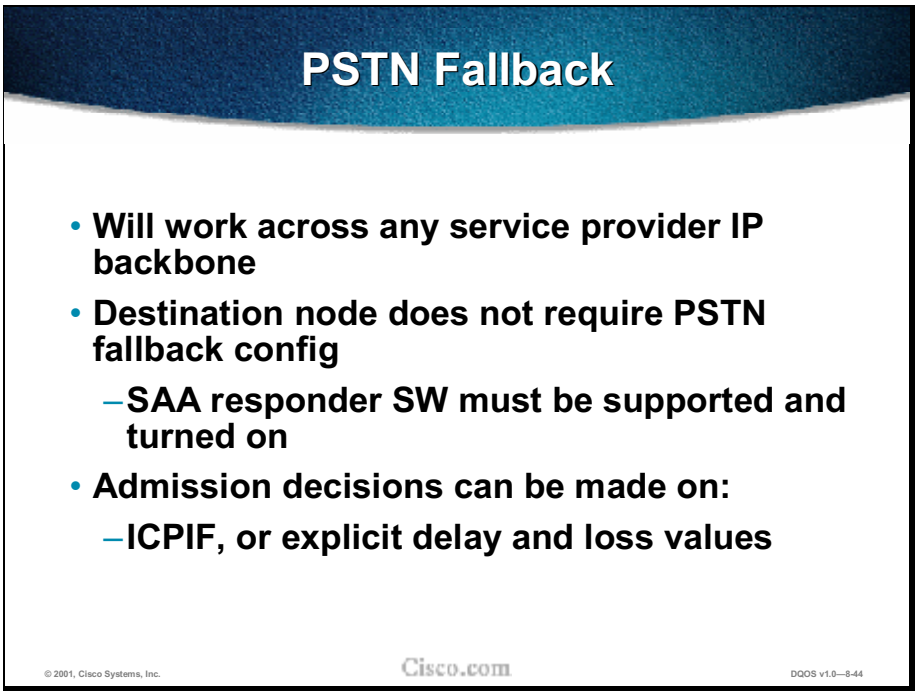
As PSTN fallback is based on SAA probes, it has all the benefits and drawbacks of measurement-based techniques. It is unusually flexible in that it can make CAC decisions based on any type of IP network, including the Internet. All IP networks will carry the SAA probe packet as just another IP packet. It therefore does not matter if the customer backbone network comprises one or more SP networks, and/or the Internet, and/or any combination of these. The only requirement is that the destination device (the owner of the IP address the probe is sent to) must support SAA responder functionality.

This destination device, hopefully, is part of the customer's network at the destination site (with an SP backbone in between). PSTN fallback can therefore not be used directly to IP phones and PC-based VoIP application destinations, but can indirectly be used if these destinations are behind an IOS router that can support the SAA responder. This destination device does not need to support the PSTN fallback feature (that is an OGW feature only); only the SAA probe responder is needed.



PSTN fallback does not require the static configuration of the IP destinations. A cache is kept (of configurable size) by the software that tracks the most recently used IP destinations to which calls were attempted. If the IP destination of a new call attempt is found in the cache, the CAC decision for the call can be made immediately (examples 1 and 2 in the figure above illustrate “call allowed” and “call rejected” scenarios respectively). If the entry does not appear in the cache, a new probe is started and the call setup is suspended until the probe response arrives (example 3 above). Therefore, an extra postdial delay is imposed *only* for the first call to a new IP destination.

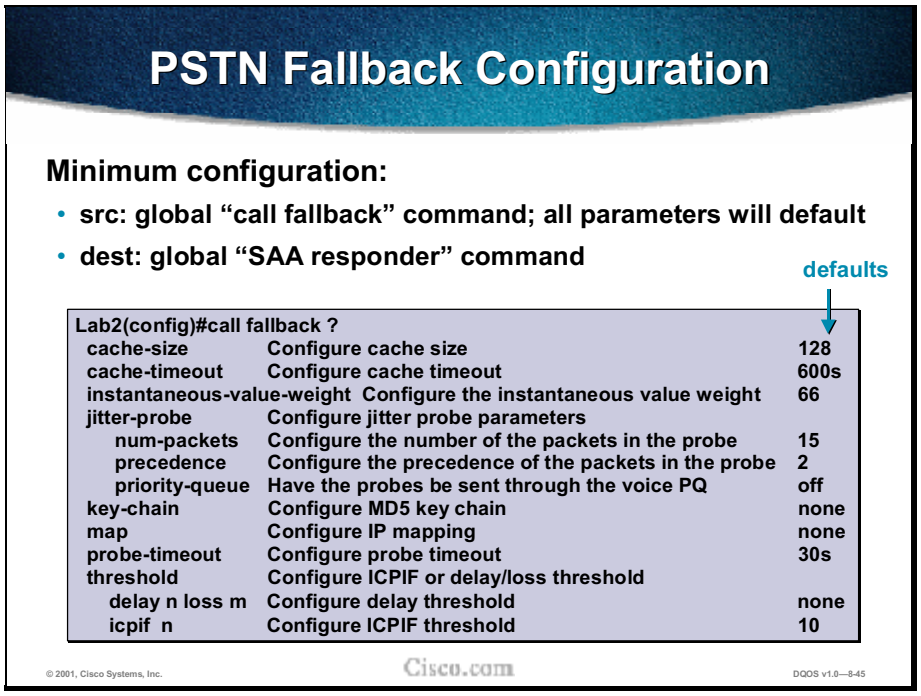
Once an IP destination has been entered into the cache, a periodic (configurable timeout) probe is sent to that destination to refresh the information in the cache. If no further calls are made to this IP destination, the entry will age (configurable inactivity timer) out of the cache and probe traffic to that destination will be discontinued. PSTN fallback therefore dynamically adjusts the probe traffic to the IP destinations that are actively seeing call activity.



PSTN fallback will work across any SP IP backbone. The destination node does not require PSTN fallback configuration; however, the SA Agent responder software must be supported and turned on. When the SA Agent starts on first call to a new IP destination, the entry is inserted into cache. The probe is repeated periodically while call activity continues and stops after the inactivity timer expires, at which point the entry is removed from the cache. Each probe consists of multiple packets—a configurable parameter of the feature. The delay, loss, and ICPIF values entered into the cache for the IP destination are calculated (averaged) from all the responses. Admission decisions can be made on ICPIF or explicit delay and loss values.

If the call uses the G.729 and G.711 codecs, the probe packet sizes will mimic those of a voice packet for that codec. Other codecs will use G.711-like probes. In software releases later than 12.1.(3)T, other codec choices may also be supported with their own exact probes.

The IP Precedence of the probe packets can be configured in order to mimic the priority of a voice packet more closely. This should be set equal to the IP Precedence used for other voice media packets in the network.



The PSTN fallback feature configuration applies to the OGW, that is, for calls initiated by the platform. It has no bearing whatsoever on calls received by the platform. The destination node (often the TGW, but not necessarily) should be configured with the SAA Responder feature. In most networks GWs generate calls to each other, so that every GW is both an OGW and a TGW. But in some networks, more often SP networks, call traffic direction is occasionally one-sided, either outgoing or incoming.

The PSTN fallback configuration is done at the global level and therefore applies to all calls attempted by the GW. The feature cannot be selectively applied only to calls initiated by certain PSTN/PBX interfaces.

There are a number of parameters that can be tuned for PSTN fallback, and all these parameters have defaults. The minimum necessary CLI to turn on the feature is simply:

OGW: the global “call fallback” command

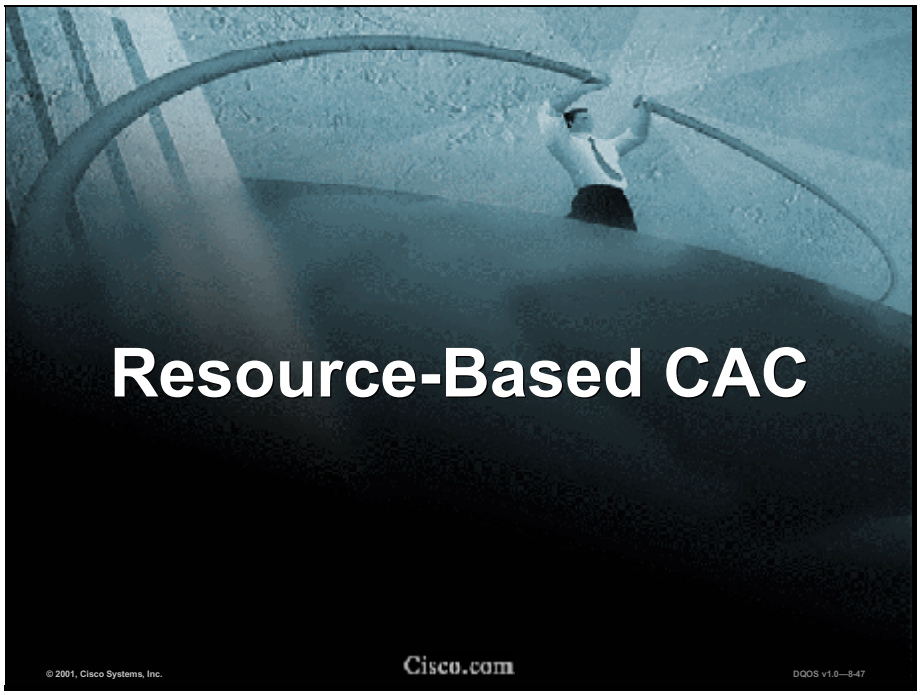
Destination node: global “SAA responder” command

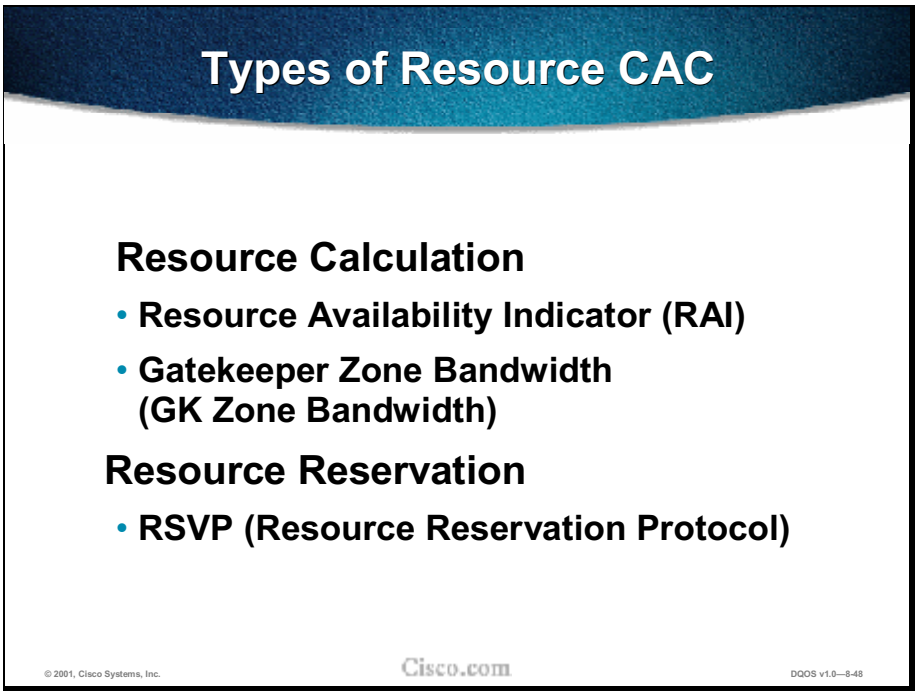
The CLI segment above shows the different parameters, and their defaults in 12.1.(3)T software, that can be tuned for PSTN fallback. Please consult the IOS feature documentation for a full discussion of what each parameter does.

PSTN Fallback: Attributes

VoX Supported	VoIP only
Trunking/IP Telephony	Trunking Calls originating from PBX, and terminating to IP telephony dest's
Platform/Release	26/36, 3810: 12.1.3T 5300: 12.2.2T 7200/7500 support SAA responder
PBX Trunk Types Supported	All PBX/PSTN trunk signaling types (analog, Digital CAS and CCS) for analog and digital CAS - alternate IP destination, hairpin for digital CCS - reject the call to PBX/PSTN for rerouting
End-to-end/Local/IP Cloud	IP cloud
Per call/interface/endpoint:	Per active/cached IP destination
Postdial Delay	Extra on first call that kicks off probe; none on subsequent calls
Messaging Network Overhead	Periodic SAA probes per active IP destination in the cache

No topology awareness, no guarantee of QoS for duration of call, no protection against changes in network conditions


Resource-Based CAC
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—8-47



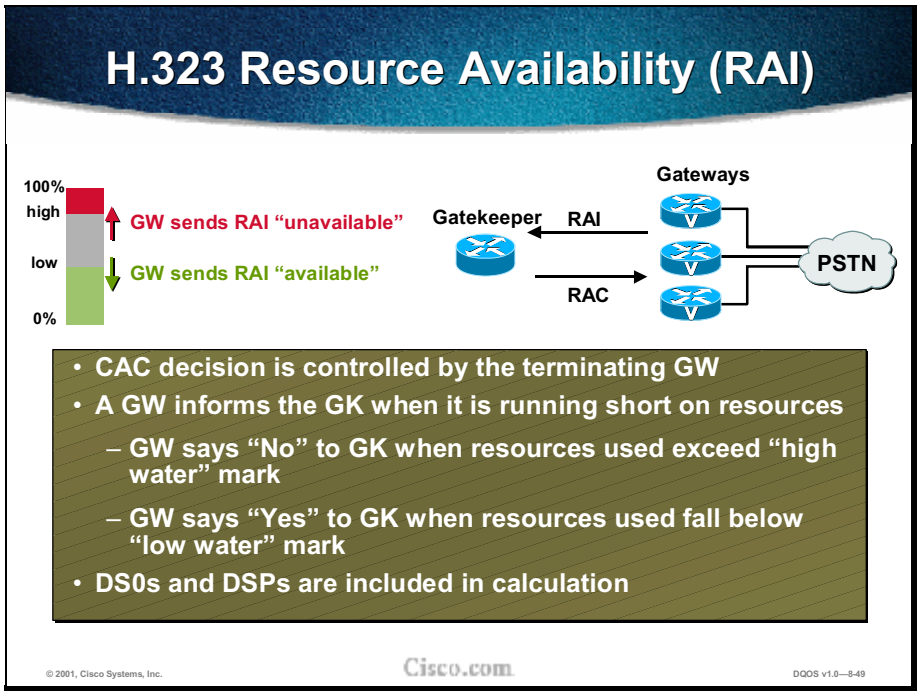
We now turn our attention to the resource-based CAC techniques. These are the second of two types of techniques that add visibility into the network itself, in addition to the local information on the OGW that can be used for CAC as discussed in the preceding sections.

There are two types of resource-based CAC mechanisms: those that monitor the use of certain resources and then calculate a value that drives the CAC decision, and those that reserve resources for the call. In the first case, resource availability indication (RAI) and gatekeeper (GK) zone bandwidth calculate resources. RSVP is the only reservation mechanism that can guarantee QoS for the duration of the call. All the other CAC mechanisms (local, measurement-based, and resource-calculation-based) simply make a one-time decision before call setup, based on knowledge of network conditions at that time.

General resources of interest to a voice call include:

- DS0 timeslot on the originating and terminating TDM trunks
- DSP resources on the originating and terminating GWs
- CPU use of the nodes, typically the GWs
- Memory use of the nodes, typically the GWs
- Bandwidth availability on one or more links in the path the call will take

The resource-calculation methods taught here in current IOS software (12.1.5T and/or 12.2 mainline) take the TGW's DS0 and DSP availability into account RAI and bandwidth at a high level (GK zone bandwidth management). The resource reservation mechanism (RSVP) takes only bandwidth availability into account.

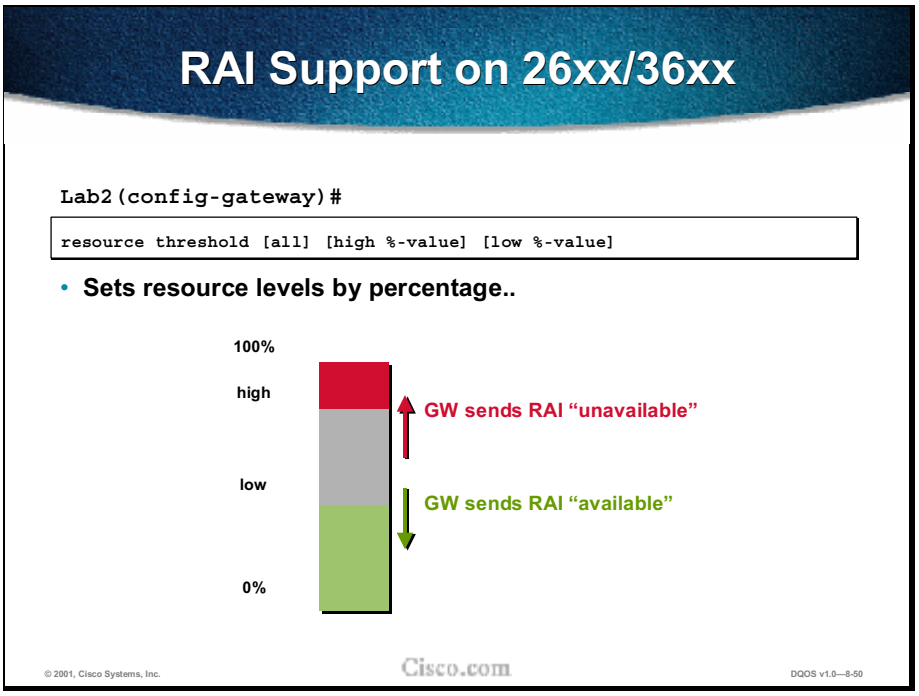


RAI CAC applies only to H.323 voice networks that utilize a GK design. RAI is a simple toggle (yes/no) indication that the TGW sends to the GK to control the routing (or not) of subsequent voice calls to it.

RAI as a CAC mechanism is unique in its ability to provide information on the terminating POTS connection and is controlled by the TGW; in all the other methods; the CAC decision is controlled by the OGW or by the GK.

The gateway reports its resource status to the gatekeeper using the RAS RAI. When a monitored resource falls below a configurable threshold, the gateway sends an RAI to the gatekeeper, indicating that the gateway is almost out of resources. When the available resources then cross above another configurable threshold, the gateway sends an RAI, indicating that the resource depletion condition no longer exists.

Using the **resource threshold** command configures resource-reporting thresholds. The upper and lower thresholds are separately configurable to prevent the gateway from operating sporadically due to the availability or lack of resources.

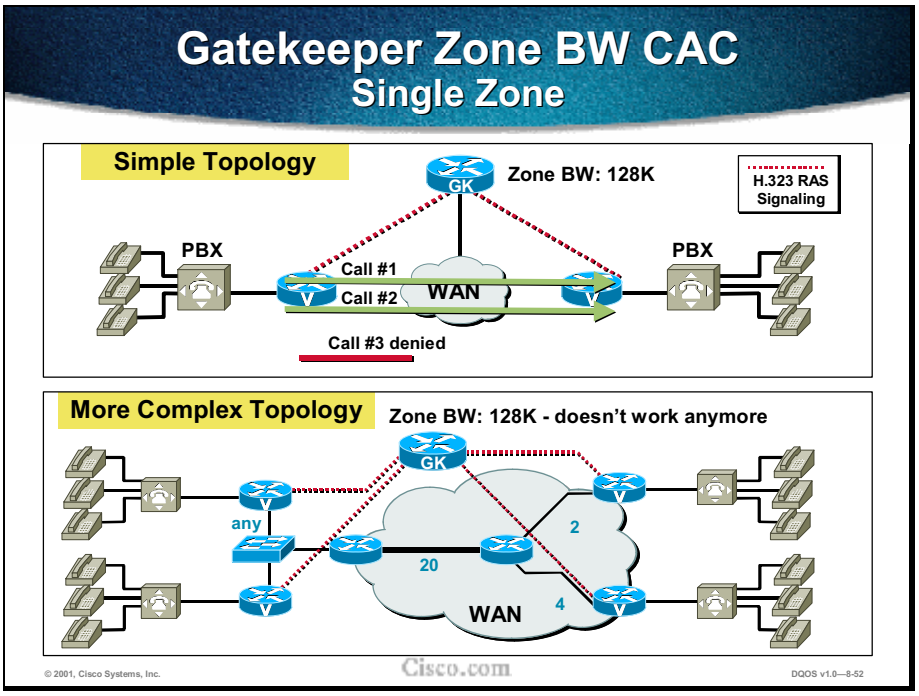


RAI on the GW is configured with high-water mark and low-water mark thresholds, as shown in the figure. When resource use (using the calculation algorithm given earlier) goes above the high-water mark (configured as a percentage), "RAI [unavailable]" is sent to the GK. When resource availability falls below the low-water mark, "RAI [available]" is sent to the GK. To prevent hysteresis based on a single call's arrival or disconnection, the high and low marks should be configured some percentage points apart.

RAI: Attributes

VoX Supported:	VoIP/H.323 only
Trunking / IP Telephony:	Trunking Potentially IP telephony, but CM does not yet support RAI
Platform/Release	5300: 12.0.5T 26/36 T1/E1: 12.1.2XH / 12.1.3T
PBX Trunk Types Supported	All
End-to-end / Local / IP Cloud	Local (at the terminating GW) DSP and DS0 resources; algorithm platform dependent
Per call / interface / endpoint	Per GW
Messaging Network Overhead	Occasional RAI toggle between GW and GK
Post-dial delay	None

No topology awareness, no guarantee of QoS for duration of call, no protection against changes in network conditions

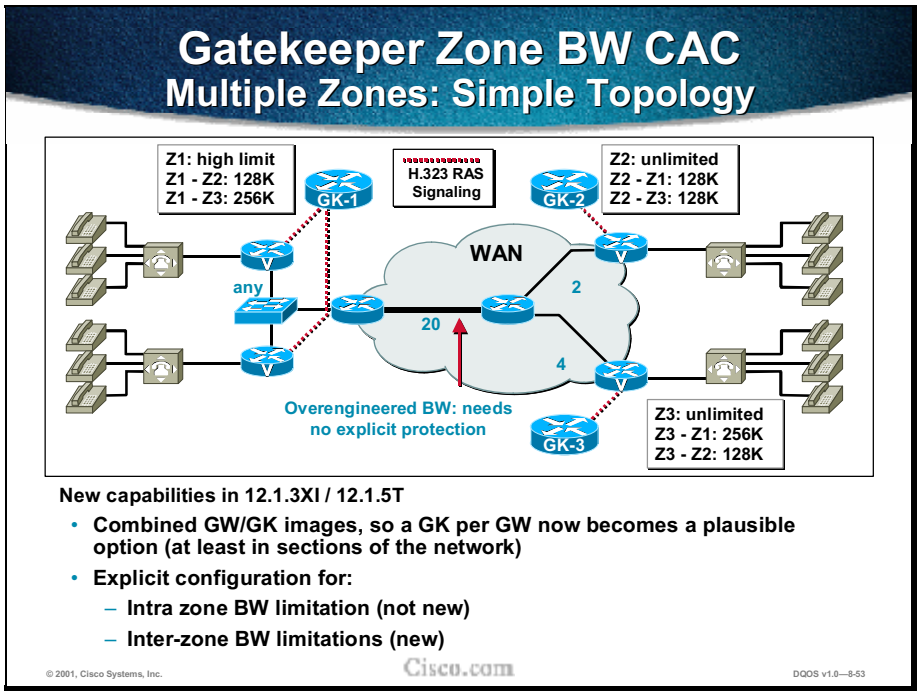


Zone management is one of the primary functions of an H.323 GK. With 12.1.(5)T / 12.2 mainline, the GK is able to limit both the bandwidth of calls in its local zone, as well as the bandwidth used between its own zone and any other remote zone in the network.

The figure on the top shows a single-zone GK network with two GWs—this illustrates GK CAC in its simplest form. If the WAN bandwidth of the link between the two GWs can carry no more than two calls, the GK has to be configured such that it denies the third call. Assuming every call is 64 K, the GK is configured with a zone bandwidth limitation of 128 K to achieve CAC in this simple topology.

Most networks, however, are not so simple. The lower figure shows a more complex topology but one still configured as a single-zone network. In this topology, the legs in the WAN topology each have separate bandwidth provisioning and therefore separate capabilities of how many voice calls can be carried across that leg. The numbers on the WAN legs in the lower figure show the maximum number of calls that can be carried across that leg.

Consider now that the GK zone bandwidth is still set to a maximum of 128 K, thus allowing no more than two simultaneous calls. This is the desired behavior of the network if both calls involve Site 1—the GK will protect the bandwidth of the WAN link from Site 1 to the WAN aggregation point by not allowing more than two calls across that link. But if both calls are within the headquarters site, there is no point in limiting to only two calls because there is plenty of bandwidth in the campus backbone.



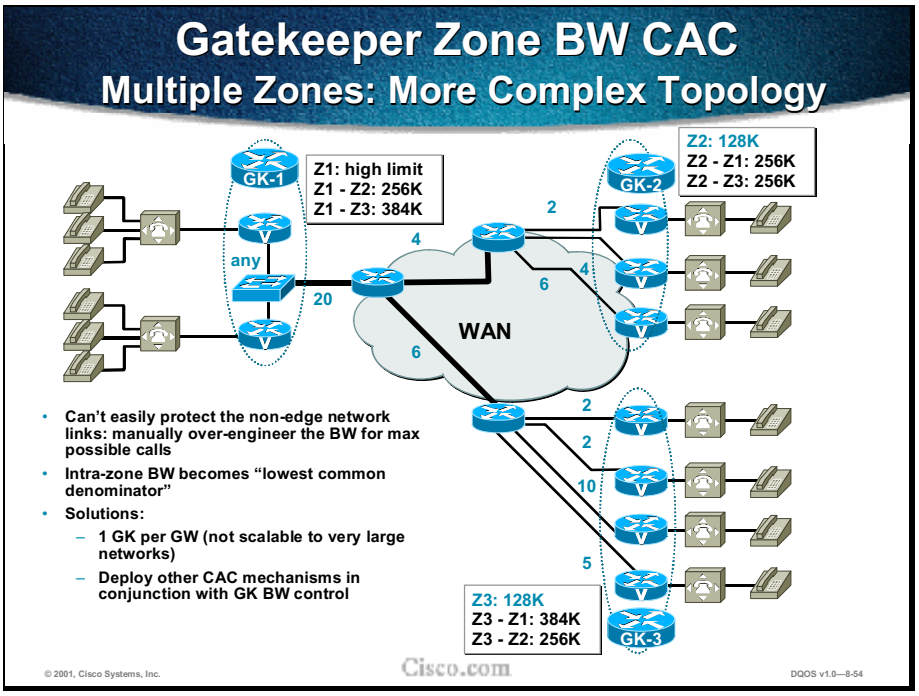
To solve the single-zone problem of reducing the network to the lowest common denominator (the lowest capacity WAN link anywhere), we can design the network with multiple GK zones. A good starting point is one zone per site as shown in the figure.

The Site 1 GK will now limit the number of calls active in Site 1 (regardless of where those calls originated or terminated) to two (128 K). Since there is only a single GW at Site 1, there is no need to configure a limit for the intrazone call traffic. All interzone traffic is limited to two calls to protect the WAN link connecting Site 1.

At Site 2 there is also a single GW and therefore no need to limit the intrazone call traffic. There are separate interzone limits for:

- Calls between Site 2 and the Headquarters site (here the limiting factor is the maximum of four calls on the WAN link connecting Site 2).

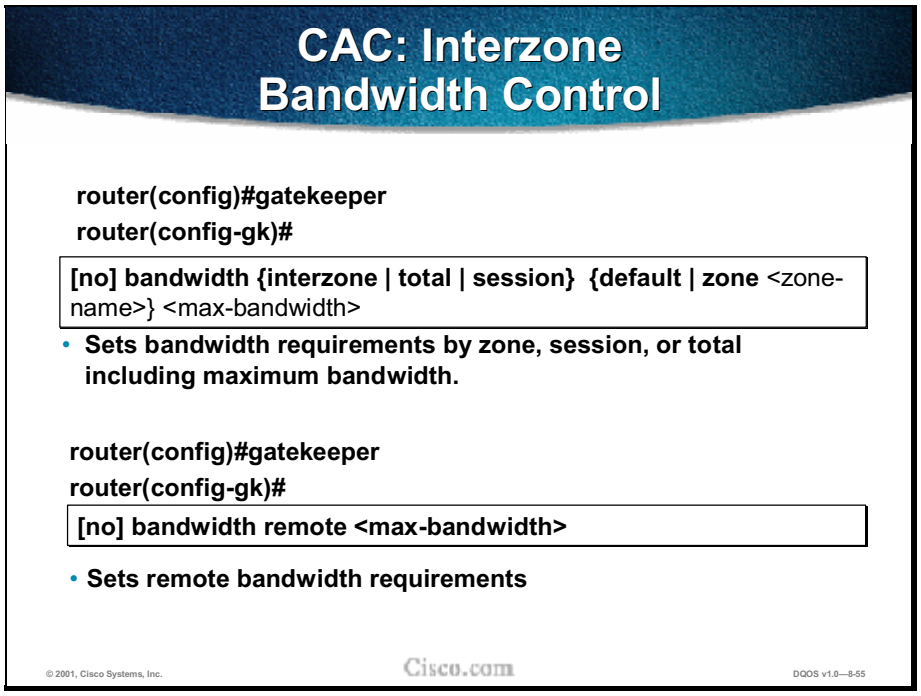
- Calls between Site 2 and Site 1 (here the limiting factor is the maximum of two calls on the WAN link connecting Site 1). The Headquarters site has a similar configuration, except that calls are unlimited within the site, in this scenario not because there is a single GW but because there is ample bandwidth between the GWs at that site.



In the preceding network topology, GK CAC provides sufficient CAC granularity to protect voice traffic across the low-speed WAN access links. But let's consider another network topology where we have multiple GWs per zone, while each GW (the remote sites) has a separate WAN link to the aggregation point. This is shown in the figure above.

Of the three GWs in Remote Site 1, the lowest WAN access link can carry a maximum of two simultaneous calls. As the bandwidth limitation is configured per zone, and not per GW, there is no facility within GK CAC to limit the calls to specific GWs within the zone. Our best choice, therefore, is to configure the network for the lowest common denominator link: For both remote Sites 1 and 2 this is 128-K bandwidth, or two calls.

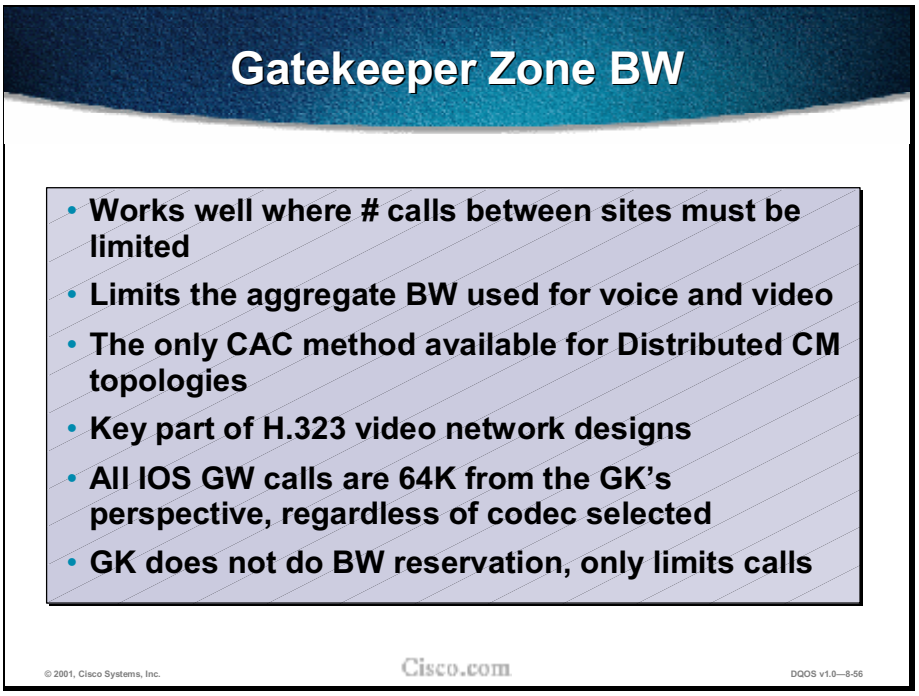
This configuration will ensure proper voice quality at all times, but it is also wasteful of the GWs that could terminate more calls without oversubscribing their WAN bandwidth. In this network configuration, CAC will be activated too soon and deflect certain calls over the PSTN, when in fact they could have been carried by the WAN. So in this type of topology, GK CAC is not sufficient to protect voice quality over the WAN link, as well as optimize the bandwidth use of all WAN links.



As of 12.1.(5)T, the following types of zone bandwidth limitations can be configured on the GK:

- The maximum bandwidth for all H.323 traffic between the local zone and a specified remote zone (configuration can be repeated individually for each remote zone if desired)
- The maximum bandwidth allowed for a single session in the local zone (typically used for video applications, not for voice)
- The maximum bandwidth for all H.323 traffic allowed collectively to all remote zones

Note Older **zone bw** command will be changed to **bandwidth** command during SW upgrade.



GK (gatekeeper) CAC works well in network designs where the desire is to limit the number of calls between sites. This may be due either to bandwidth limitations or to business policy. If it is done for bandwidth limitations on the WAN legs, manual calculations can be done to translate the maximum number of calls to be allowed between sites into a bandwidth figure that will cause the GK to deny calls beyond that number. GK CAC limits the aggregate bandwidth used for local zones, and now with IOS releases 12.1.3XI / 12.1.(5)T GK, CAC can limit interzone aggregate bandwidth.

A major advantage of GK CAC is that it is the only CAC method that can incorporate mixed networks of IOS GWs and CallManagers with IP phones.

GK zone bandwidth control is a key part of H.323 video network designs—here bandwidth is more of an issue because different video sessions can request different sizes of video transmissions making the manual calculation method easily used for voice almost unusable. Video also uses a lot more bandwidth per session than voice.

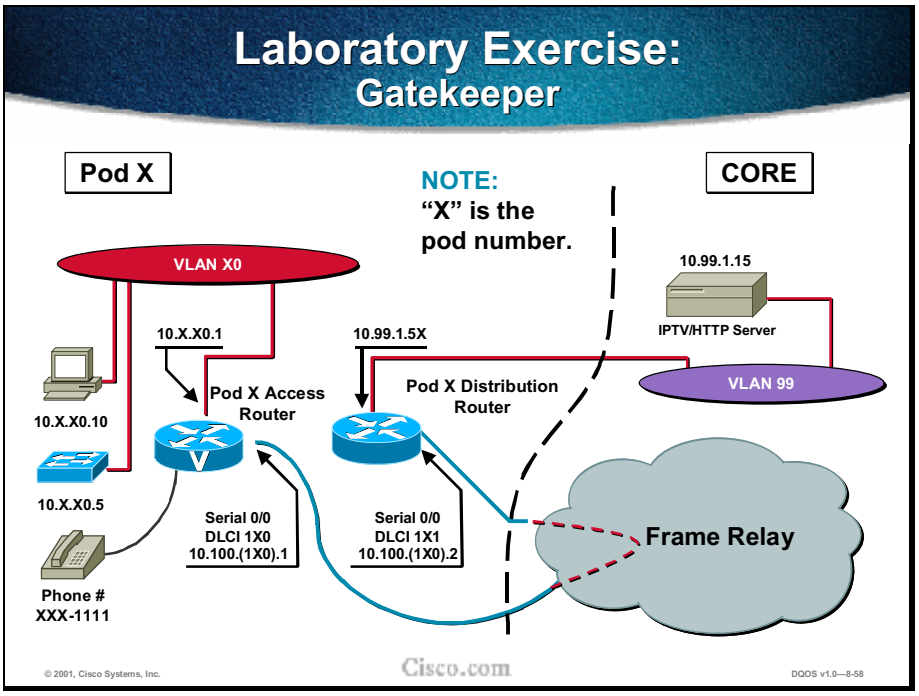
The GK does not have any network topology knowledge and does not know how much bandwidth is truly available for calls. What the GK does is take a fixed amount of bandwidth—statically configured on the GK as we've seen in the preceding network examples—then subtract a certain amount of bandwidth for each call setup. Bandwidth is returned to the pool when a call is disconnected. If a request for a new call causes the remaining bandwidth to be decremented below zero, the call is denied. The GK therefore does *not* do bandwidth reservations of any kind; it merely does a static calculation to decide if a new call should be allowed or not.

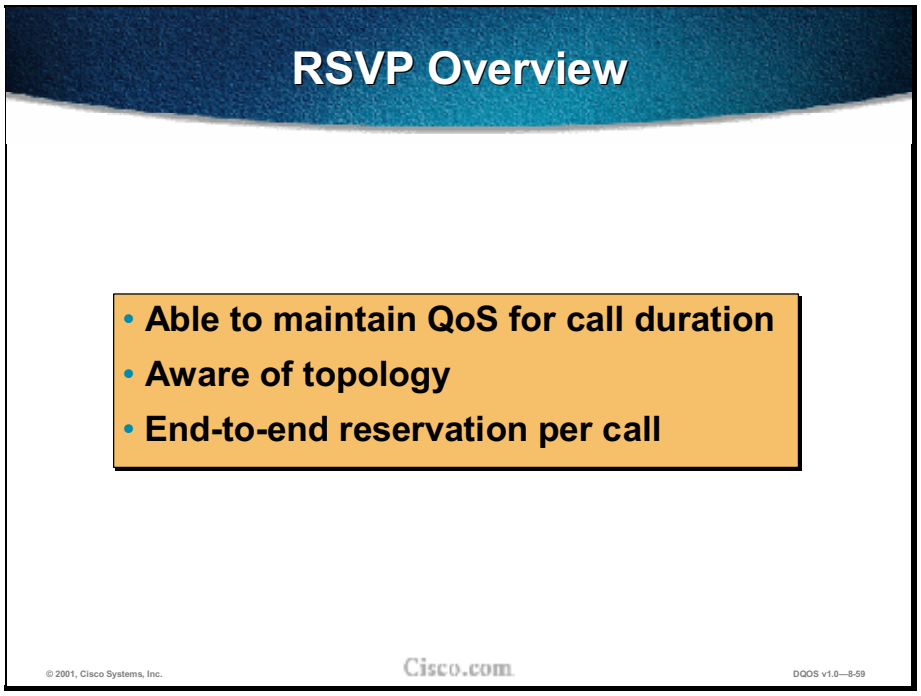
IOS GWs assume 64 kbps in current software. The codec and other bandwidth-determining features (such as CRTP) are not currently being taken into account when a call's bandwidth is considered by the GK zone bandwidth calculation.

GK Zone BW: Attributes

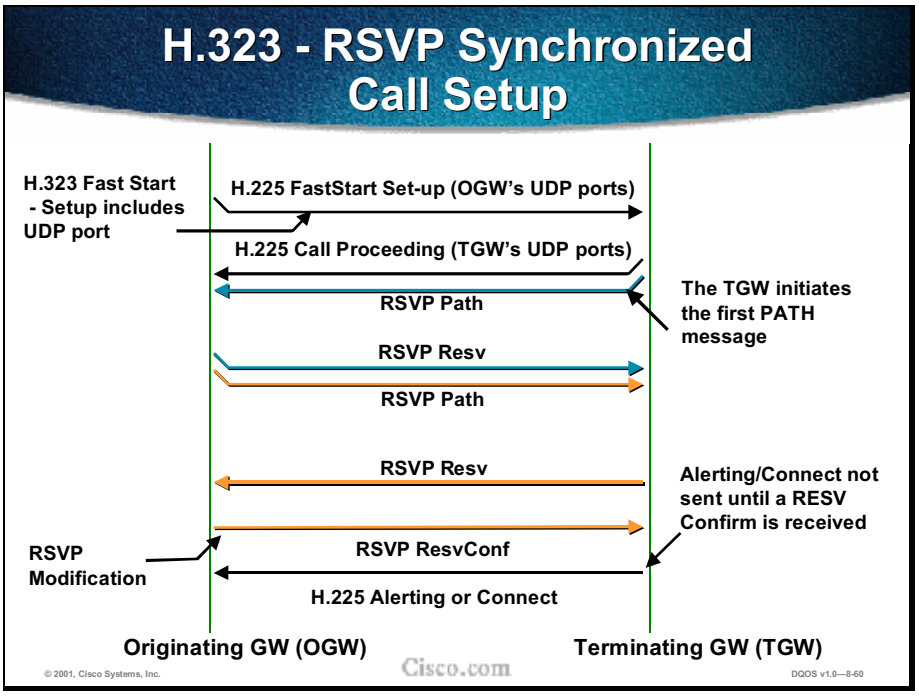
VoX Supported	VoIP/H.323 only
Trunking/IP Telephony	Trunking and IP telephony Some caveats if both CM and IOS GWs used in the same zone
Platform/Release	IOS GWs since 11.3 CM has recent changes in E.164 registration, and BW requested per call
PBX Trunk Types Supported	All
End-to-end/Local/IP Cloud	E2E between OGW and TGW, although not aware of the network topology (BW availability) in between
Per call/interface/endpoint	Per Call
Messaging Network Overhead	Part of the GK RAS messaging
Post-Dial Delay	None

No topology awareness, no guarantee of QoS for duration of call, no protection against changes in network conditions



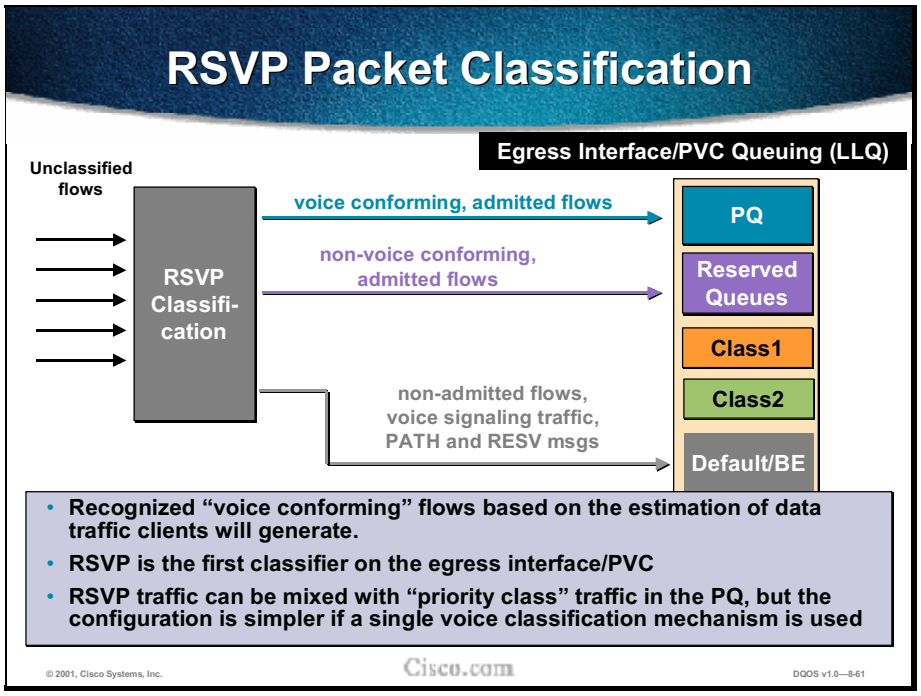


RSVP (Resource Reservation Protocol) differs from all other methods of conducting call admission control introduced in this chapter. It is the only CAC method that does a bandwidth reservation and not just a “best guess look-ahead” before the call is set up to drive the call admission decision. This gives RSVP the unique advantage of not only providing CAC for voice, but also guaranteeing QoS for the duration of the call against changing network conditions.



IOS release 12.1.(5)T offers the feature of synchronization of H.323 call setup messages with RSVP reservation messages.

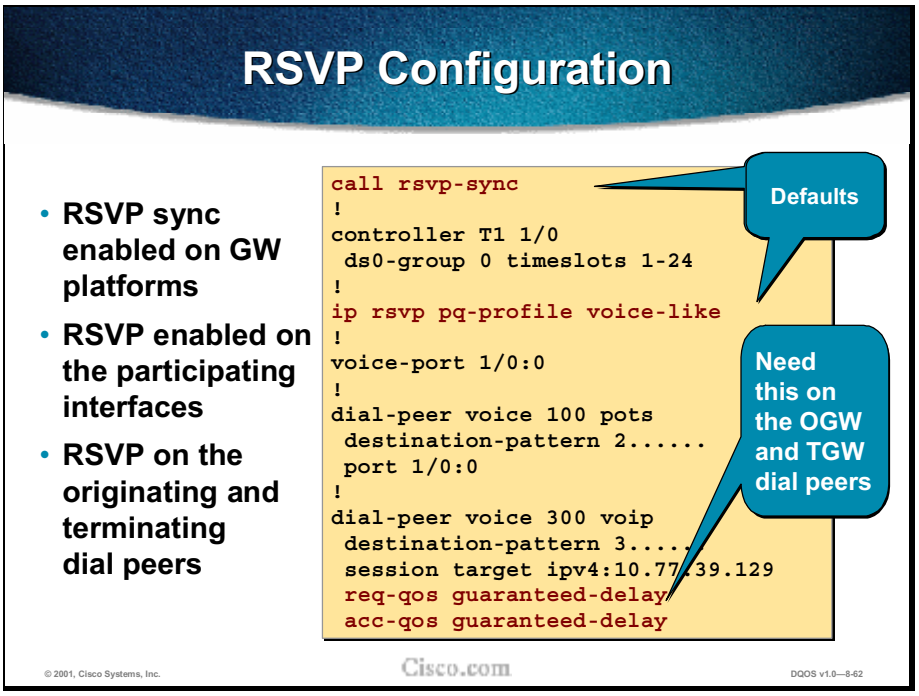
The H.323 setup is suspended before the destination phone starts ringing (triggered by the H.225 alerting message). The RSVP reservation is done in both directions since a voice call requires a two-way speech path and therefore bandwidth in both directions. The TGW ultimately makes the CAC decision based on whether or not both reservations succeed. At that point the H.323 state machines continue either with an H.225 Alerting/Connect (the call is allowed and proceeds), or with an H.225 Reject/Release (the call is denied). The RSVP reservation is in place by the time the destination phone starts ringing and the caller hears ring-back.



RSVP, as a general IOS feature, has its own set of reserved queues within WFQ for traffic with RSVP reservations. These queues, though they have a low weight, are separate from the priority queue (PQ), and packets in reserved queues do not get priority (other than by virtue of their low weight) over packets from other queues. It has long been known that this treatment (a low-weight queue inside weighted fair queuing [WFQ]) is insufficient for voice quality over a congested interface with several different flows of traffic. When RSVP is configured for a voice call, it is therefore necessary for those packets to be classified into the PQ, but RSVP data flow packets should not be.

RSVP uses a profile to determine if a flow of packets is a voice flow or not. This traffic-specification profile takes, among other parameters, packet sizes and arrival rates into account, and if a packet flow conforms to the parameters, it is considered a voice flow. If not, it is considered a nonvoice flow, including both data and video. The internal profile is tuned such that all voice traffic originating by a Cisco IOS GW will fall within the parameters and will therefore automatically be considered a voice flow without needing extra configuration. For third-party applications such as NetMeeting, the profile may have to be tuned to pick up that kind of traffic.

RSVP is the first egress interface classifier to examine an arriving packet. If RSVP considers this a voice flow, the packets will be put into the PQ portion of low latency queuing (LLQ). If the flow does not conform to the voice profile but is nevertheless an RSVP reserved flow, it will be placed into the normal RSVP reserved queues. If the flow is neither a voice flow nor a "data " RSVP flow, the other egress interface classifiers (such as ACLs and match statements within a class map) will attempt to classify the packet for queuing.



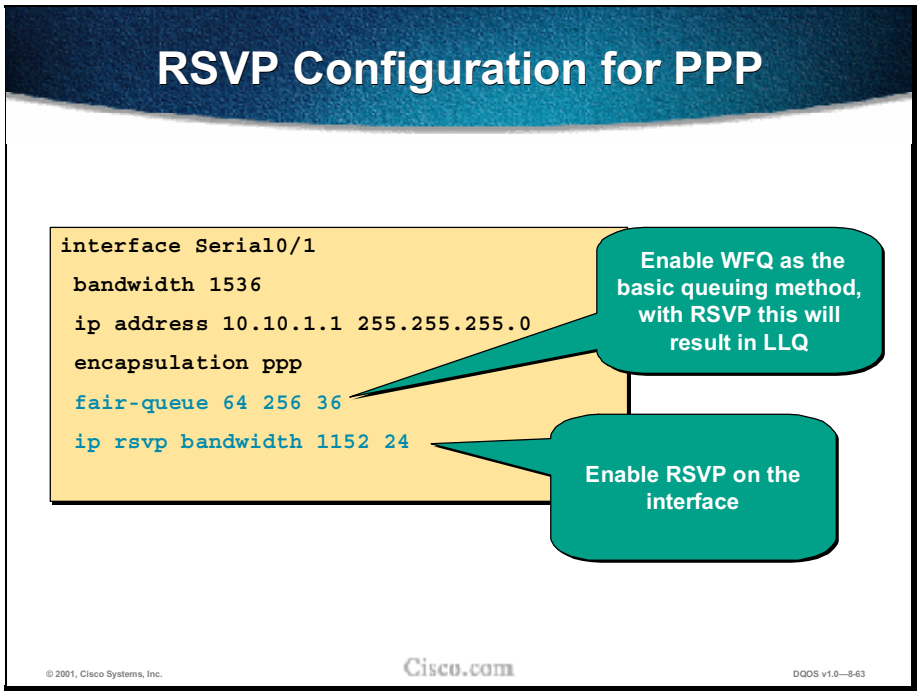
There are three things to configure on a GW that will originate or terminate voice traffic using RSVP:

- The synchronization feature between RSVP and H.323 must be turned on. This is a global command and is turned on by default when IOS 12.1.(5)T or later is loaded.

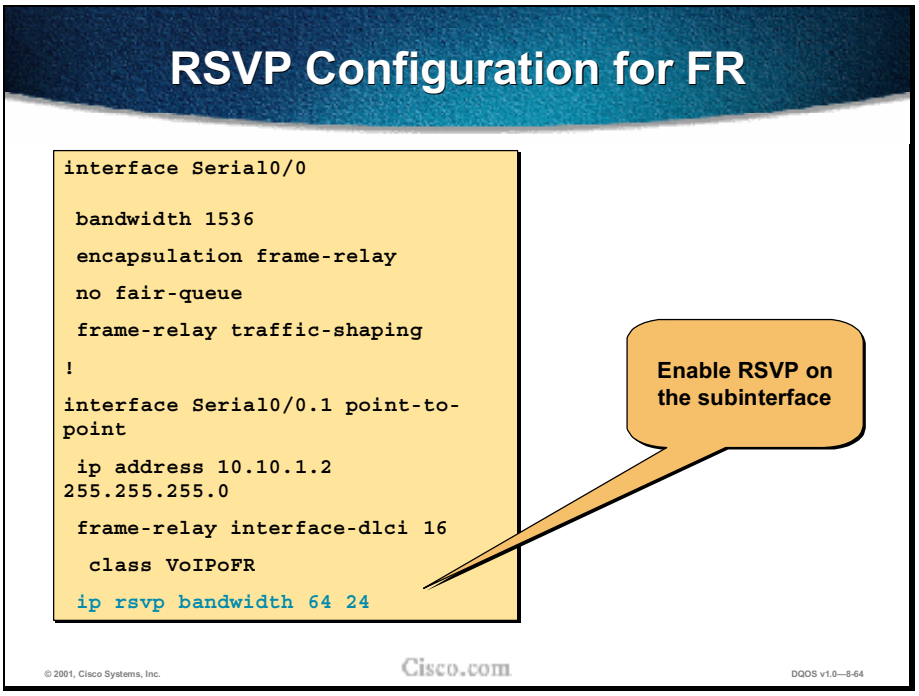
- RSVP on the VoIP dial peers, both originating and terminating sides of the dial peers. Configure both the requested QoS (**req-qos**), and the acceptable QoS (**acc-qos**) to guaranteed-delay for RSVP to act as a CAC mechanism. Other combinations of parameters may lead to a reservation, but no CAC. Please consult the 12.1.(5)T IOS documentation for details.

- RSVP must be enabled, and maximum bandwidth specified, on the interfaces that the call will traverse.

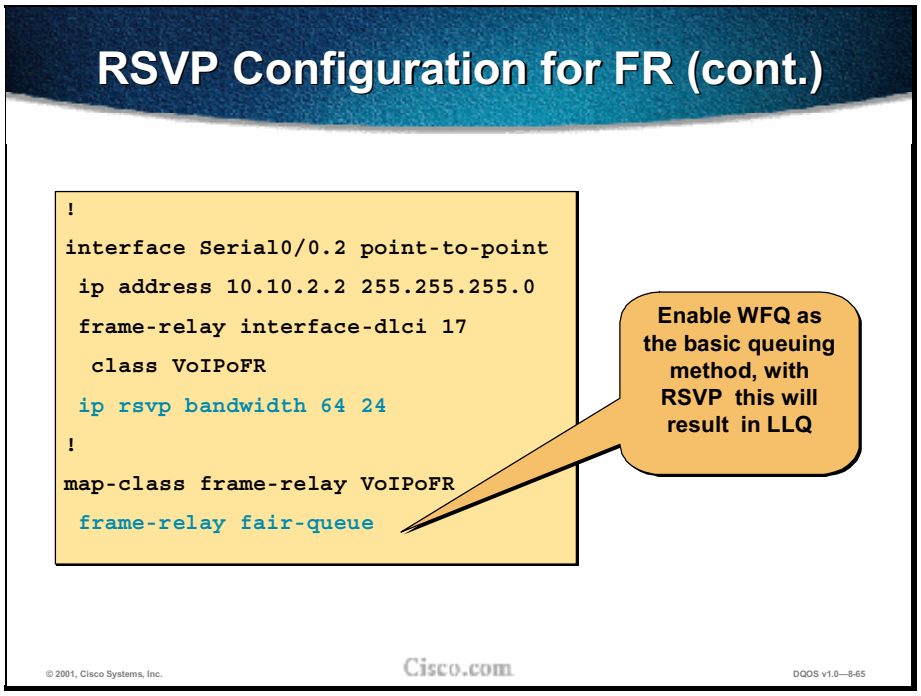
It is important to note that RSVP will classify only voice bearer traffic, not signaling traffic. One of the other classification mechanisms (for example, ACLs and DSCPs) must still be used to classify the voice signaling traffic if any treatment better than best effort is desired for that traffic. If left up to RSVP alone, signaling traffic will be considered best-effort traffic as shown in the figure above.



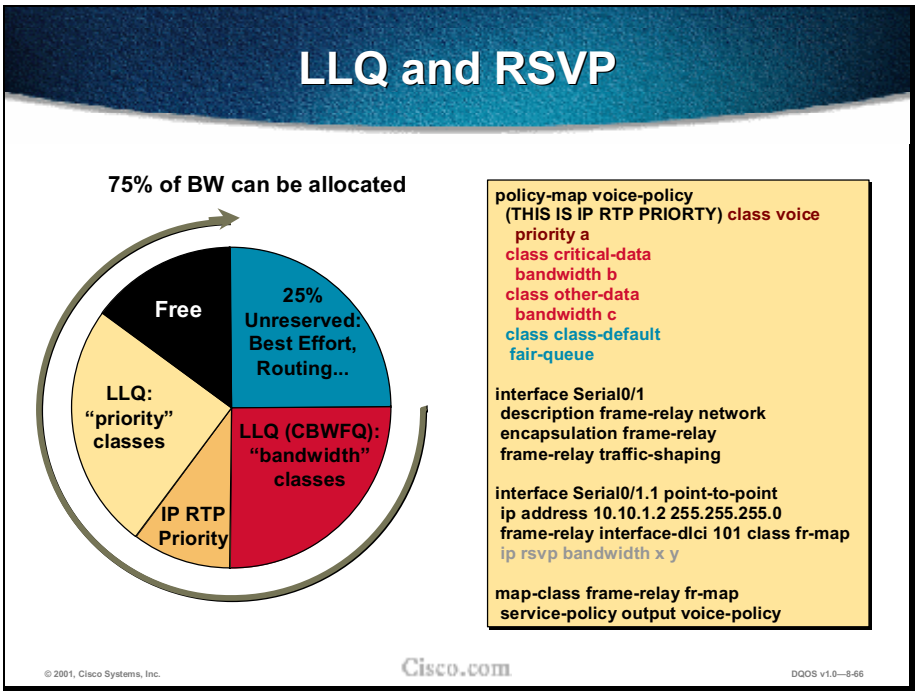
This sample configuration shows how to enable RSVP CAC when using PPP on serial interfaces. Note that WFQ is specified, but when used with RSVP, results in LLQ.



This sample configuration (continued in the next figure) shows how to enable RSVP CAC when using Frame Relay on serial interfaces. Note that WFQ is specified, but when used with RSVP, it results in LLQ.



This figure continues from the previous figure.



In Chapter 4, Congestion Management, we saw how to direct voice to a PQ and other traffic to the WFQ/CBWFQ. This was achieved using either IP RTP Priority or the **priority** keyword within a class map. RSVP provides a third method for achieving this, as illustrated above for FR or PPP links.

LLQ and RSVP (cont.)

Traffic is directed to the PQ portion of LLQ (PQ/CBWFQ) by any of these configurations:

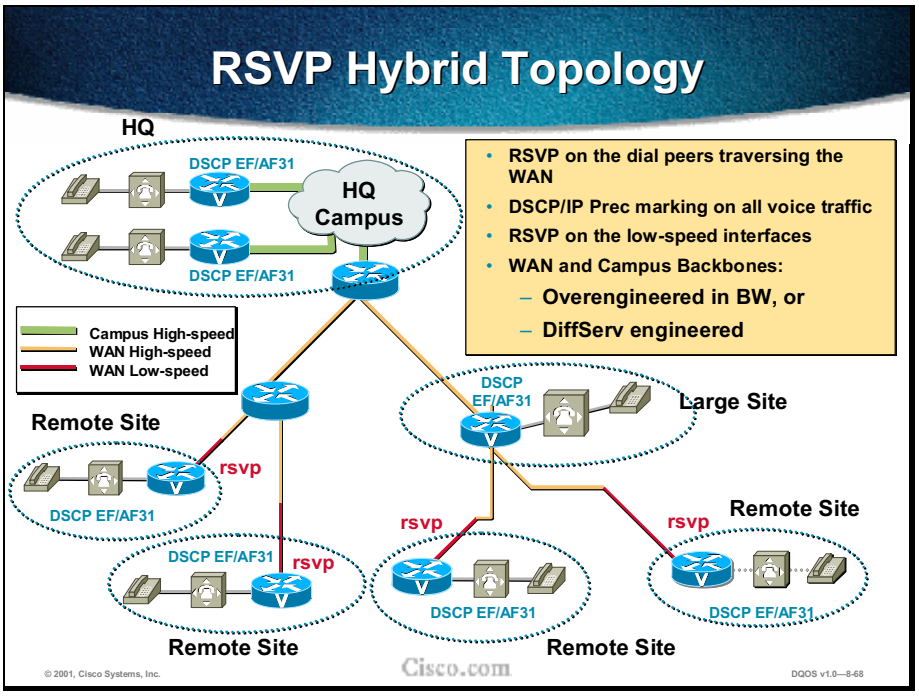
- **WFQ + IP RTP Priority**
- **WFQ + "priority" keyword within a class-map**
- **WFQ + RSVP**

PPP Interface

```
interface Serial0/1
 bandwidth 1536
 ip address 10.10.1.1 255.255.255.0
 encapsulation ppp
 fair-queue 64 256 36
 ip rsvp bandwidth x y
```

FR PVC

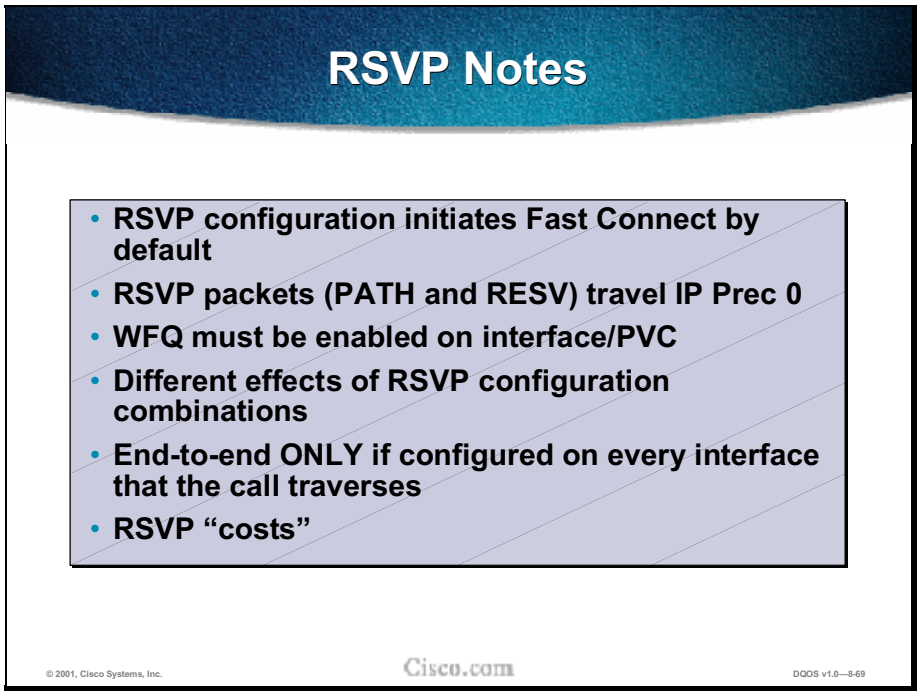
```
interface Serial0/1
 description frame-relay network
 encapsulation frame-relay
 no fair-queue
 frame-relay traffic-shaping

interface Serial0/1.1 point-to-point
 frame-relay interface-dlci 101 class fr-map
 ip rsvp bandwidth x y

map-class frame-relay fr-map
 service-policy output voice-policy
 frame-relay fair-queue
```



Concern is often expressed about RSVP scalability in terms of a large number of individual flow reservations that may be necessary across high-speed backbone links where many voice calls have aggregated. Indeed, it may not make sense to do individual flow management over, say OC-12, backbone network links. For this reason, IOS 12.1.(5)T code, and later, incorporates another change: If RSVP is not configured on the platform at all (any interface on the platform), RSVP messages will be passed through transparently. No reservation will be made or managed, but the PATH and RESV packets will no longer be dropped.

This makes it possible to build hybrid topologies where RSVP is used around the edges of the network to protect slower WAN access links from oversubscription, but the high-speed campus and WAN backbone links do not use RSVP. Of course, this topology compromises the true end-to-end reservation and guaranteed QoS promise of RSVP, but it may be a workable compromise. The backbone links can receive a measure of protection from either overengineering or one of the other CAC mechanisms discussed earlier, while the highest contention links (typically the WAN edge) can make use of RSVP.

The figure above shows such a hypothetical network that is configured for DiffServ in the backbone and campus, but uses RSVP reservations across the WAN edge links.



In using RSVP, several aspects of the protocol should be remembered.

In 12.1.(5)T, calls are initiated using FastConnect by default when RSVP is configured. These RSVP packets (PATH and RESV) travel IP Prec 0. WFQ must be enabled on interface/PVC.

There are different effects of RSVP configuration combinations:

- Enabled on platform, enabled on int/PVC: Reservation made
- Enabled on at least one int/PVC: Packets dropped
- Disabled on platform: Packets passed through

RSVP is end-to-end *only* if configured on every interface that the call traverses.

The use of RSVP "costs" in:

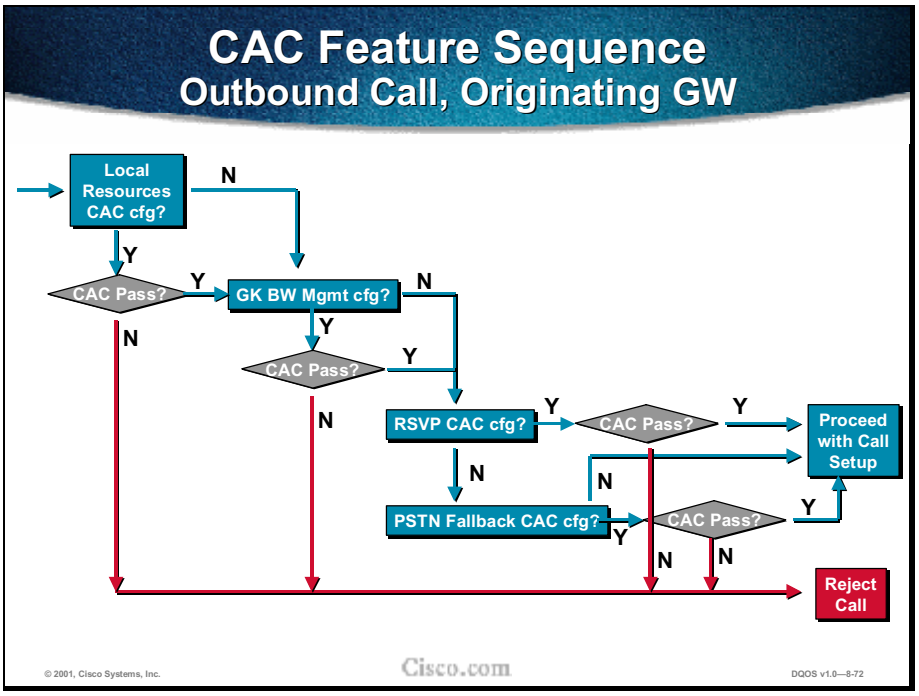
- Signaling (messaging and processing)
- Per-flow state (memory)
- Postdial delays

RSVP: Attributes

VoX Supported	VoIP/H.323 only
Trunking/IP Telephony	Trunking today RSVP for IP telephony is under development
Platform/Release	IOS GWs 12.1.5T Planned for CM 3.1
PBX Trunk Types Supported	All
End-to-end/Local/IP Cloud	E2E between OGW and TGW (provided all intermediate nodes are RSVP configured) Could be used at WAN edge with DiffServ backbone
Per call/interface/endpoint	Per Call
Messaging Network Overhead	PATH/RESV & periodic keepalives
Post-dial Delay	Yes

Topology Awareness: Yes Guarantees QoS for duration of call: Yes
Protect against changes in network conditions: Yes


Feature Combinations,
Interactions and Sequencing
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—8-71

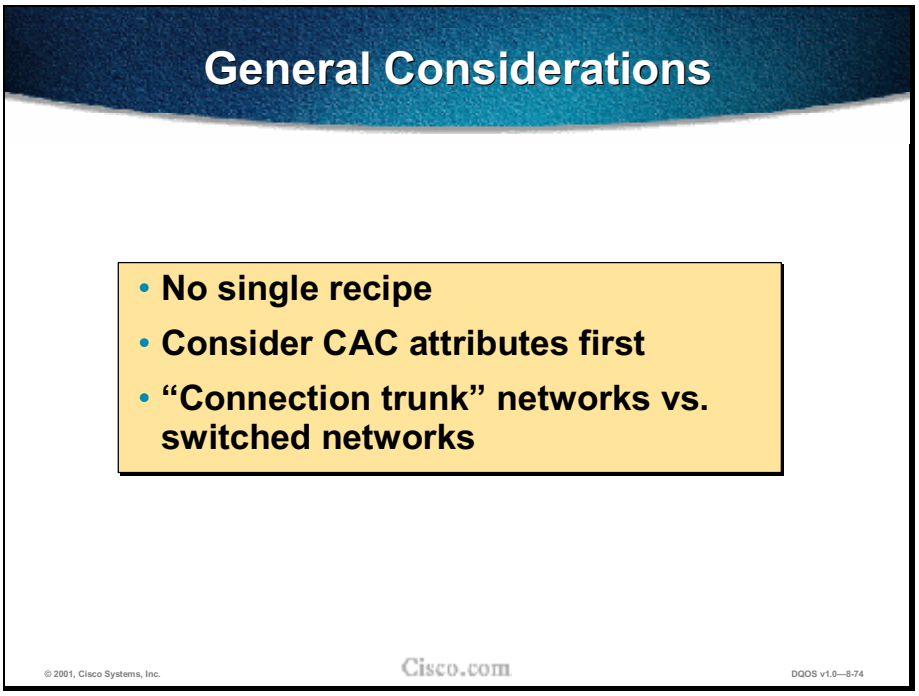


In this chapter, 10 different CAC mechanisms have been taught. While there is overlap in the functionality that some of the CAC mechanisms provide, several CAC mechanisms solve different aspects of the CAC problem and therefore would make sense to use together in a network design.

The figure above summarizes the sequencing of CAC features that can be active on an OGW.

The information supporting this figure is for 12.1.(5)T, and as features and software releases change (and bugs are fixed), this information may change without notice. As can be seen from the flow diagram above, the only features that are mutually exclusive are RSVP and PSTN fallback.


When to Use What
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—8-73

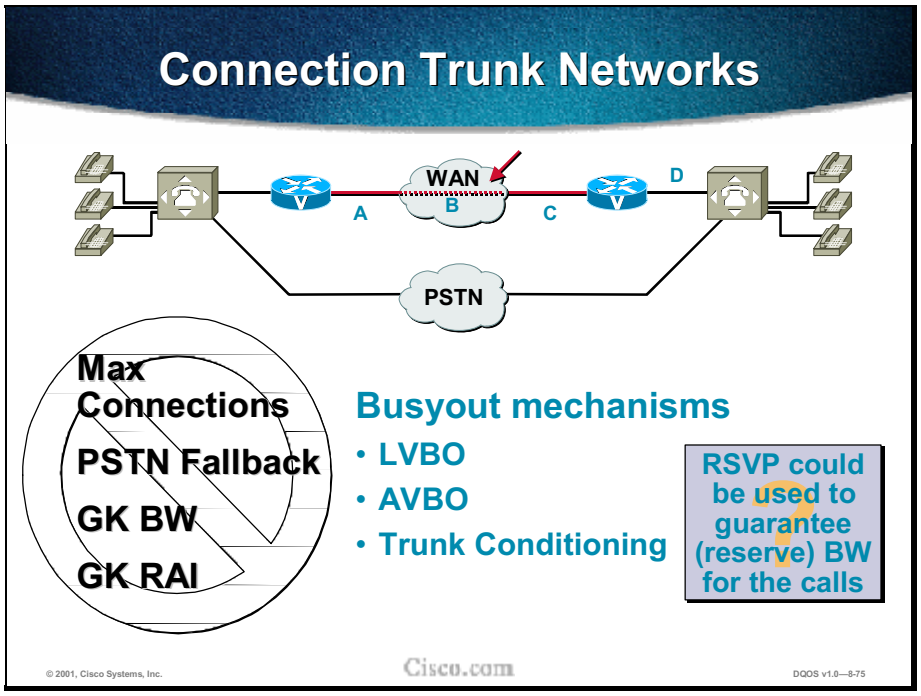


With a plethora of CAC tools available in IOS software, the immediate design question is "When should I use what?" The various features often do different things and solve different aspects of the CAC problem. Some of these aspects may be more important design criteria for the network than others. There is no single recipe of exactly when to use what—like all other software features, the decision has to be made while considering the network design goals.

The first-level consideration should be governed by the attributes of the CAC mechanism. For instance, if an SIP-based VoIP network is being designed, there is no point in considering an H.323 CAC feature.

Different CAC mechanisms are appropriate depending on the kind of network that is being configured. Connection-trunk networks consist of nailed-up connections across the packet network. The PBX may perceive that it makes each call individually, but the packet network has a permanent trunk in place, in concept similar to a leased line, that is always present, always ready, and always terminates to a fixed and predetermined destination (that is, point-to-point link). These nailed-up packet network configurations are typically used when some signaling is present between the PBXs that must pass transparently and unchanged through the packet network, that is, the GWs cannot interpret the signaling and merely tunnels it through the packet network.

In switched networks where each call is set up individually across the packet network after the user dials, CAC methods are most useful and most needed. It is often impossible to predict exactly how many calls might want to use a particular network leg at a given point in time.

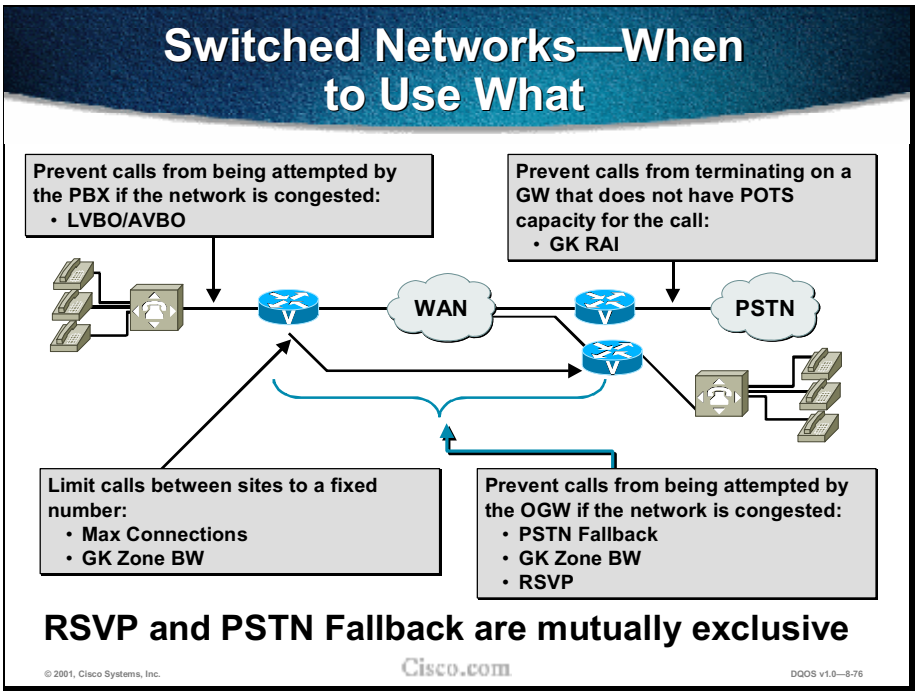


IOS GW connection-trunk configurations use the same basic tools as switched networks to set up connections (dial peers, etc.). The difference is that these "calls" are set up only once, when the GW boots up or when the configuration is inserted, and remain in place indefinitely. If a link in the network should fail and bring the "call" down, the router will reestablish it at its earliest opportunity. Whether or not there is actually a real call (live people talking) active over this connection is transparent to the GWs. For this reason the standard CAC mechanisms, in most cases, do not apply. Connection-trunk configurations will not come up properly if there is not enough bandwidth for the connections, so once the configuration is in place, it is largely given that there is sufficient bandwidth available for the calls.

These call-by-call CAC mechanisms would apply *only* to switched networks and should *not* be used (they can be configured, but will have little to no effect) with connection-trunk configurations: max connections, PSTN fallback, GK bandwidth, or GK RAI.

Connection-trunk configurations can, however, benefit from the PBX busyout CAC features. When something in the network is down and the nailed-up connections fail or the interface they use fail, it would certainly be useful to busy-out the trunk to the PBX. These features include LVBO, AVBO, and trunk conditioning

In concept, RSVP could be used to guarantee (reserve) bandwidth for the nailed-up calls to protect the voice quality from fluctuating network conditions. However, connection-trunk networks are fixed, point-to-point connections, and therefore the number of calls (from the router's perspective) active across any network segment is fixed and relatively easily designed by manually engineering the bandwidth and using standard LLQ configurations to ensure bandwidth. The value-add RSVP can bring to the table here should be carefully considered.



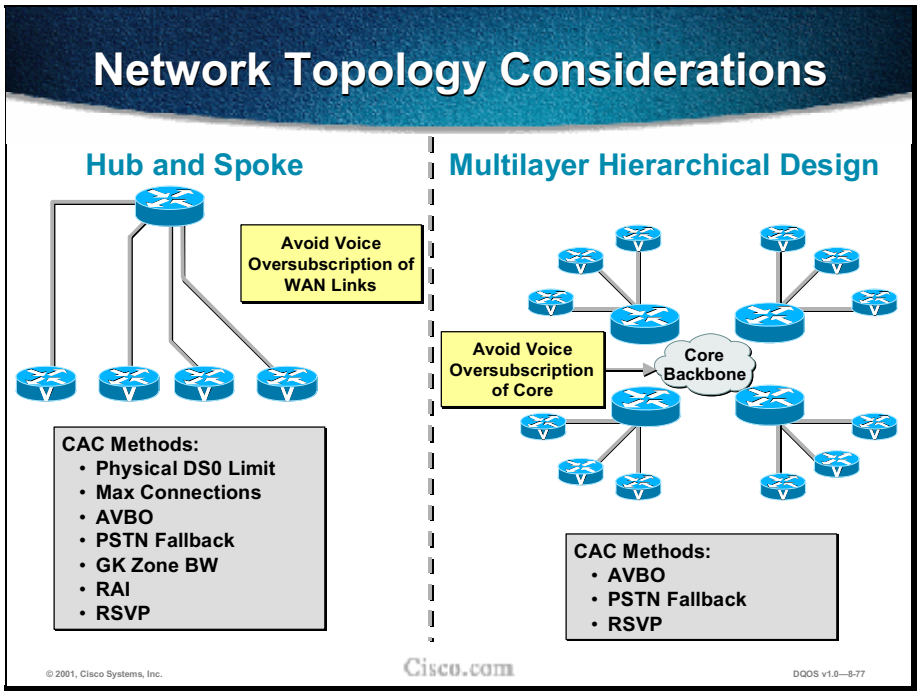
The area labeled “A” is the originating POTS connection. If it is important to keep the originating PBX from attempting to place a call onto the packet network when the network is incapable of completing the call, the busyout CAC features should be considered. These include LVBO and AVBO. This may be important if hairpinning is an unacceptable call reject recovery method, or if the PBX/Key-System does not have the ability to choose another route for a rejected or hairpinned call.

Area B is the terminating POTS side of the connection. If it is important in the network, and likely because of specific traffic patterns, that the terminating POTS side may be the part of network most susceptible to oversubscription, GK RAI should be used.

Area C is the cloud of the network, the IP backbone. This is the most typical area of the packet network that enterprise customers wish to protect their calls against, because this infrastructure is not dedicated to voice, but is shared by many types of traffic. The CAC features protecting the cloud include PSTN fallback, GK zone bandwidth, and RSVP.

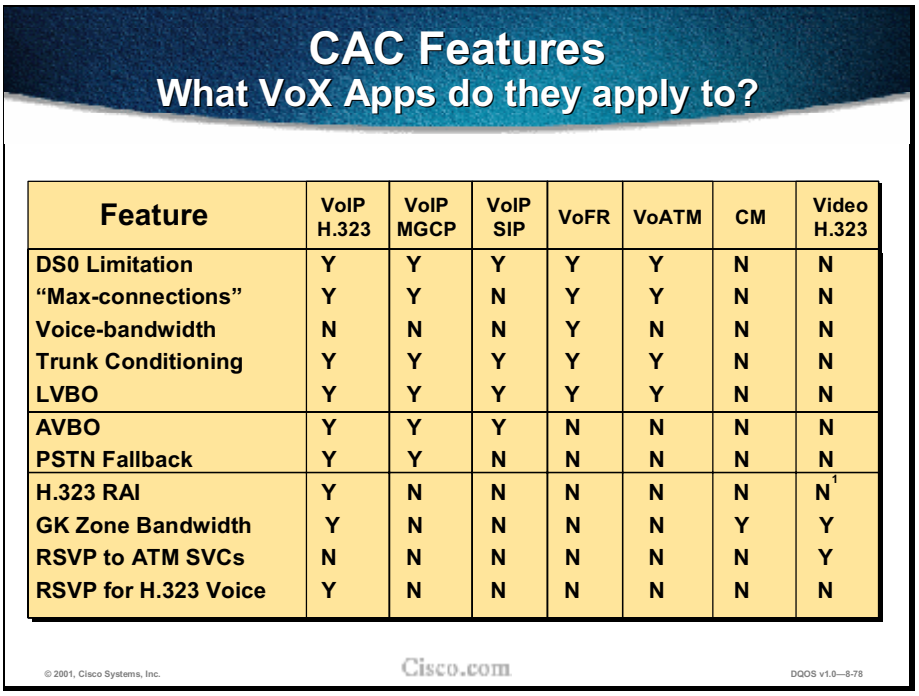
The above methods are all IP-based methods, which means implicitly that there are more CAC methods available for VoIP networks than for VoFR and VoATM networks. VoIP also needs it more, because the Layer 2 technologies like FR and ATM cannot intrinsically protect against VoIP packet loss, as they can with VoFR and VoATM traffic.

Area D is a logical section of the network, which is the piece of the network between sites. Regardless of actual infrastructure is connecting sites together, there may be the desire not to limit traffic within a site (or at the very least to limit it based on very different criteria) than traffic between sites. For example, if the headquarters location has the capability to handle 24 active calls at once, you may want to make sure that all 24 calls cannot be used by a single remote site at any one time, but make sure that there is a certain amount of capacity available to different remote sites, so the low-traffic sites do not get locked out by the high traffic sites.



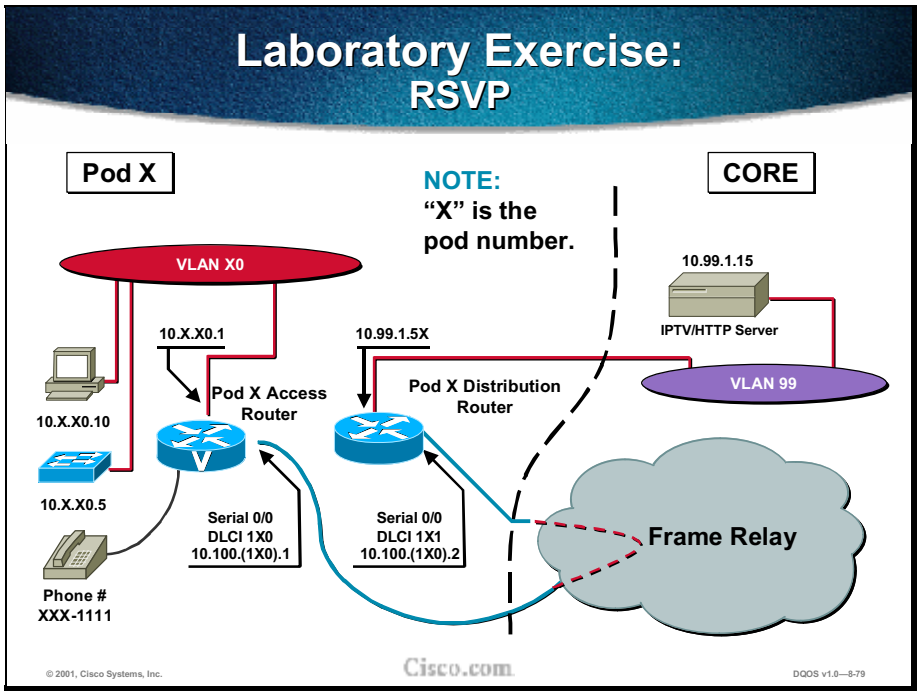
The hub-and-spoke network is, of course, the easiest to take care of. In this case most of the CAC features are useful since it is only the spokes of the network that need protection: There is no invisible backbone and the spoke links may well be the very links connected to the GWs at the remote sites. Almost any of the CAC features can be used to good effect in this type of network: physical DS0 limit, max connections, AVBO, PSTN fallback, GK zone bandwidth, RAI, and RSVP.

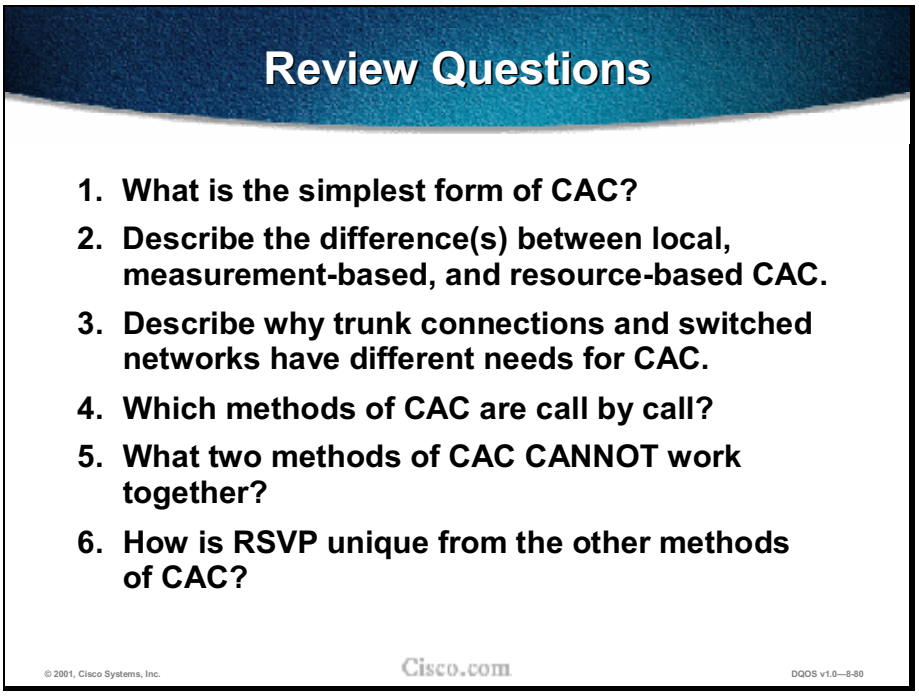
The multilayer hierarchical network is more representative of larger networks where outlying sites aggregate at intermediate points, perhaps several layers of these, before a core network that connects the highest layer aggregation sites. Many of the CAC features will protect the WAN link at the lowest layer of the network, but few of them have visibility into the aggregation and core legs of the network. The ones that do include AVBO, PSTN fallback, and RSVP.



The table above summarizes the information about which CAC feature can be used with which voice application.

Note that the H.323 RAI capability does in concept apply to H.323 video applications. However, it is listed here as No since the GWs under consideration in this chapter are Cisco IOS Voice GWs, and these will not generate RAI indications for video traffic.


Laboratory Exercise:
RSVP
Pod X
CORE
NOTE:
"X" is the
pod number.
VLAN X0
10.99.1.15
IPTV/HTTP Server
10.X.X0.1
10.99.1.5X
Pod X Access
Router
Pod X Distribution
Router
VLAN 99
10.X.X0.10
10.X.X0.5
Serial 0/0
DLCI 1X0
10.100.(1X0).1
Serial 0/0
DLCI 1X1
10.100.(1X0).2
Frame Relay
Phone #
XXX-1111
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—8-79

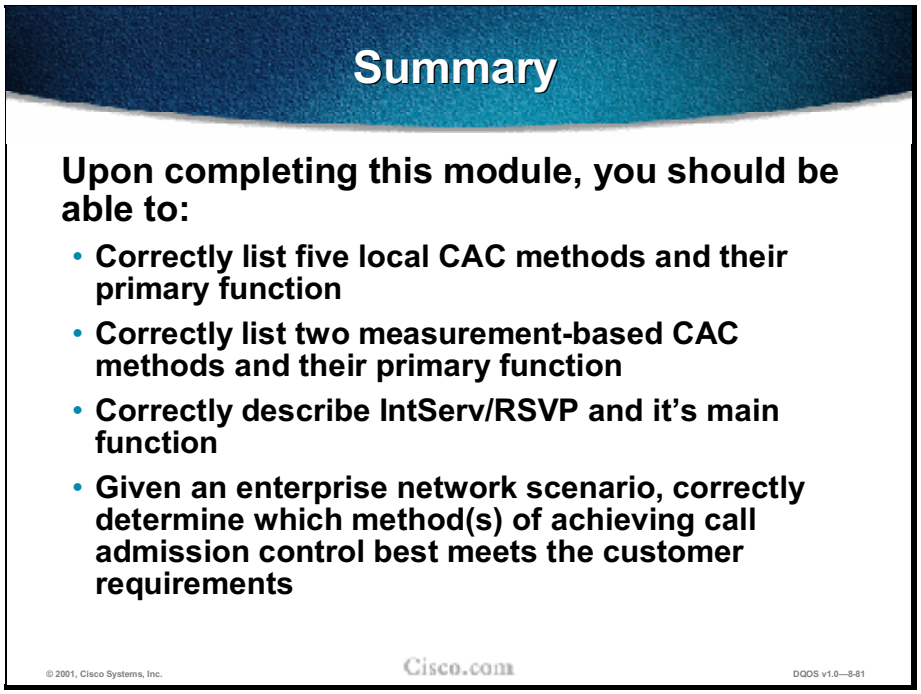


Answer these questions:

1. What is the simplest form of CAC?
2. Describe the difference(s) between local, measurement-based, and resource-based CAC.
3. Why don't call-by-call mechanisms work for connection-trunk networks?
4. Which methods of CAC are call by call?
5. What two methods of CAC *cannot* work together?
6. How is RSVP unique from the other methods of CAC?

Answers to the review questions appear in Appendix B.

Summary


Summary
Upon completing this module, you should be able to:
• Correctly list five local CAC methods and their primary function
• Correctly list two measurement-based CAC methods and their primary function
• Correctly describe IntServ/RSVP and it's main function
• Given an enterprise network scenario, correctly determine which method(s) of achieving call admission control best meets the customer requirements
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—8-81

9

Management Tools

Overview

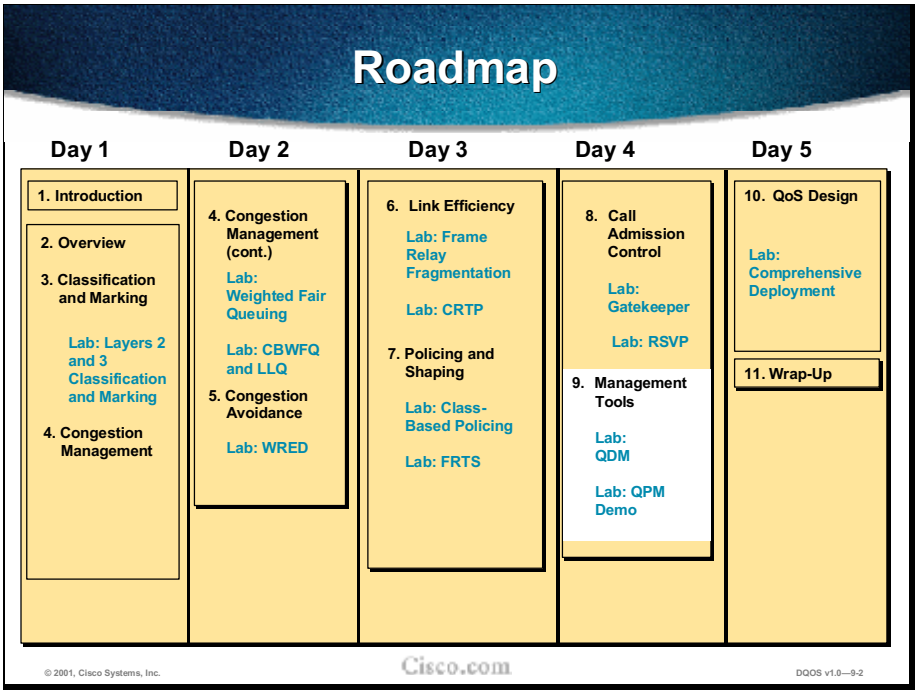
This chapter covers the tools that are available to network administrators to provision and monitor quality of service on the converged network. You will learn how QoS Device Manager and QoS Policy Manager are used. You will configure Cisco Service Assurance Agent to measure SLA metrics and monitor local and remote devices. You will know how to use IPM (Internet Performance Manager) and SMS (Service Management Solution) to monitor and troubleshoot network performance.

Objectives

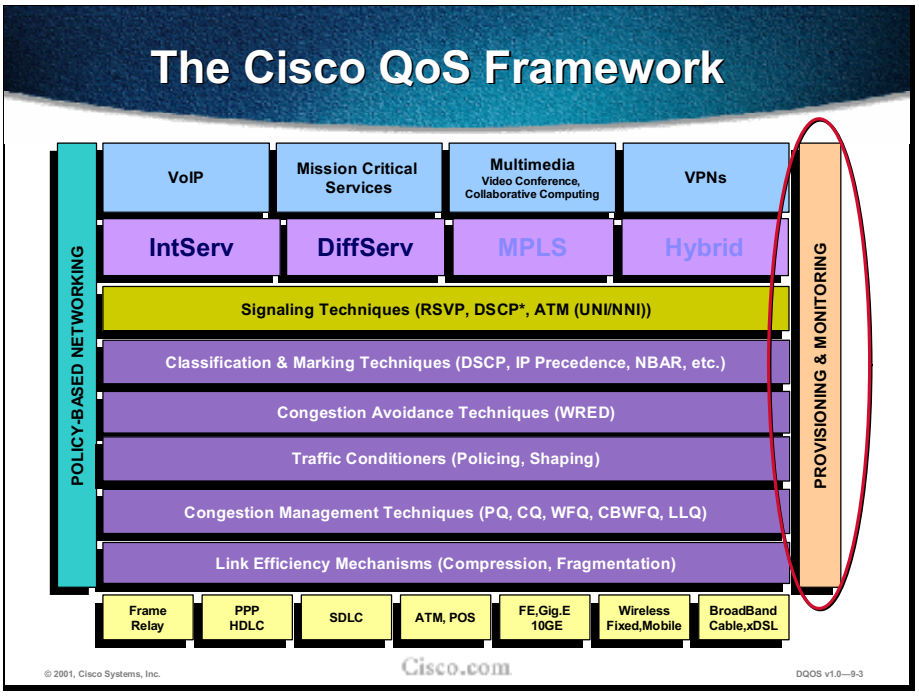
Upon completing this chapter, you will be able to:

- Utilize QoS Device Manager to monitor performance, establish baselines and configure QoS policies
- Utilize QoS Policy Manager to configure advanced QoS policies, scale policy deployment, upload/verify/rollback policies, and deploy QoS policies by external time-based/event-based scripts
- Configure Cisco Service Assurance Agent to measure key SLA metrics and monitor network performance between local and remote devices
- Monitor and troubleshoot network performance with IPM and SMS

Outline

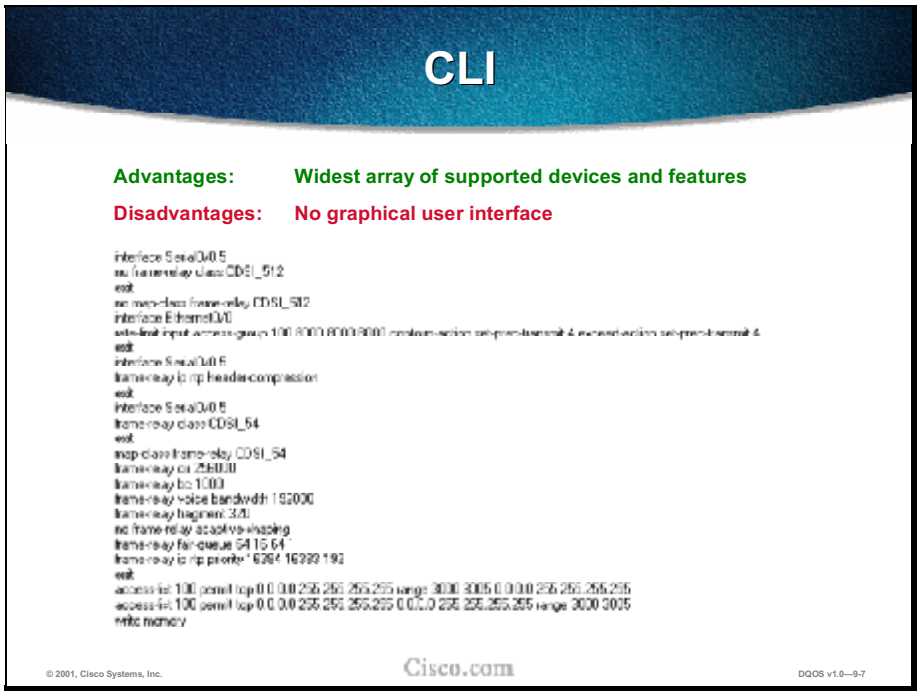


The figure shows the plan for the week.

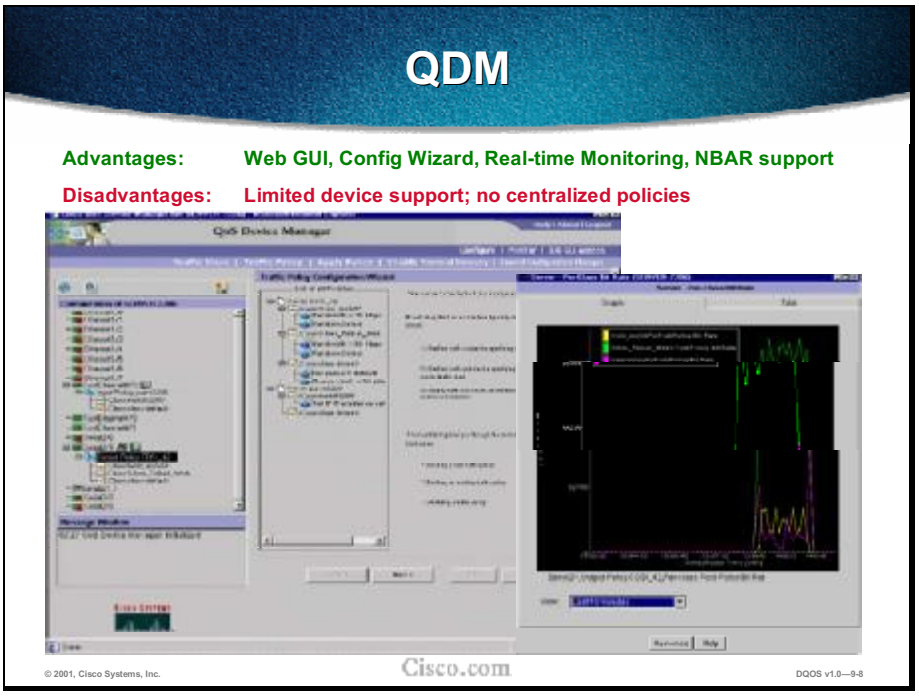


Agenda

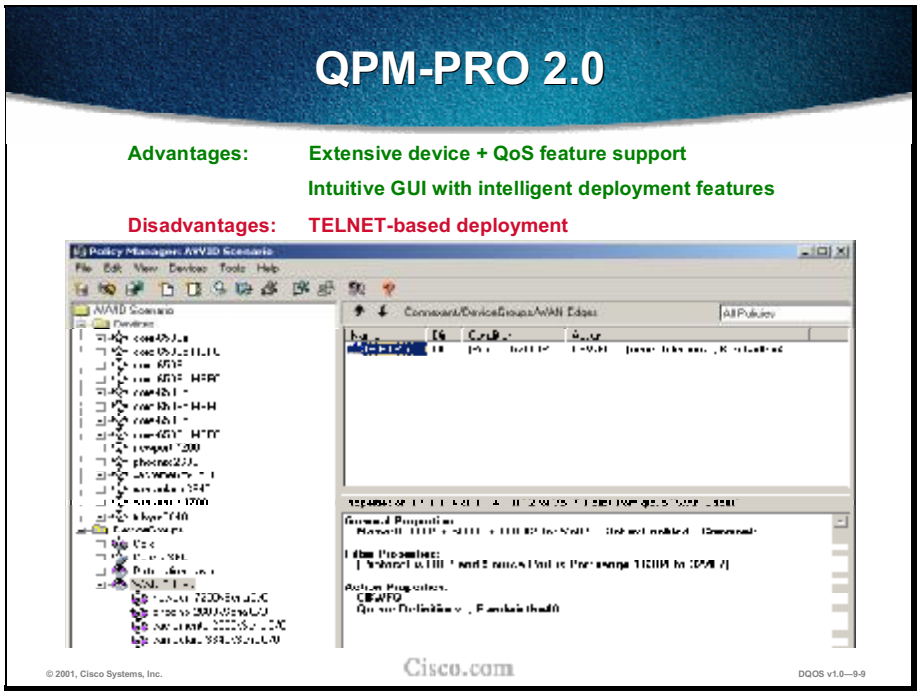
- Overview
- Quality Device Manager (QDM)
- Quality Policy Manager (QPM)
- Service Assurance Agent (SAA)
- Internetwork Performance Monitor (IPM) and Service Manager Solution (SMS)



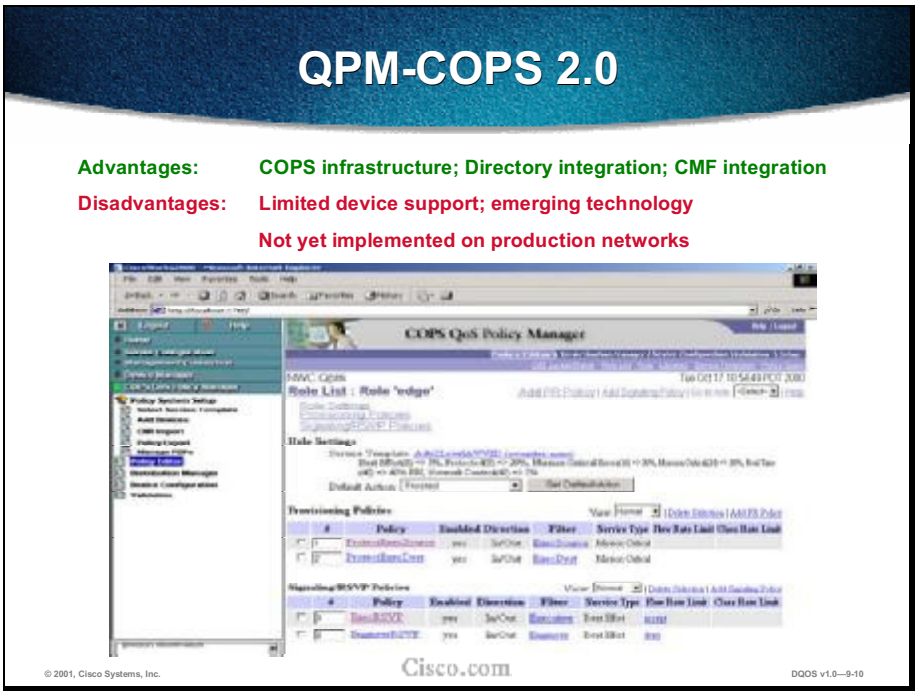
The CLI (command-line interface) is the backbone for the tools taught in the chapter. Its advantage is that it supports the widest array of devices and features. However, CLI has the disadvantage of not having a graphical user interface.



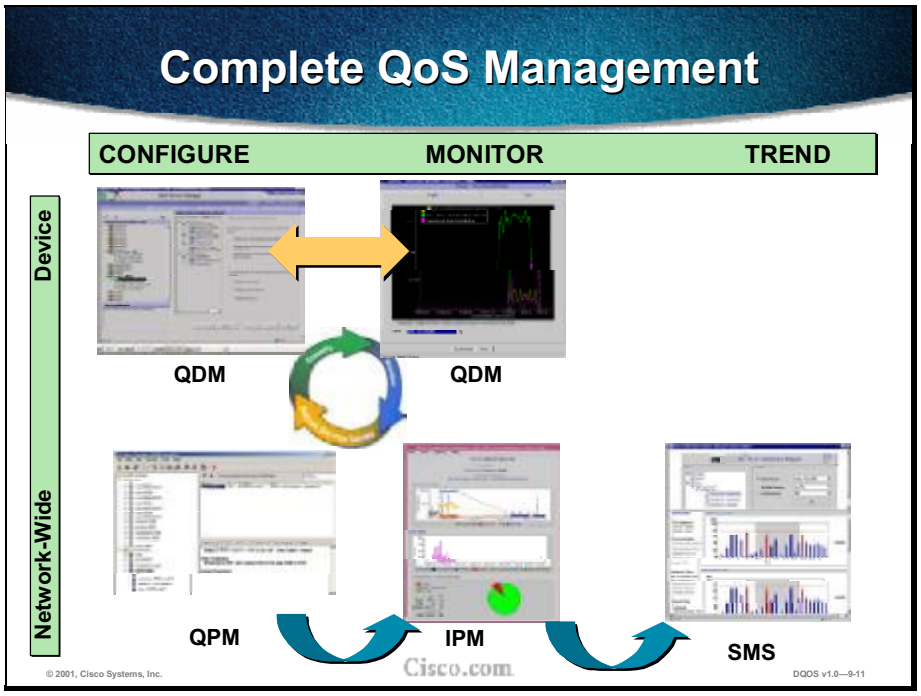
QoS Device Manager (QDM) has the advantage of a web interface with point-and-click commands, eliminating the keystroke errors possible with CLI. QDM has a configuration wizard to guide the user through the setup. It offers real-time monitoring and support for NBAR. QDM is limited in the number of devices that it supports. It does not support centralized quality policies for an entire network.



QPM-PRO 2.0 (QoS Policy Manager—Provisioned 2.0) provides provisioned QoS control using differentiated services to enforce QoS end-to-end. QPM-PRO's configuration interface allows you to define and deploy policies more easily than using device commands directly via the command-line interface. QPM-PRO supports a broad base of Cisco devices and QoS features. Its disadvantage is that it is TELNET-based.

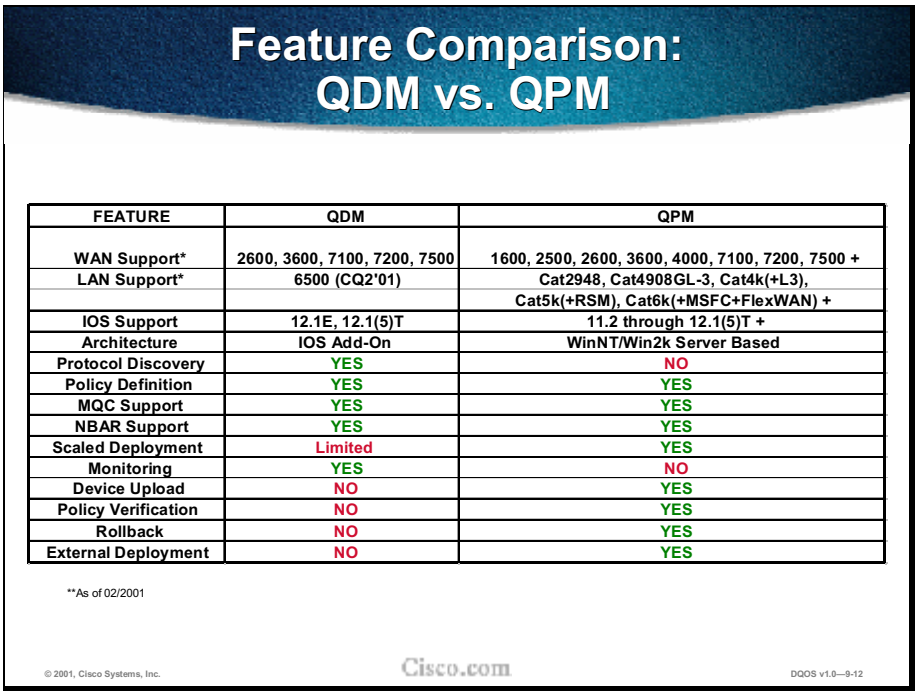


QPM-COPS (Common Open Policy Service) supports newly emerging, standards-based technology for both signaled and provisioned QoS using the industry-standard COPS protocol. QPM-COPS complements the provisioning provided by QPM-PRO by allowing the definition and classification of user-based policies stored in a standard LDAP directory. It is integrated with the Common Management Framework (CMF) in CiscoWorks2000. QPM-COPS is limited in the devices that are supported. It is an emerging technology that is not yet implemented on production networks.



Setting QoS policies alone is not sufficient for successful deployment of QoS: Network administrators must monitor network traffic before and after policies are set to ensure that the desired effects are indeed occurring. Furthermore, long-term trending is essential to ensure that as the enterprise grows and changes, service levels are still being met.

QDM (Quality of Service Device Manager) has the ability to configure and monitor policies on a single device. QDM is the obvious tool for experimenting with QoS and observing its impact on a small or experimental network. Once a network administrator is satisfied that the correct policies have been configured on a device, the network administrator may wish to scale deployment throughout the network enterprise. To accomplish this successfully, QPM (Quality of Service Policy Manager) is used. This is because QDM does not scale or allow for centralized policy management. Once policies have been set enterprise-wide, IPM (Internetwork Performance Monitor) can be used to monitor the immediate effects of the deployment, but SMS (Service Management Solution) is Cisco's recommended tool for long-term trending.



QDM and QPM are complementary products. Both products add value to network administrators with a knowledge of QoS to quickly and easily define QoS policies by using a graphical user interface (GUI) instead of a command-line interface (CLI). QDM performs QoS configuration and monitoring on one Cisco router at a time and thus provides a good starting point for implementing QoS in a network. QPM adds value when customers move toward enterprise-wide configuration of QoS policies. However, QDM is still valuable as a monitoring tool, regardless of whether the QoS policies have been set by QDM, QPM, or CLI.

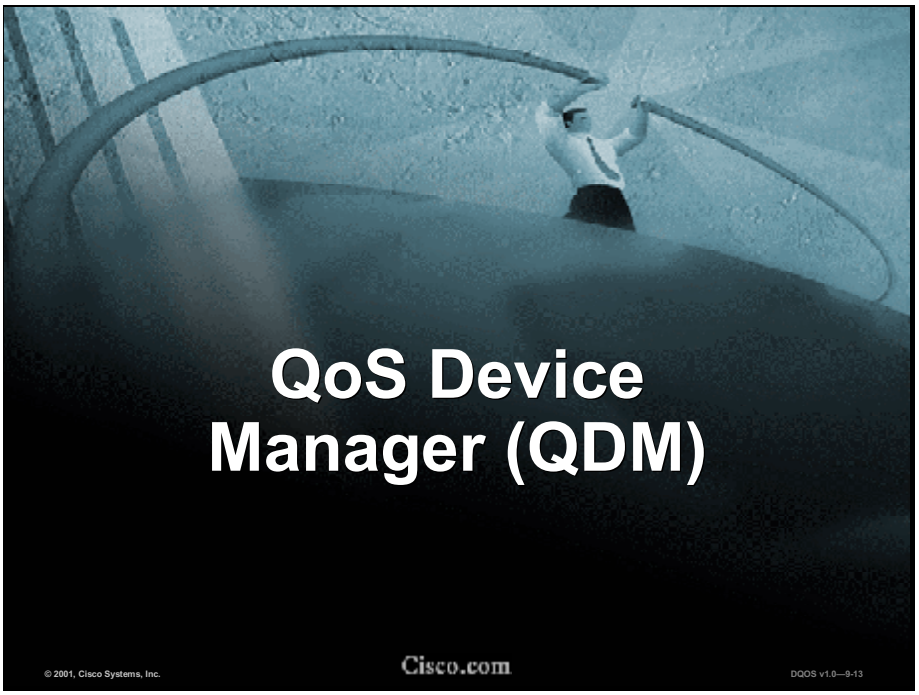
QPM is a network management application that supports enterprise-wide configuration of QoS functionality in Cisco devices based on user-defined policies. QPM provides network managers with the ability to set consistent and centralized QoS policies throughout their enterprise. There is much greater device and IOS support offered by QPM. Additionally, QPM offers several advantages.

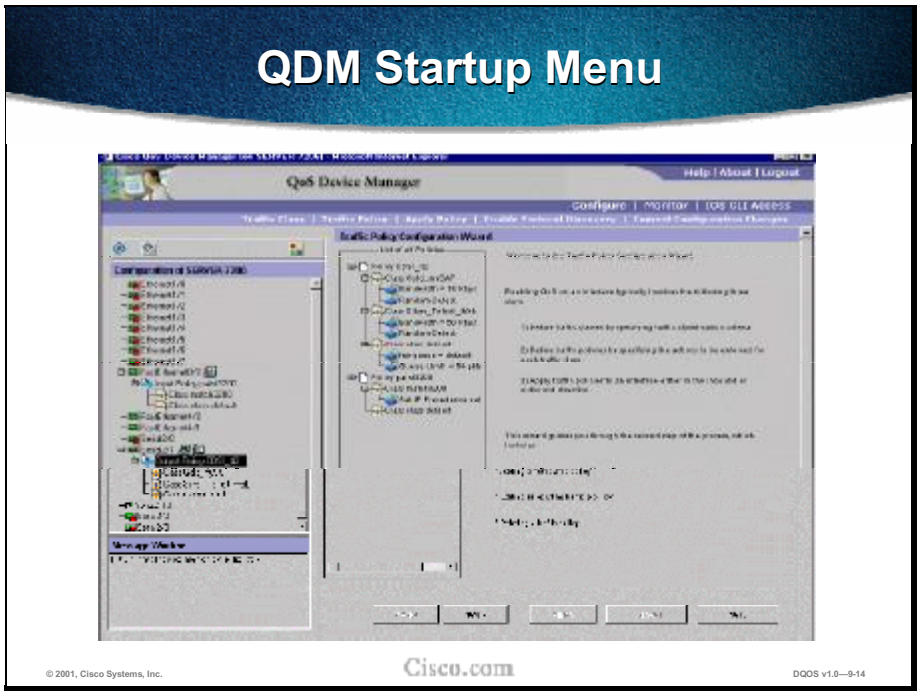
- Device upload— The ability to parse the configuration of a device being added to the QPM database for existing QoS policies, then representing these policies in a GUI format for ease of comprehension and management

- Device verification— The ability to verify whether a QoS policy that has been previously deployed is unaltered in the device's configuration (at times configuration changes have been made external to QPM that may affect policies that the network administrator believes to be intact)

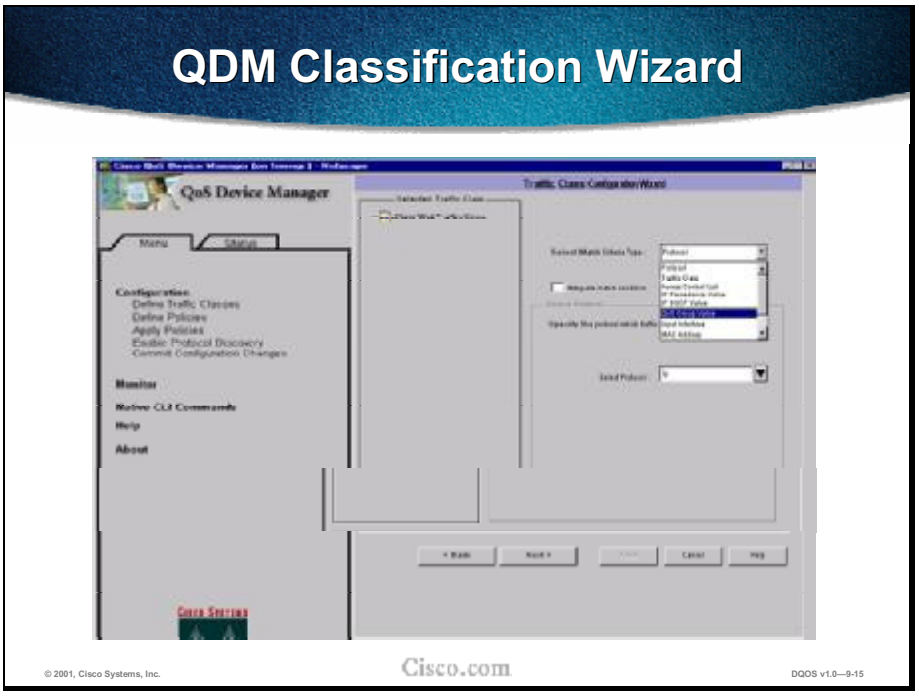
- Rollback— The ability to quickly and easily revert to a previous set of configurations in case an incorrect policy has been deployed or in case a particular set of policies is no longer required

- External **deployment**—The ability to push out QoS policies via time-based or event-based scripts





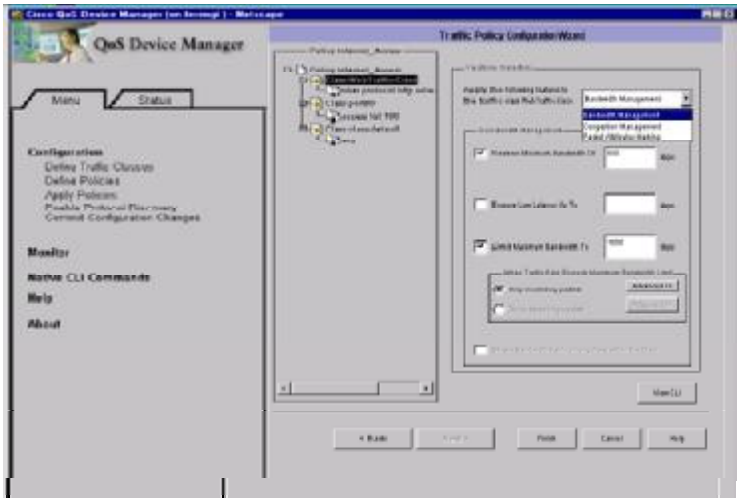
QDM is an IOS add-on that leverages the IOS feature IP HTTP-SERVER (which enables a router to function as a web server). By installing the QDM add-on software and images to the flash of the router, it becomes a web server that allows a client to set policies on it and monitor (real-time) the effects of the policies. This software runs on a Java-enabled web browser.



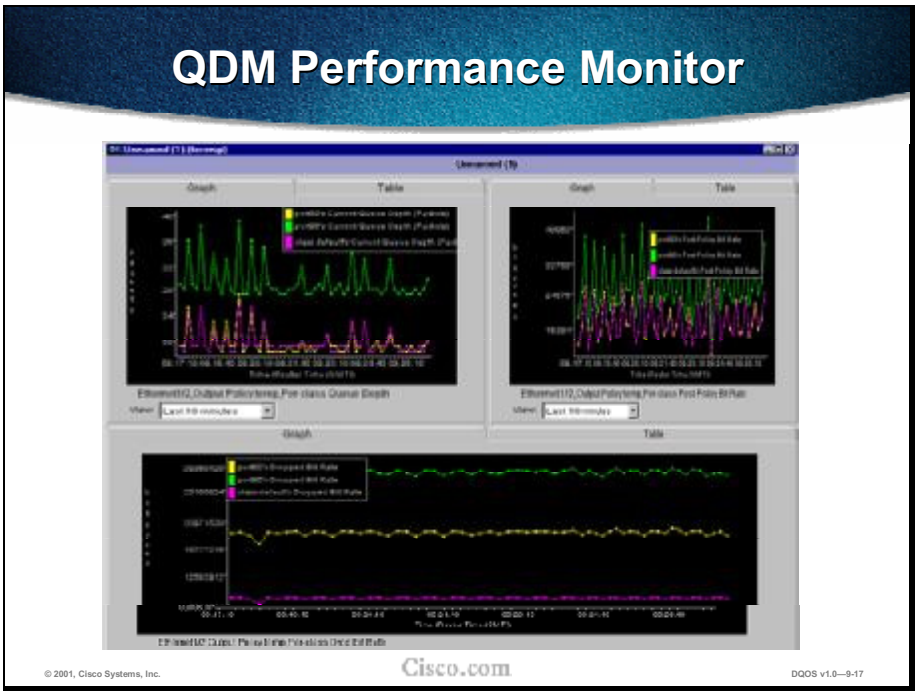
Here is a screen capture of the QDM configuration wizard for traffic classification.

The advantages of a GUI interface over the CLI are twofold. The GUI offers point-and-click for configuration, thus eliminating the possibility of keystroke error. In addition, the wizard provides options for each configuration, thus ensuring that the configuration is correct and complete.

QDM QoS Config Wizard



Here is a screen capture of the QDM configuration wizard for traffic policy.

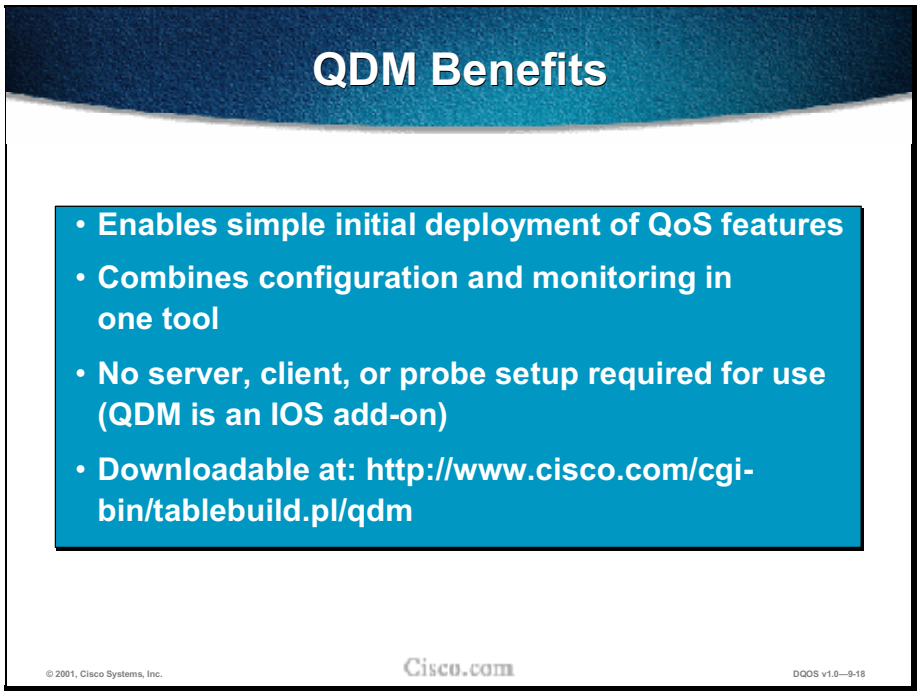


The screen capture in the slide shows some real-time performance graphs of various QoS metrics.

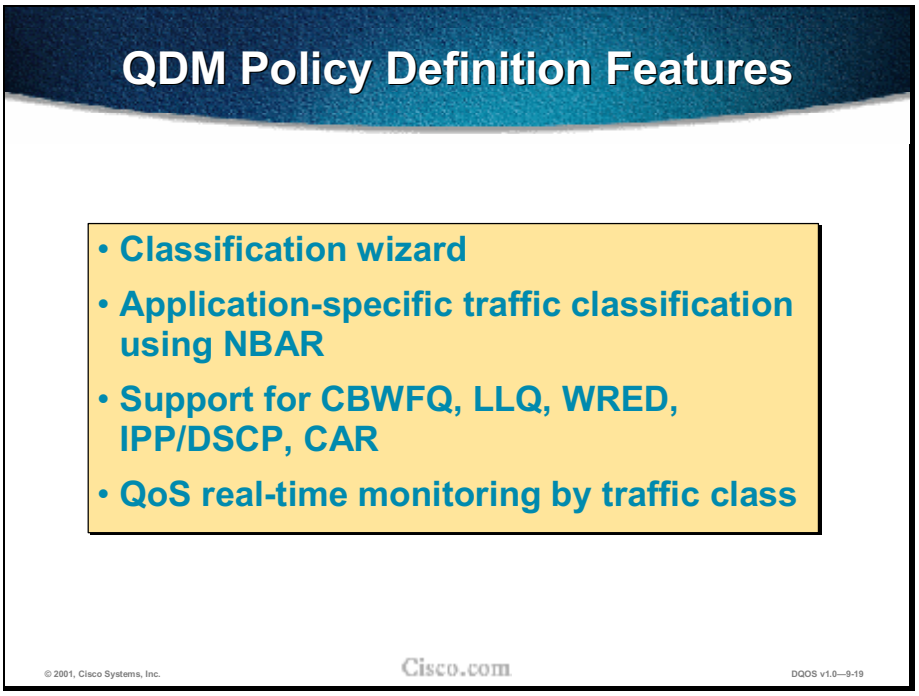
Clients can select which metrics they want QDM to graph, including:

- Queue depth
- Drop rate
- Pre/post policy packet count
- Pre/post policy byte count
- Pre/post policy bit rate

The traditional methods of viewing this data, HP Open View, SunNet Manager, Spectrum, and so forth, are more difficult to use and more expensive.



The benefit of the QoS Device Manager is that QDM provides a simple way to deploy QoS tools and monitor them. QDM is an IOS add-on that is downloadable from the web.



QDM allows the network administrator to analyze existing traffic patterns in a network. This is known as baselining.

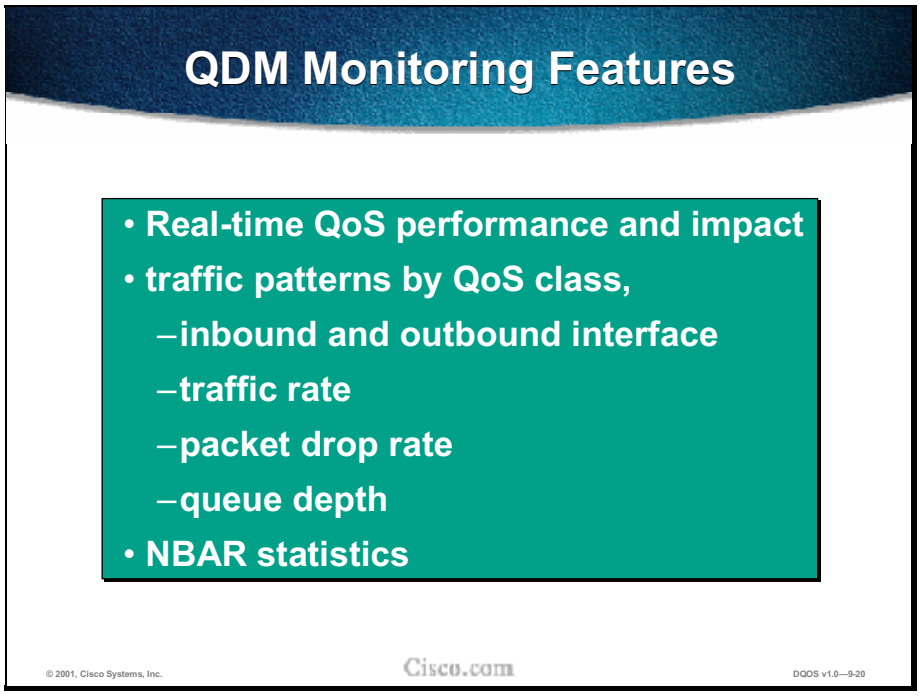
- Real-time reports show network traffic utilization by application type, based on statistics provided by NBAR protocol discovery feature within the Cisco 7100/ 7200/3600/2600 routers

Features:

- Simplified definition of QoS traffic classes using a classification wizard
- Traffic classification based on several criteria, including access control list (ACL) parameters (source/destination addr/port, IPP/DCSP, etc.) or by NBAR parameters
- Application-specific traffic classification using NBAR
- Packet coloring to mark packets for actions by downstream devices; supports packet marking (through QDM traffic classification) using standards-based marking schemes, including user-defined IP Precedence value (IP DSCP)

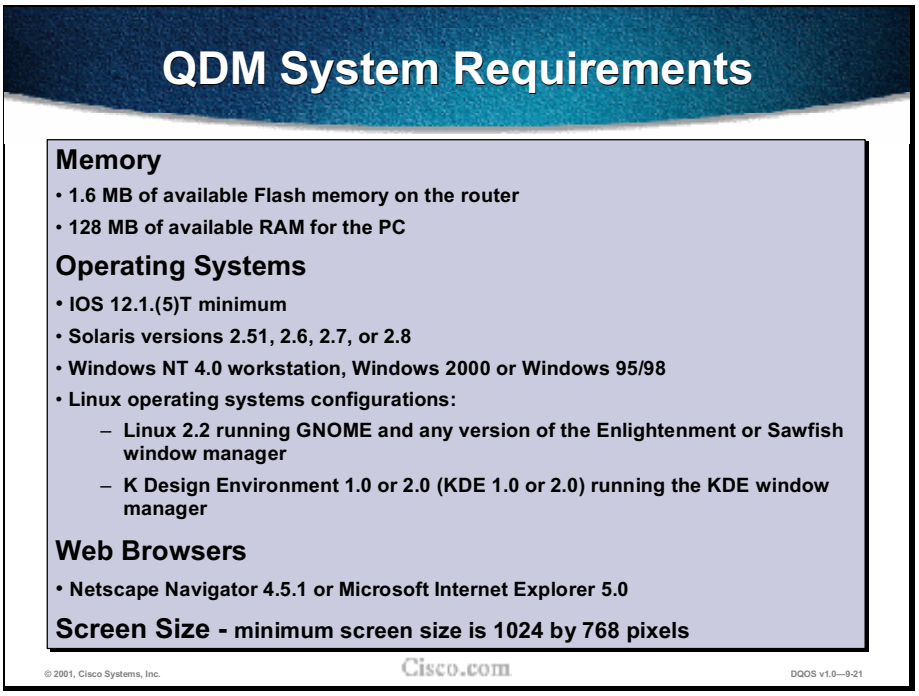
QDM provides support for the following QoS mechanisms:

- Minimum guaranteed bandwidth using class-based weighted fair queuing (CBWFQ), low latency queuing (LLQ), rate limiting using traffic policing, congestion control using traffic shaping, congestion avoidance using weighted random early detection (WRED), fair bandwidth allocation using flow-based weighted fair queuing (FBWFQ)



QDM provides monitoring of QoS mechanisms. With QDM you can:

- Monitor the real-time performance of QoS functionality through the router
- Observe the effect on traffic patterns of QoS mechanisms that have been configured
- Monitor by QoS class, on either an inbound interface or outbound interface, the traffic rate, packet drop rate, or queue depth
- Monitor NBAR statistics to observe the allocation of bandwidth achieved resulting from QoS configuration on the router



The basic system requirements for QDM 2.0 are listed in the figure above.

Additional Notes

Flash Memory

QDM 2.0 cannot be installed on a partitioned flash file system. If you have a partitioned flash file system, the **no partition flash** command can be used to eliminate the partitions.

Web Browsers

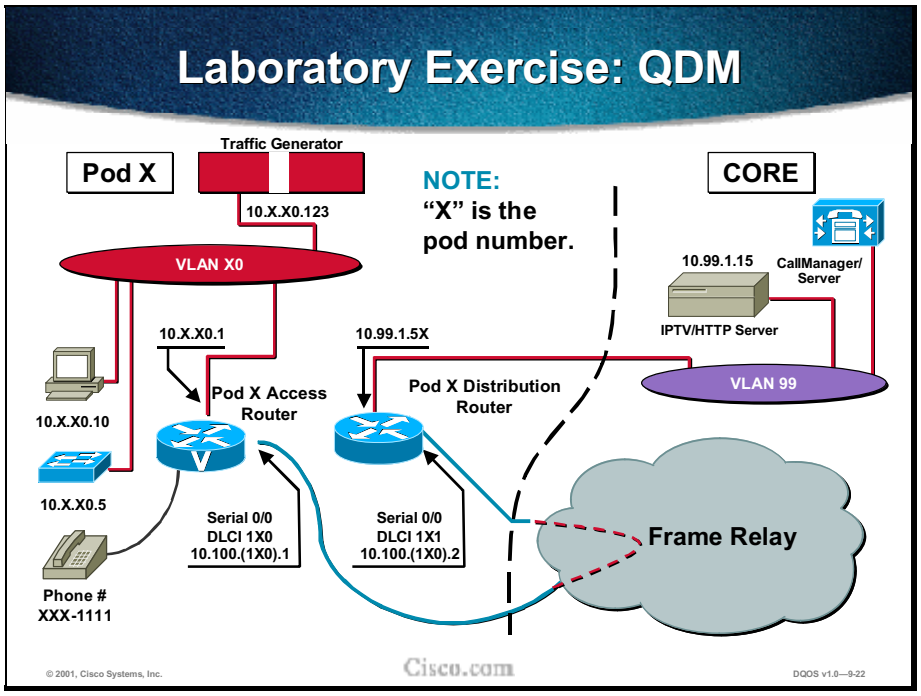
QDM 2.0 supports Netscape Navigator 4.5.1 or later and Microsoft Internet Explorer 5.0 or later. For the QDM 2.0 application, Microsoft Internet Explorer web browsers generally provide better overall performance than Netscape Navigator web browsers.

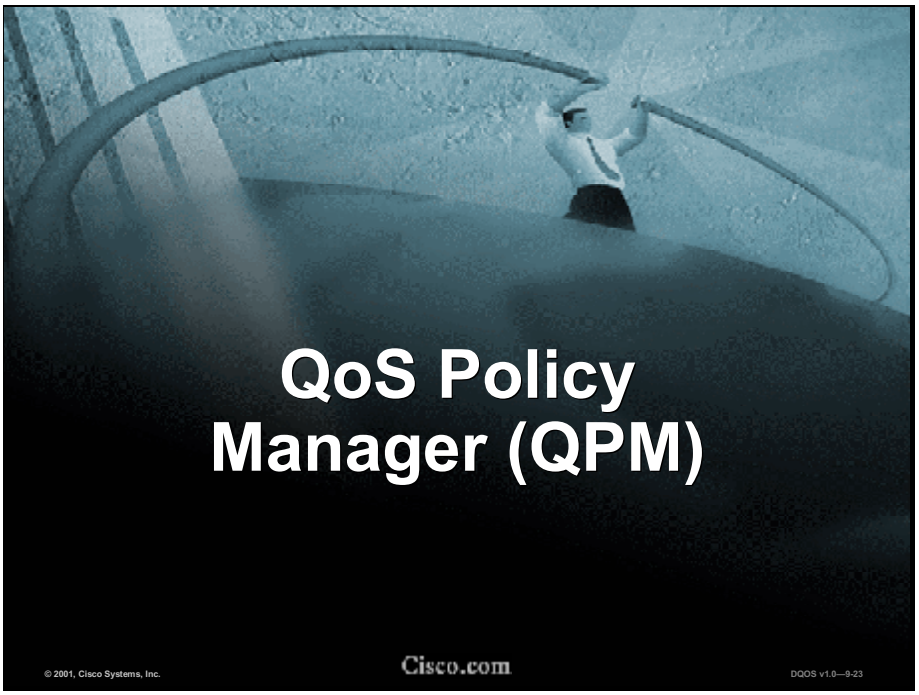
Screen-Size Requirement

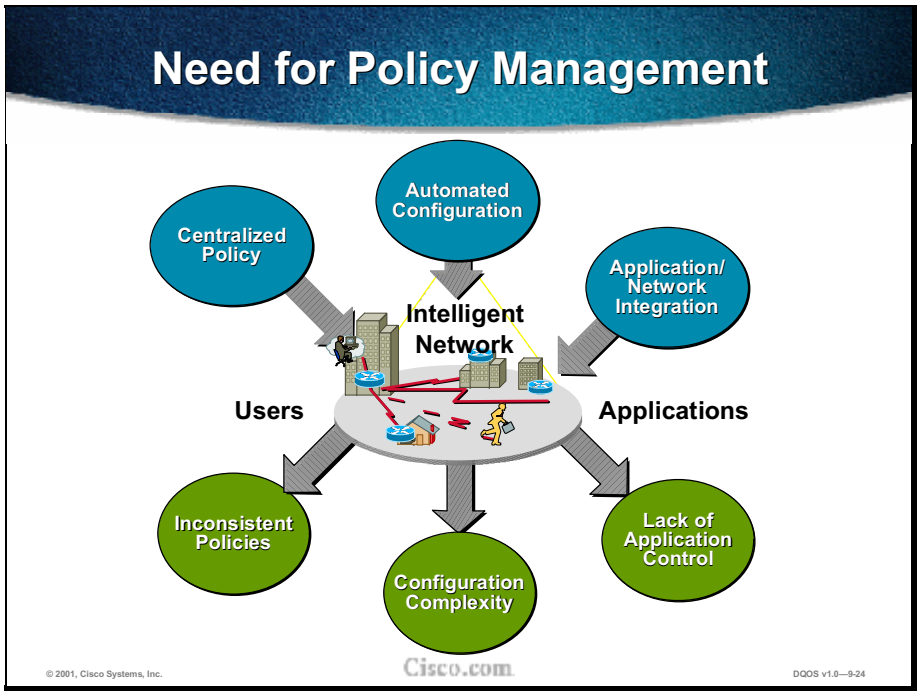
The minimum screen-size requirement for QDM 2.0 for the desktop area is 1024 by 768 pixels. With a smaller screen certain graphs might be unreadable and certain functions might be unusable.

VTY Lines

Cisco Systems recommends having 16 available virtual terminal (VTY) lines for QDM. If 16 VTY lines are unavailable, Cisco Systems suggests having at least five VTY lines for QDM. QDM might not properly install or run on systems with limited numbers of VTY lines.


Laboratory Exercise: QDM
Pod X
Traffic Generator
10.X.X0.123
VLAN X0
NOTE:
"X" is the
pod number.
CORE
10.99.1.15
CallManager/
Server
IPTV/HTTP Server
VLAN 99
10.X.X0.1
10.99.1.5X
Pod X Access
Router
Pod X Distribution
Router
10.X.X0.10
10.X.X0.5
Phone #
XXX-1111
Serial 0/0
DLCI 1X0
10.100.(1X0).1
Serial 0/0
DLCI 1X1
10.100.(1X0).2
Frame Relay
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—9-22





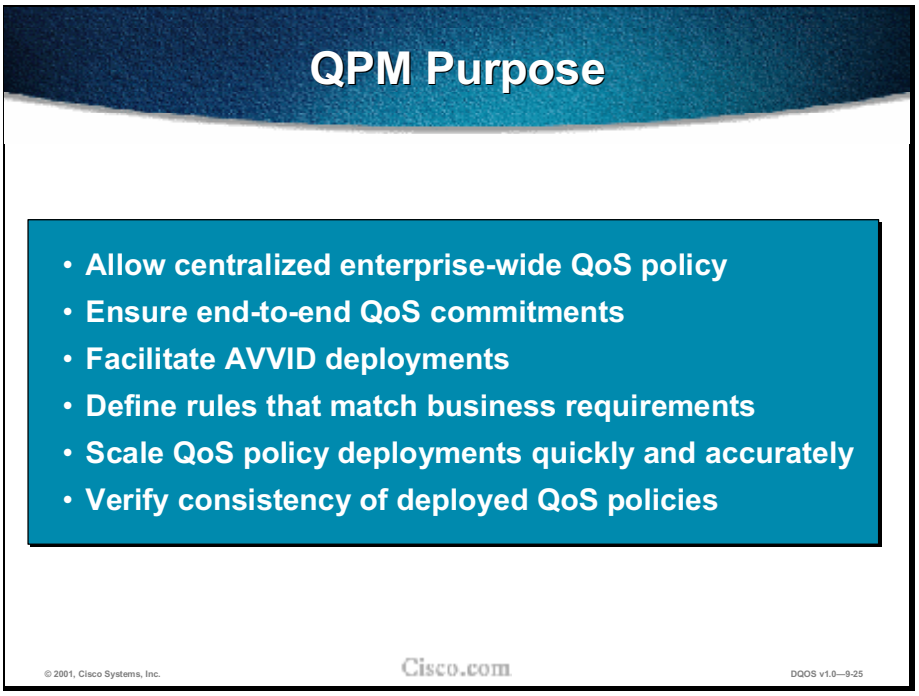
Quality policy is developed and deployed to ensure end-to-end QoS, to eliminate human error and to control the impact of emerging applications. These three challenges are addressed through QPM.

QPM offers:

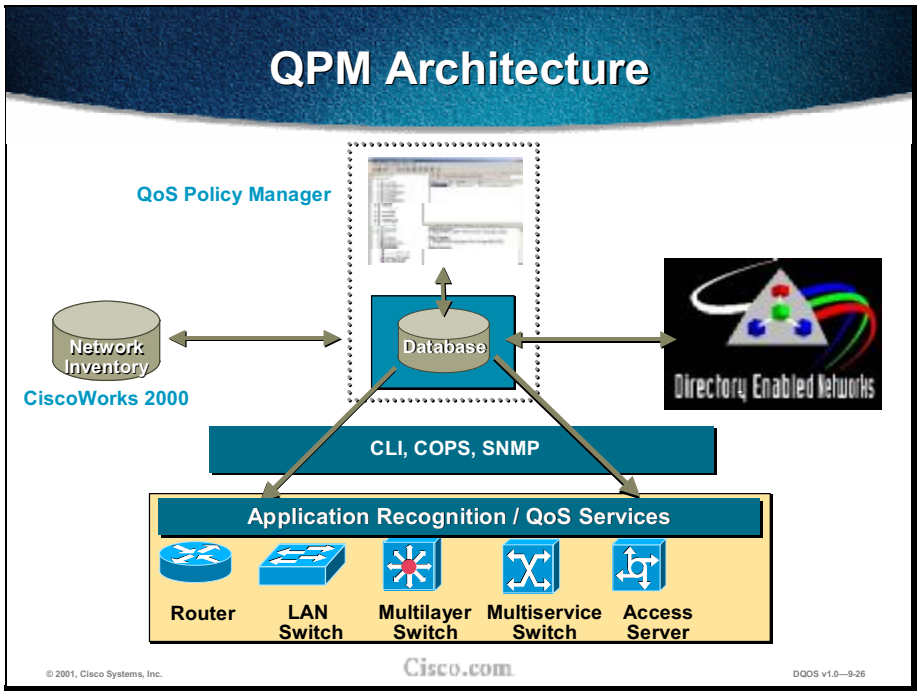
- Centralized policy—To ensure end-to-end QoS, policies need to be consistent throughout the enterprise. For manually configured QoS this is difficult and arduous to achieve.

- Automated configuration—QoS CLI is difficult, complex, and time-consuming to configure; additionally, it is highly prone to human error (such as typos) when scaling deployment.

- Application integration—Many applications are bandwidth-intensive and are often difficult to identify and control (for example, Napster); when a new application is introduced to the network, it would be difficult and/or time-consuming to manually reconfigure the devices to adapt to the new application.



QPM provides a means to deploy enterprise-wide QoS policy from a centralized location, ensuring commitments throughout the network, guaranteeing consistency, and making it easy to do accurately and quickly.



QPM maintains your QoS definitions and policies in a QoS database. These databases are maintained on the machine that runs the QoS Manager service in the database directory in the QPM installation directory.

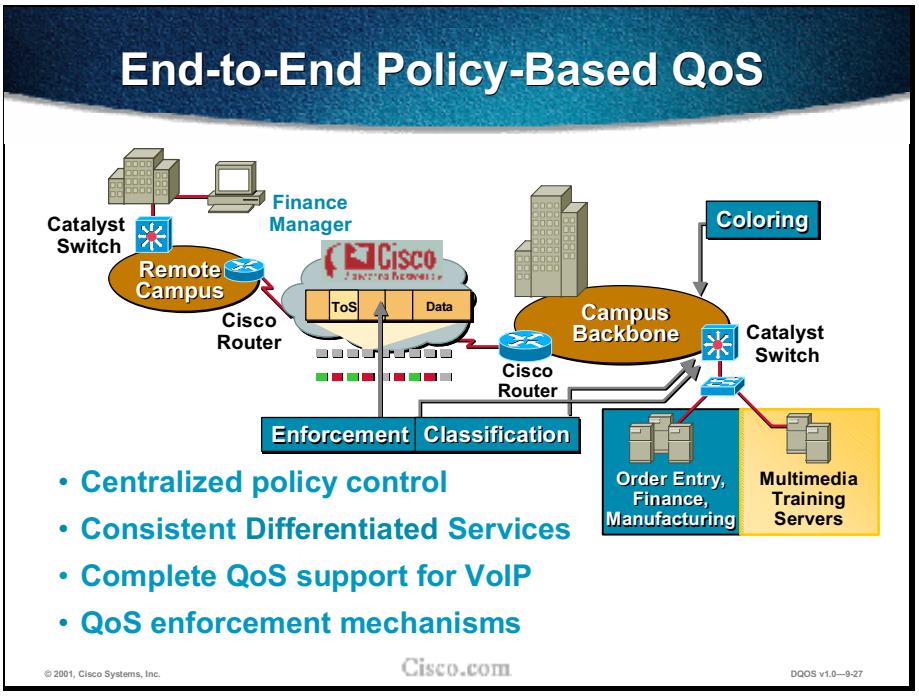
You can create more than one database. Typically, you should define no more than 200 devices in a single database. If you have more than 200 devices in your network on which you want to configure QoS, divide them into logical groups and define the groups in separate databases. For example, you could create one database for core devices and another database for edge devices. The 200-device limitation is not a rigid one, but is based on Cisco test modeling.

The figure above shows how the QPM database links to the CiscoWorks 2000 network inventory and with LDAP (Lightweight Directory Access Protocol).

With Cisco's QoS Policy Management solutions you can define QoS policies for all types of network devices:

- QPM-PRO provides provisioned QoS control using differentiated services to enforce QoS end-to-end. QPM-PRO's configuration interface allows you to define and deploy policies more easily than using device commands directly via the command-line interface (CLI). QPM-PRO supports a broad base of Cisco devices and QoS features.

- QPM-COPS supports newly emerging, standards-based technology for both signaled and provisioned QoS using the industry-standard COPS protocol. QPM-COPS complements the provisioning provided by QPM-PRO by allowing the definition and classification of user-based policies stored in a standard LDAP directory.



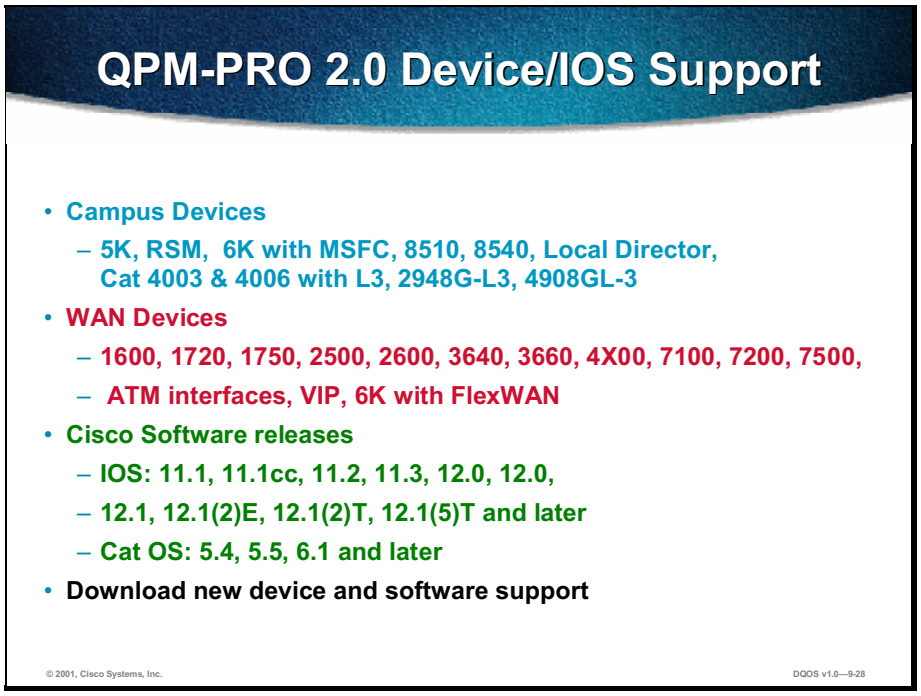
QPM delivers differentiated services across network infrastructures with converged voice, video, and data applications simply by taking advantage of Cisco IOS® and Catalyst® OS software with built-in QoS mechanisms in local-area network (LAN) and wide-area network (WAN) switching and routing equipment.

- Centralized, simplified policy control—Network administrators can use the QPM graphical user interface for accurate end-to-end QoS configuration and automated, reliable policy deployment, while eliminating device-by-device command streams.

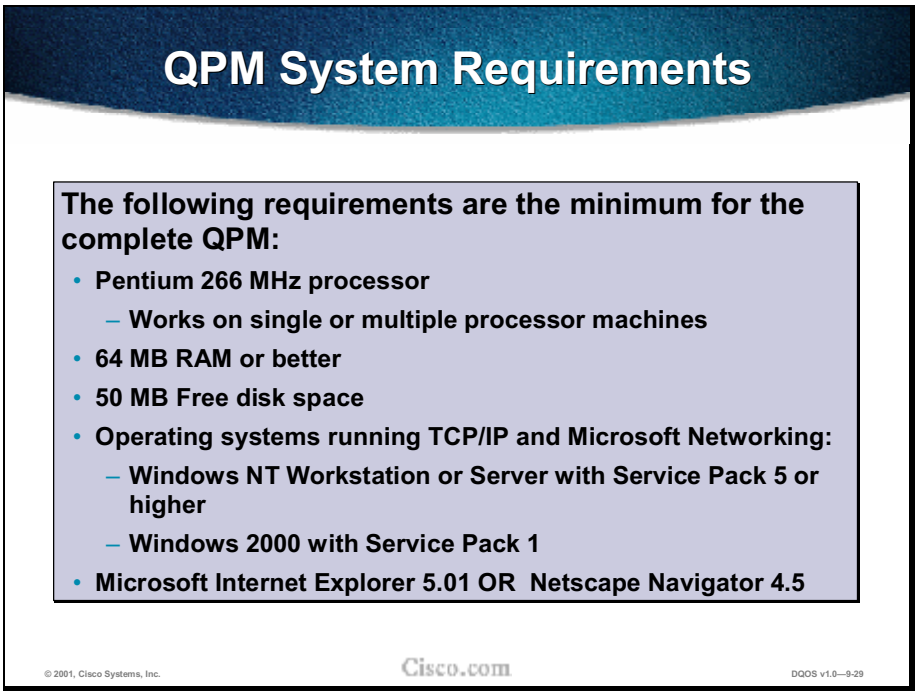
- Differentiated services for various types of traffic—Achieve business-driven service levels across the enterprise network by using QPM to configure traffic classification and allow QoS policy enforcement through Cisco devices.

- Complete QoS support for voice over IP—Define and apply policies that ensure strict priority for voice traffic in Cisco AVVID (Architecture for Voice, Video and Integrated Data) networks.

- QoS Enforcement Mechanisms—An integral part of Cisco Content Networking, QPM delivers the appropriate service-level to business-critical applications by supporting the extension of IP packet classification to include application signature, web URLs, and negotiated ports.



The figure above lists the campus, WAN devices, and Cisco IOS releases that support QPM-PRO as of February 2001.

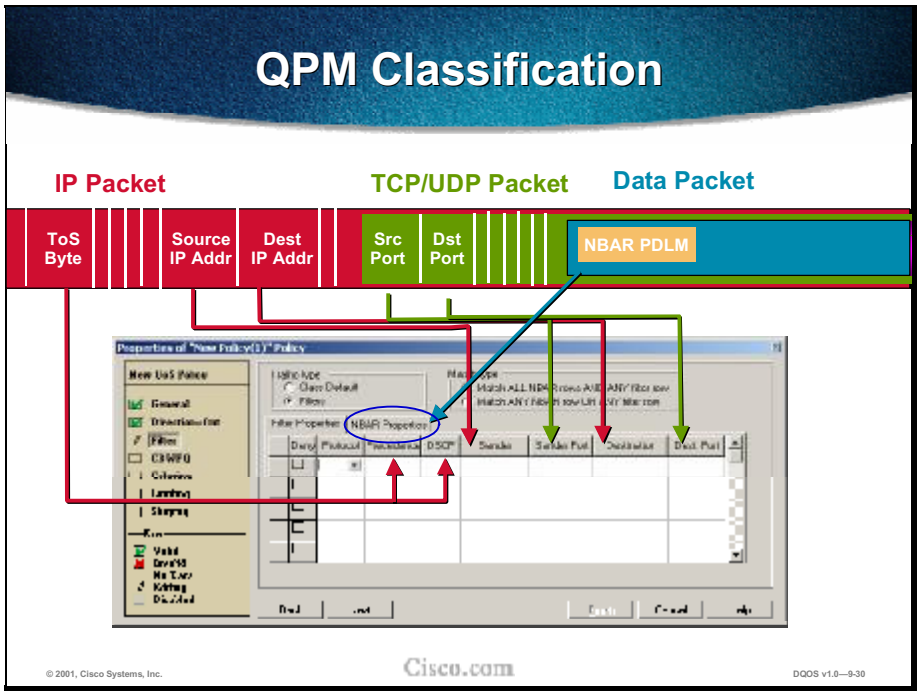


The system requirements for complete QPM are listed in the figure above.

Remote QPM

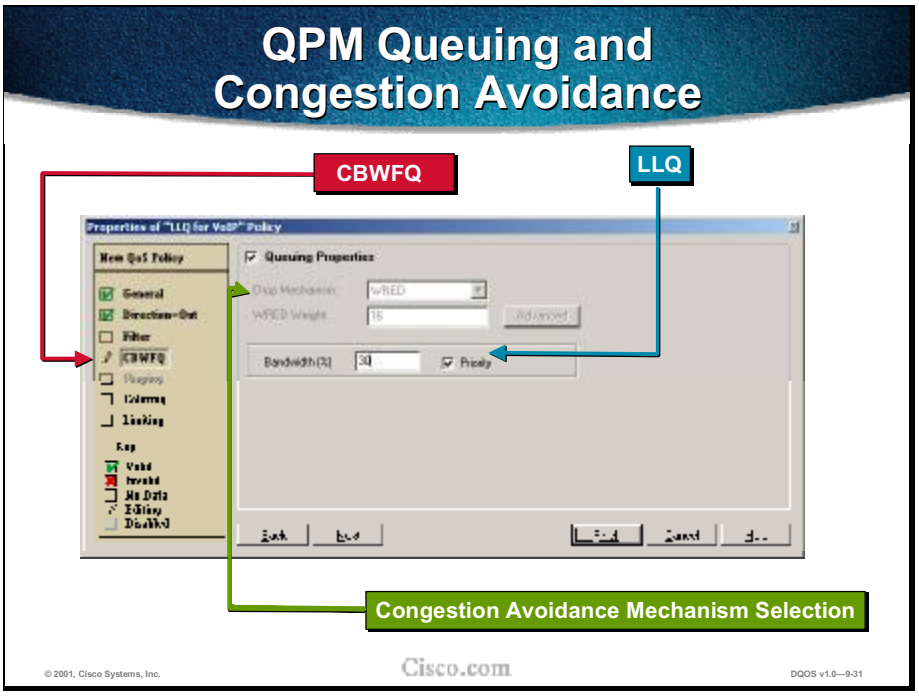
For remote QPM, any of these operating systems running TCP/IP and Microsoft Networking can be used:

- Windows 95 with the year 2000 upgrade
- Windows 98
- Windows 2000 with Service Pack 1
- Windows NT Workstation or Server with Service Pack 5 or higher



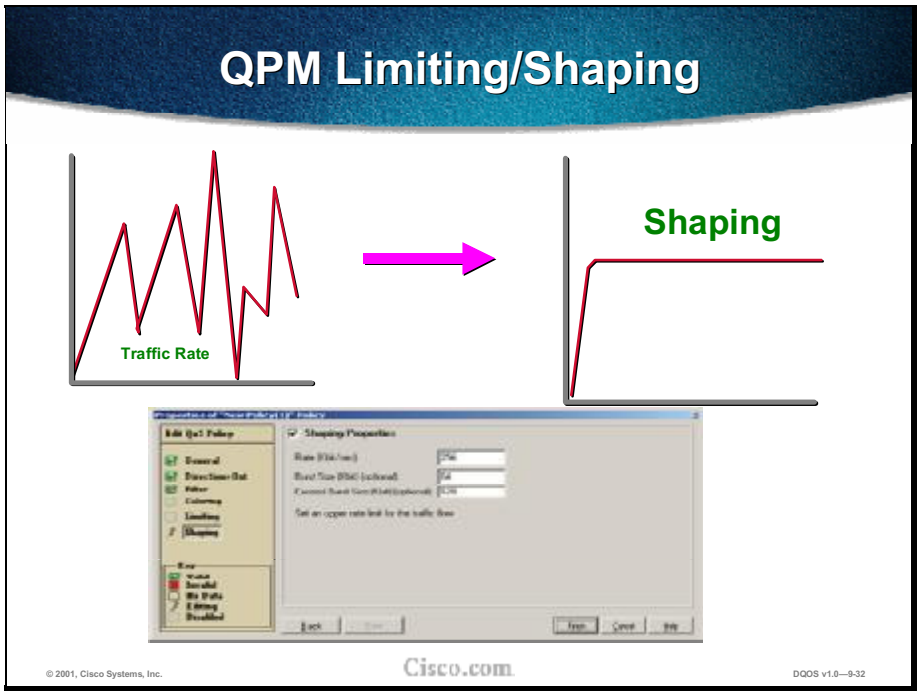
Creating a QoS policy with QPM is done by a simple step-by-step wizard:

1. General Properties: A name is given to the policy and optional description; also the policy can be enabled or disabled.
2. Direction: Specifications are made to apply this policy in the IN or OUT direction of the interface(s).
3. Filter (screenshot above): It serves to classify the traffic to which QoS will be applied; parameters include IP source/destination addr, IPP/DSCP, TCP/UDP port, and NBAR parameters.
4. Queuing: Once the traffic has been identified, it can be queued, colored, limited, or shaped. When the queuing mechanism has been applied to the interface, the software will determine what options are available in this step.
5. Coloring: Coloring options.
6. Limiting: Limiting parameters.
7. Shaping: Shaping parameters.

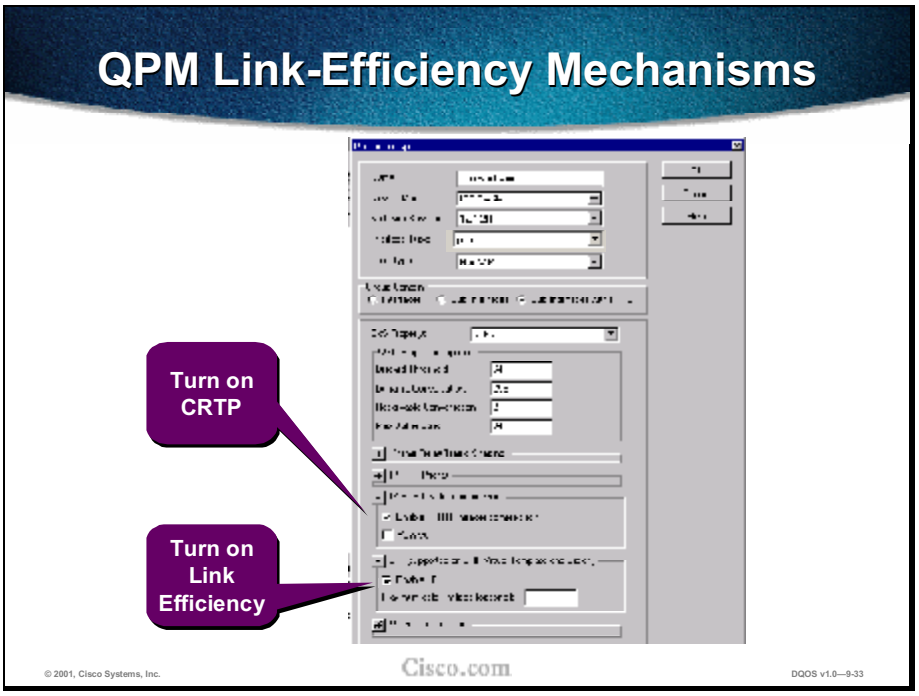


The queuing property assigned to the interface determines the subparameters presented under this option. In this case CBWFQ has been assigned to the interface, and additionally 30 percent of the available bandwidth has been reserved for LLQ.

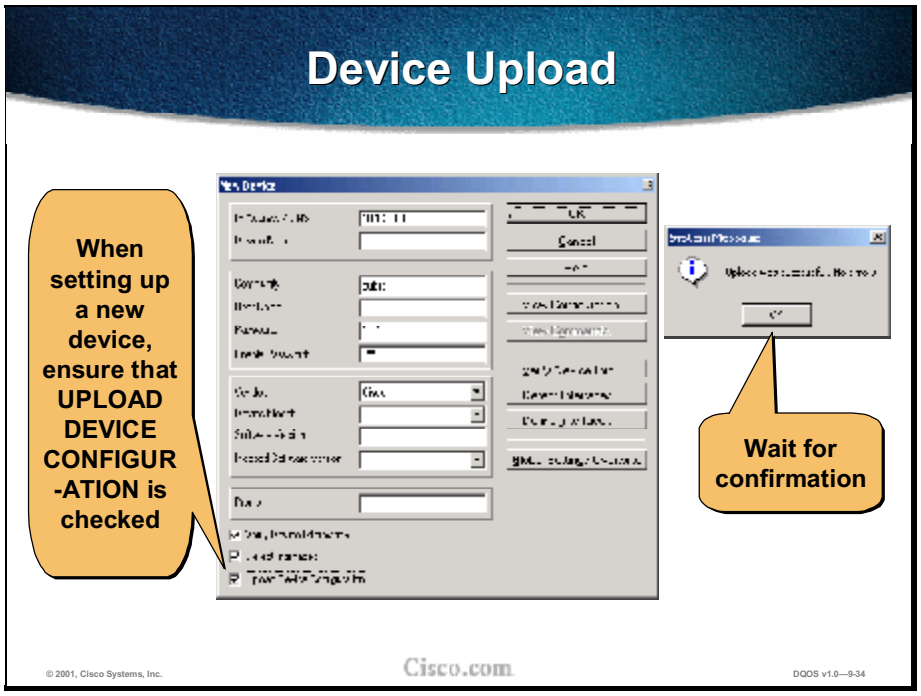
Also highlighted is the option to enable WRED as a congestion avoidance mechanism (in this particular example, WRED has been grayed-out as an option since it is incompatible with LLQ).



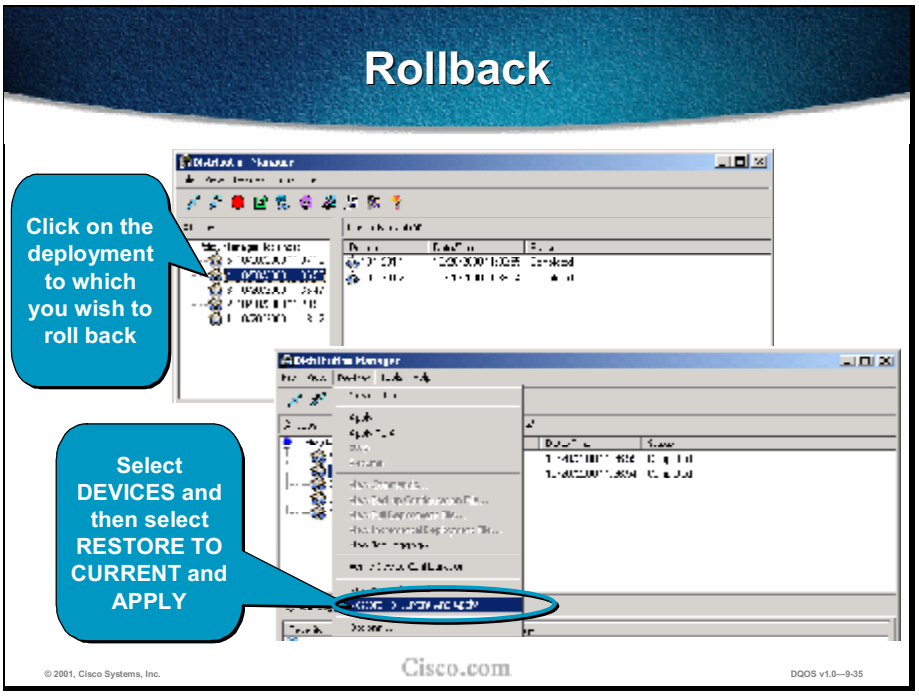
The standard parameters to configure limiting and shaping are presented in a simple GUI.



Turning on link-efficiency mechanisms is a matter of a couple of clicks; in this particular example, several interfaces (bundled in a device group) are enabled with CRTP and LFI with just two clicks.

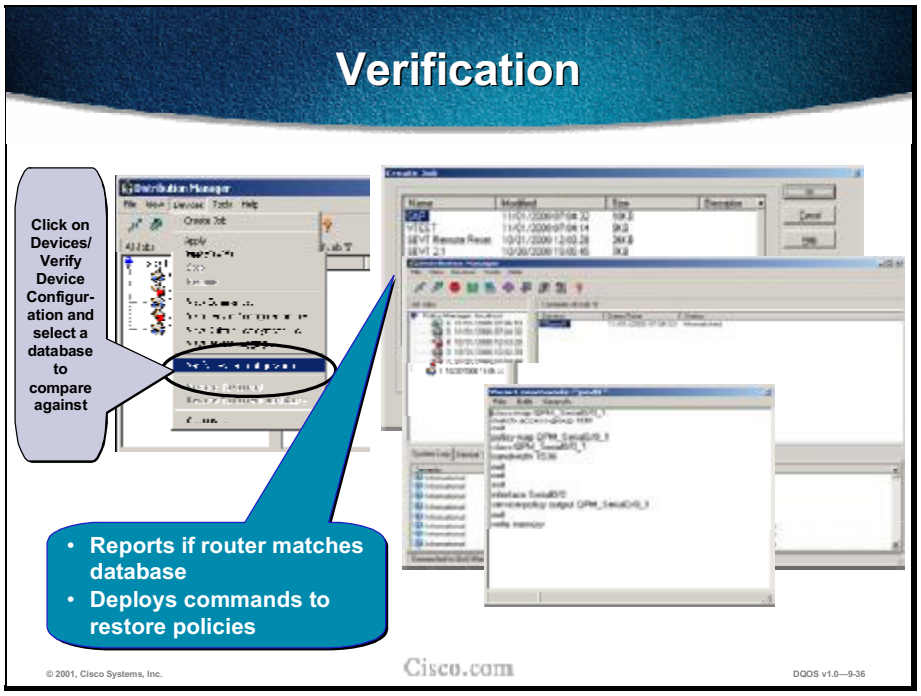


When devices are being added to QPM, the option Upload Device Configuration is presented. This allows the administrator to have QPM parse the existing device configuration for QoS policies and then graphically represents them within QPM for ease of management.



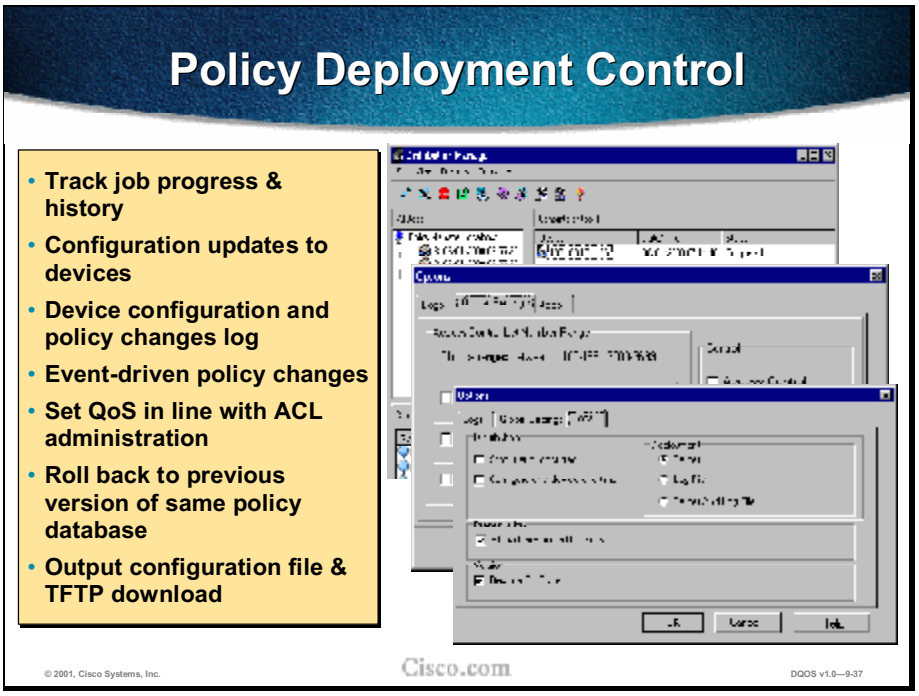
If a policy is deployed and is having no effect or an adverse effect on the network, QPM presents the option to quickly revert back, or roll back, to a previous configuration.

It is similar to an “Undo” feature within a word processor.



Sometimes individuals with access to network devices modify or delete QoS policies that an administrator has deployed via QPM. If the administrator suspects that this is the case, he can take advantage of QPM's verification feature to check if the current device configuration matches or mismatches what QPM previously deployed. Additionally, he can quickly correct any mismatches, if necessary.

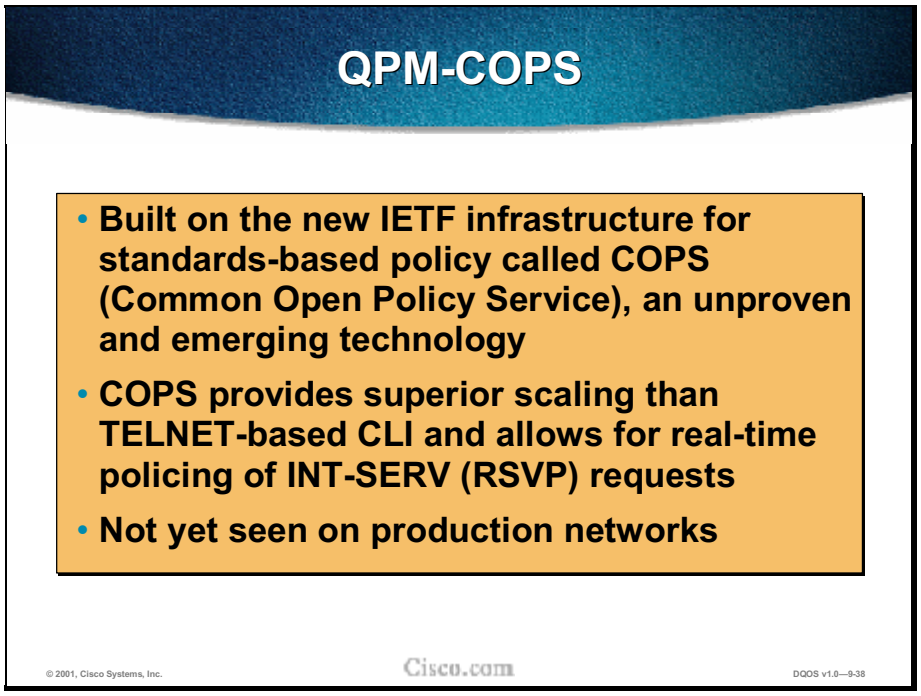
To use this function, click **Devices/Verify Device Configuration** and select a database to compare against. QPM will report if policies in the router match those against the database (or if they have been changed outside of QPM) and can quickly deploy any necessary commands required to restore the policies.



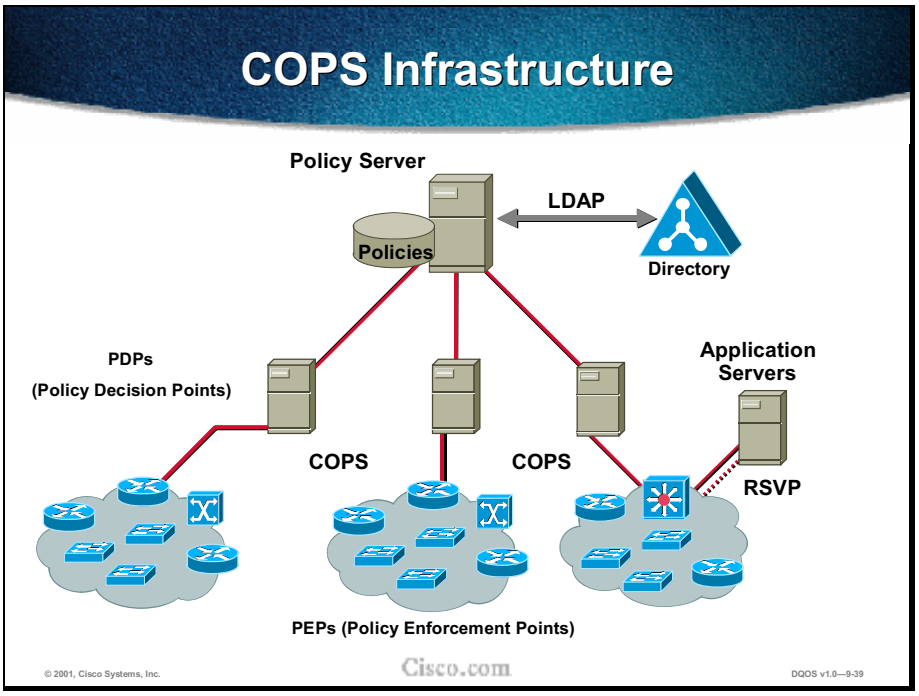
QPM includes tools to enable the network administrator to maintain control of the network.

The network administrator can:

- Track job progress and history
- Read incremental configuration updates to devices
- Obtain detailed log of device configuration and policy changes
- Drive policy changes from events
- Set QoS access control list number range in line with ACL administration
- Roll back to previous version of same policy database in case of errors or unwanted network behavior
- Print the configuration file and conduct a TFTP download of the configuration file



It cannot be stressed enough that COPS is an emerging technology that has yet to prove itself. There is much heated debate as to whether COPS will be the standard policy protocol of the future or not. These few slides provide an overview and context for QPM-COPS, but they are not intended as detailed curriculum. Often the inclusion of COPS/QPM-COPS takes a significant and time-consuming tangent from more relevant technologies.

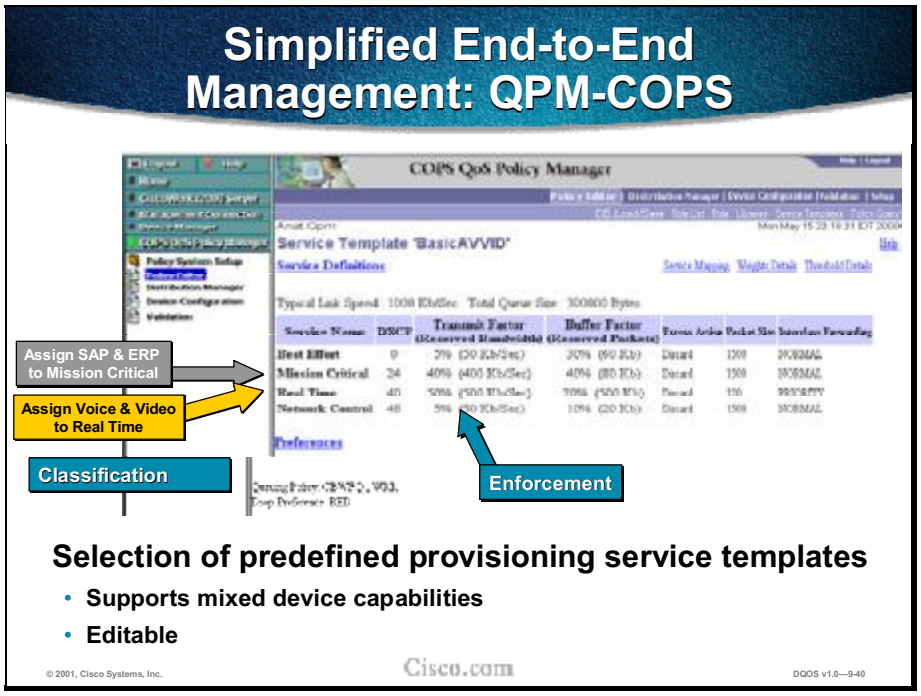


The COPS infrastructure consists of three primary components:

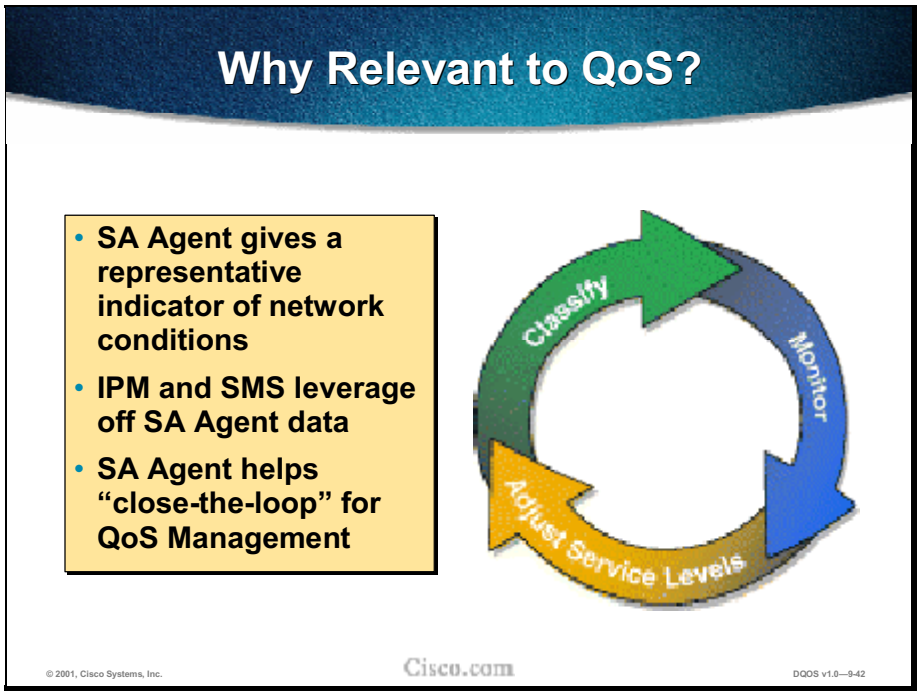
- Policy servers—A centralized policy server where administrators define enterprise-wide policies; these policies may be exported or imported from a directory via LDAP.

- Policy Decision Points (PDPs)—PDPs are various servers throughout the enterprise that maintain open connections to the policy server and are quickly updated of any new change in policy. PDPs are, in effect, proxy policy servers and facilitate scaling and fast deployment; PDPs make RSVP policing decisions; PDPs can control anywhere from 50 to 200 network devices each.

- Policy enforcement points (PEPs)—PEPs are the actual network devices that maintain an always-open TCP connection (port 3288) running the COPS protocol to their primary and also to their secondary PDP; whenever a device comes online, it will be preconfigured to function in a specific role and will immediately contact its PDP to request the policies that correspond to the role that it has been assigned. Any RSVP requests received are forwarded to the PDP for a decision to accept or reject the request.

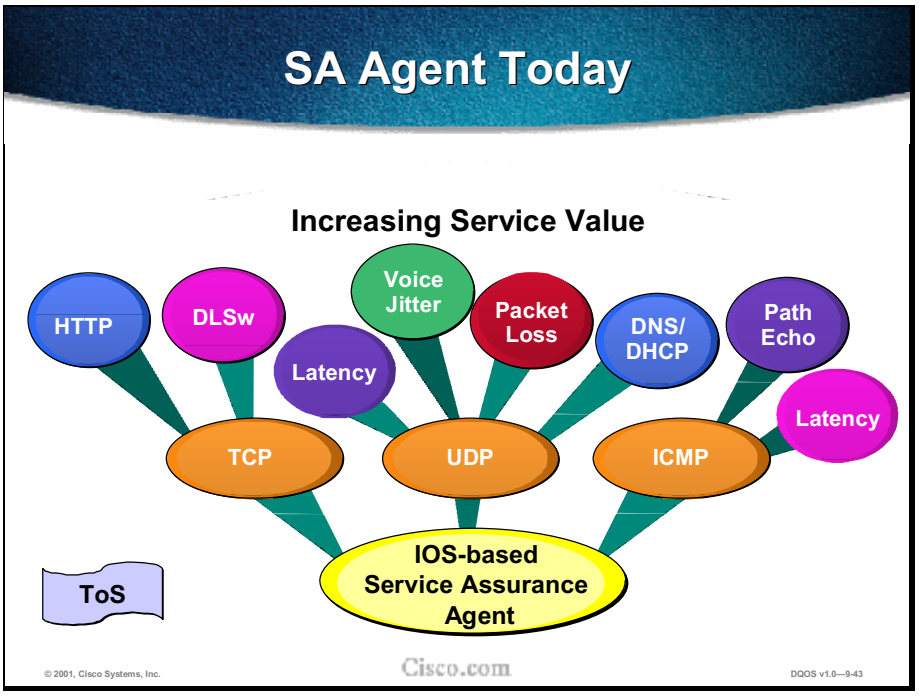


QPM-COPS is integrated into the CiscoWorks2000 family of management products and provides a GUI-based user-friendly template to facilitate use-accelerated implementation.



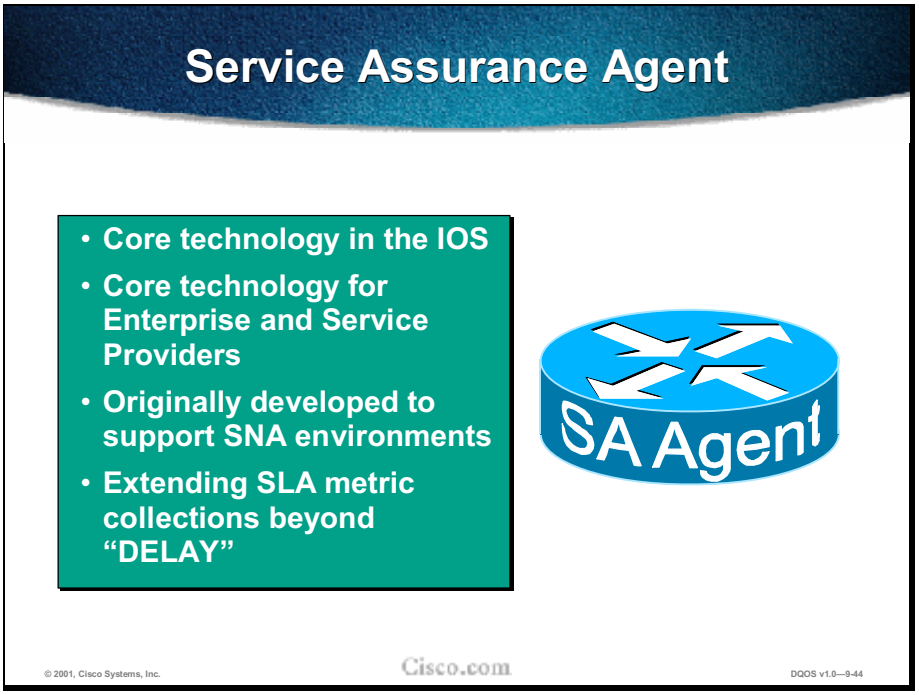
Another tool for network managers is the Service Assurance Agent (SA Agent). SA Agent generates synthetic traffic. This traffic is used to monitor network performance by measuring key service level agreement (SLA) metrics such as response time, network resources, availability, jitter, connect time, packet loss, and application performance metrics, giving a representative indicator of network conditions. The data that is collected by SA Agent is used by the applications Internetwork Performance Monitor (IPM) and Service Management Solution (SMS) (taught in another section) to provide charts and graphs to facilitate baselining, monitoring, trending, and troubleshooting. By providing information on the network, SA Agent "closes the loop" for QoS management.

The SA Agent maintains a local history of up to two hours, ensuring that no valuable data is lost during intermittent link failures.



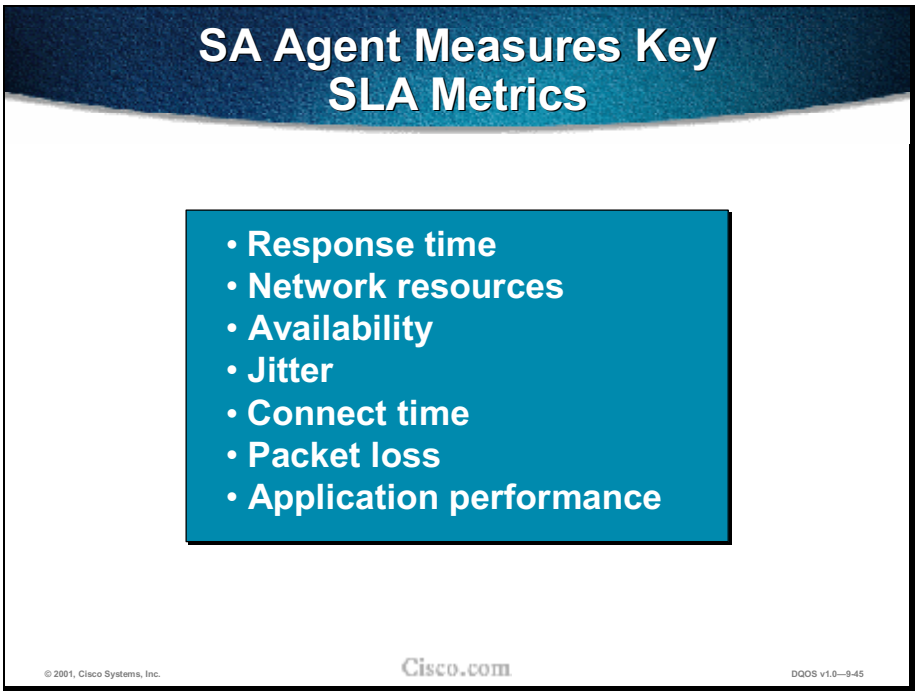
Service Assurance (SA) Agent is a management engine technology used to collect service-level metric data required to validate that service levels are being met. SA Agent technology is embedded in the Cisco IOS software. Generally speaking, the SA Agent provides a way to configure a Cisco IOS device to perform tests in the network to end systems or to other Cisco IOS devices.

SA Agent technology supports TCP, UDP, and ICMP communications methods and a wide range of test types or SA operations such as jitter, path echo, and packet loss. These SA operations are set up to send test packets and measure the service-level metrics as required.



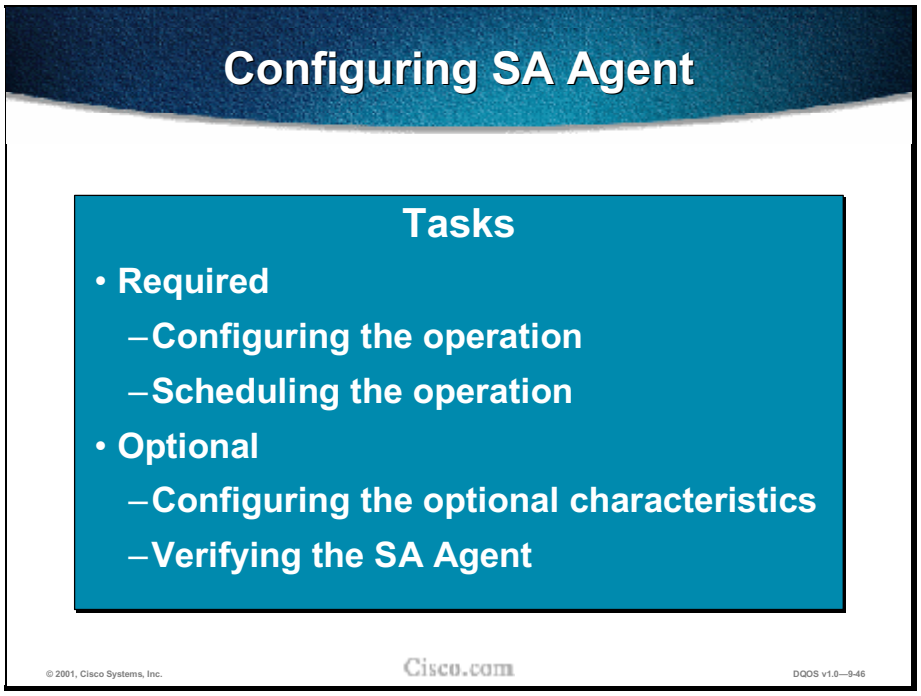
For some time Cisco has been working on building in intelligence in the IOS that supports network and service management. This technology was first released as Response Time Reporter (RTR) or RTTMON MIB and was primarily focused on delay measurements in IBM environments. It is now common to all IOS releases.

The technology has been considerably enhanced to support Service Level Management (SLM), extending it to measure various other SLM metrics beyond just delay. As such, it has been relaunched as the Service Assurance Agent (SA Agent).



The SA Agent allows you to measure and monitor the following:

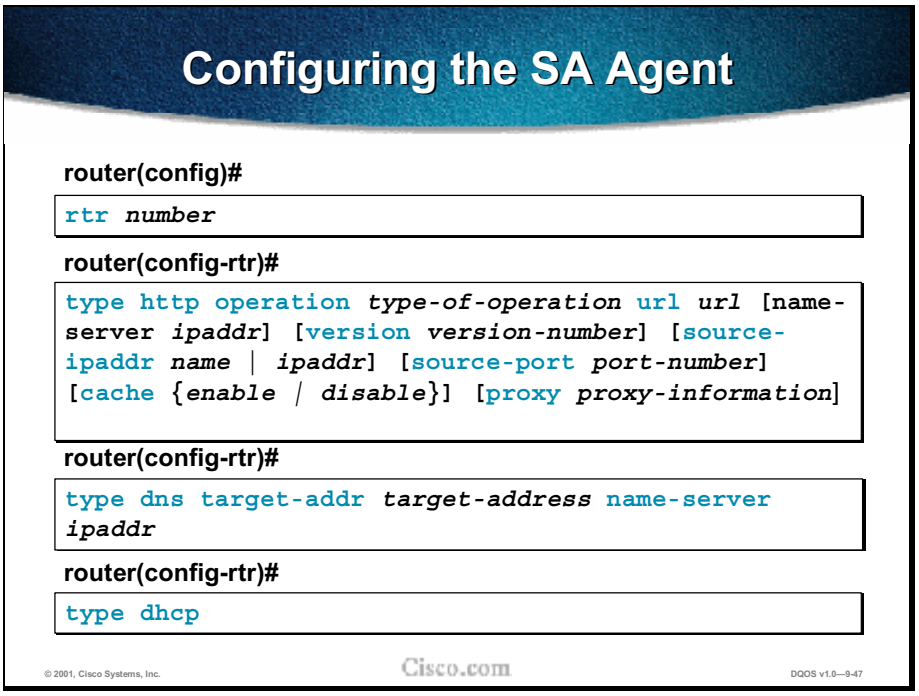
- SLA metrics such as round-trip response time and availability
- Web metrics and applications
- Voice over IP (VoIP) metrics such as availability of synthetic VoIP traffic, jitter, connect time, and packet loss



Response time and availability information is collected by operations (formerly known as probes) that you configure on a Cisco device such as a router or access server. Operations use synthetic packets specifically placed in a network to collect data about the network. These packets simulate other forms of network traffic, as determined by the type of operation you configure. SA Agent operations are given specific identification numbers so you can track the various operations you configure and execute. SA Agent operations are configured in RTR configuration mode. You must configure the operation type before you can configure any of the other characteristics.

It is also required to schedule the operation.

Network managers can configure SA Agent to collect additional information with the optional characteristics option. The full list of optional characteristics can be found at: http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/fun_r/frprt3/frd3001c.htm



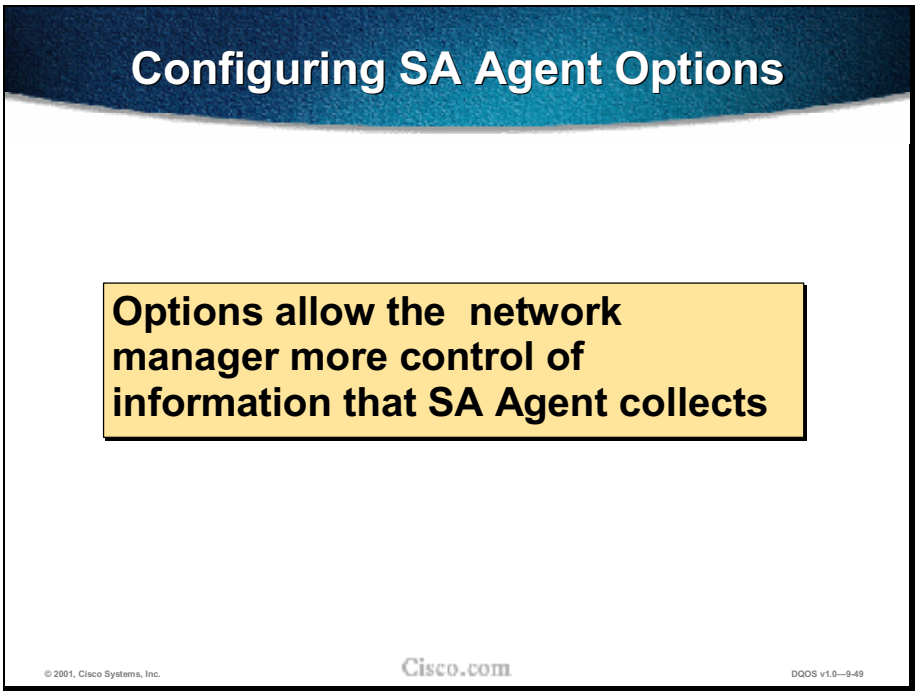
These are the commands for configuring SA Agent. To configure operations, you must first change the prompt to the **config-rtr** prompt. Then the operations can be configured. In this slide the commands for the operations of HTTP, DNS, DHCP are shown. Other operations that are available include jitter, DLSw, FTP, and one-way delay for jitter. For a complete list of commands refer to: http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/fun_r/frprt3/frd3001c.htm

Scheduling SA Agent

router(config)#

```
rtr schedule number [life seconds] [start-time
{pending | now | hh:mm  [month day | day month]}]
[ageout seconds]
```

- **Configures the time parameters for an SA Agent operation**
- **Operations can be configured and scheduled for execution at a later time**



SA Agent has many options to provide the network manager with additional information. These commands include:

buckets-of-history-kept ***size***

distributions-of-statistics-kept ***size***

filter-for-history **{*none* | *all* | *overthreshold* | *failures*}**

frequency ***seconds***

hours-of-statistics-kept ***hours***

http-raw-request

lives-of-history-kept ***lives***

lsr-path **{*name* | *ipaddr*} [*name* | *ipaddr*] ...**

owner ***text***

request-data-size ***bytes***

samples-of-history-kept ***samples***

statistics-distribution-interval ***milliseconds***

tag ***text***

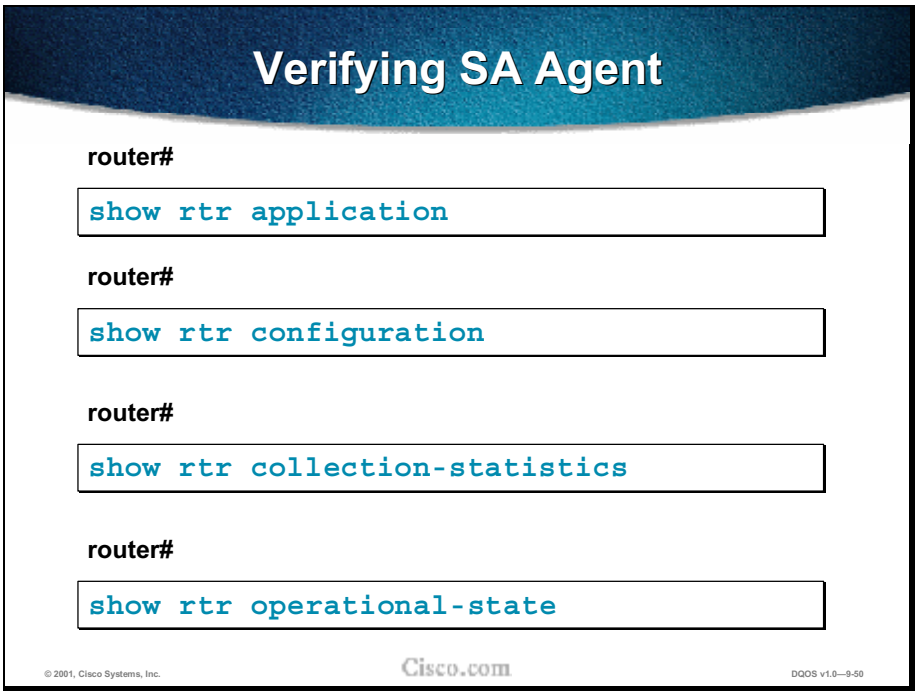
timeout ***milliseconds***

threshold ***milliseconds***

tos ***number***

verify-data

A full description of each of these commands can be found at:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120t/120t5/saaoper.htm#xtocid1543115



Properly, the network manager has four commands. The **show rtr application** command verifies how many operations are running. The **show rtr configuration** verifies that the SA Agent is configured. The **show rtr collection-statistics** verifies that the statistics are being collected for the operation. The **show rtr operational-state** verifies that the operations are running.

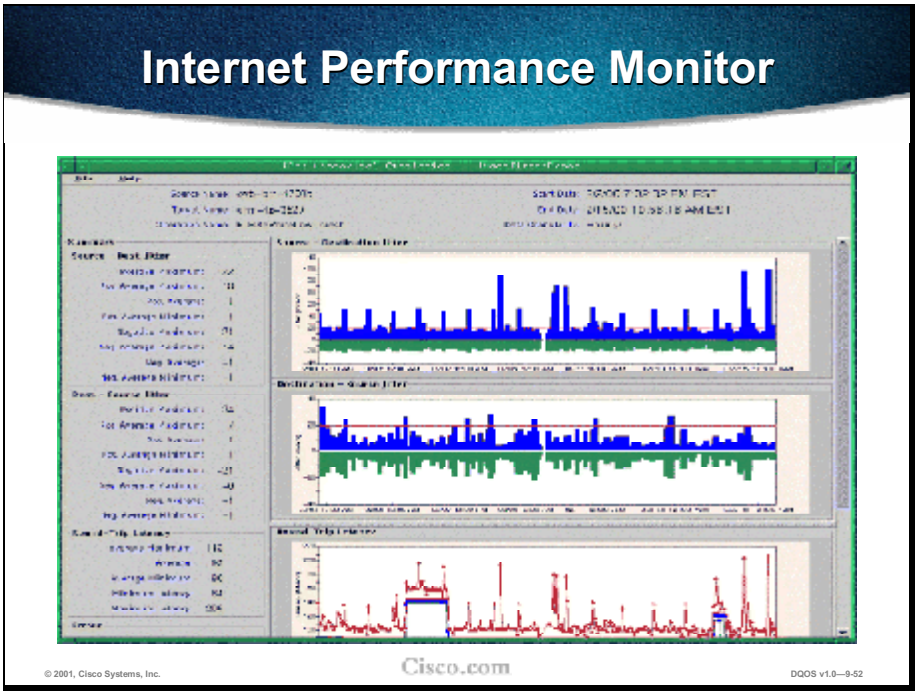
This is an example of the output on the **show rtr collection-statistics** command.

```
        Collected Statistics

Entry Number:1

HTTP URL:http://172.20.150.200

Start Time:*00:01:16.000 UTC Mon Mar 1 1993

             Comps:1              RTTMin:343

             OvrTh:0              RTTMax:343

        DNSTimeOut:0              RTTSum:343

        TCPTimeOut:0             RTTSum2:117649

        TraTimeOut:0              DNSRTT:0

          DNSError:0            TCPConRTT:13

         HTTPError:0             TransRTT:330

          IntError:0             MesgSize:1771

            Busies:0
```

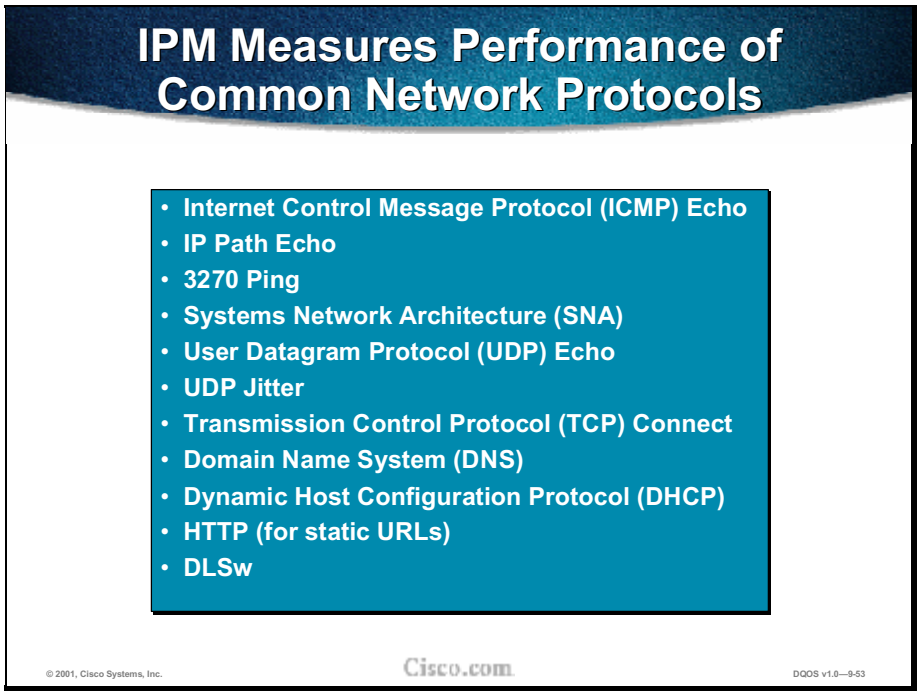


Internetwork Performance Monitor
and Service Management Solution
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—9-51

The next section covers the two management tools: Internetwork Performance Monitor (IPM) and Service Management Solution (SMS).



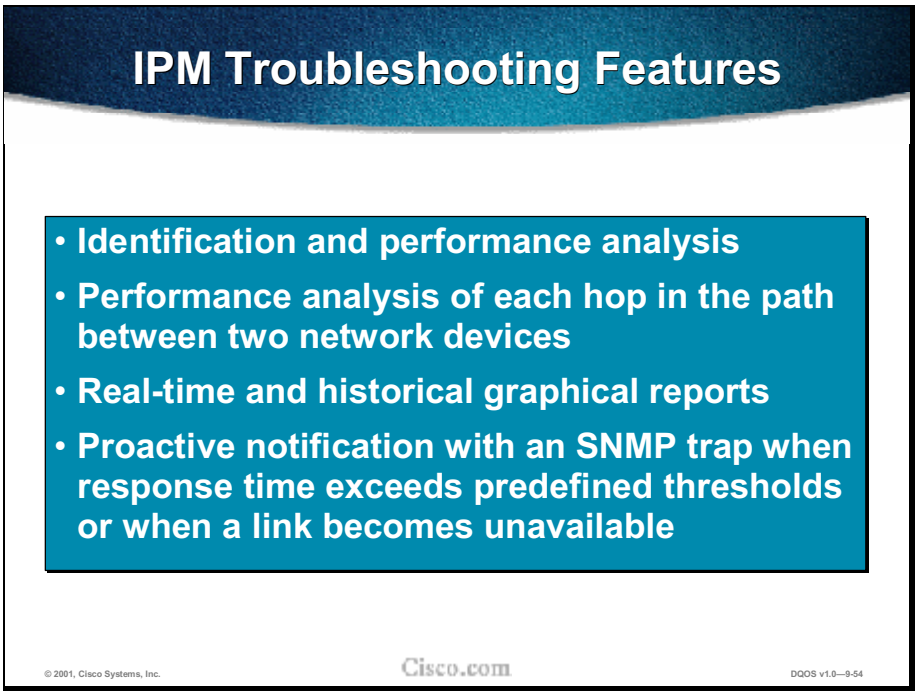
Internetwork Performance Monitor (IPM) leverages data generated by the SA Agent, in addition to other router counters, to present (graphically) an overview of networking conditions.



Furthermore, for networks that have deployed QoS based on IP Precedence values, IPM can measure performance for any of these protocols across any of the six values of IP Precedence. As a result, IPM provides an accurate representation of network performance by measuring the performance of "synthetic" traffic that closely resembles "real user" traffic.

Once an IPM collector is configured and deployed in the source router, IPM will continuously collect performance information, based on the parameters of the collector that have been defined, for the following performance metrics:

- Latency
- Jitter (for UDP jitter operation type only)
- Availability
- Errors
- Packet loss

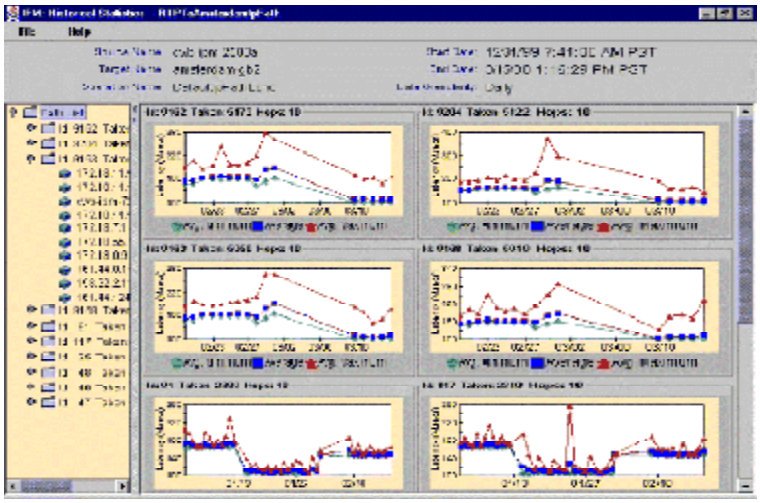


IPM enables the network engineer to proactively manage network response-time problems. IPM notifies the network engineer when network response time degrades or a monitored link becomes unavailable and helps pinpoint the device or link causing the problem.

IPM enables performance measurements to be taken automatically for all paths between two devices on a network or for each hop (that is, link or device) in the path between two network devices. Network managers can quickly and easily narrow down the source of a performance problem to a single hop in the network. As a result, rapid problem diagnostic capabilities lead to higher network availability by allowing network managers to alleviate performance bottlenecks quickly.

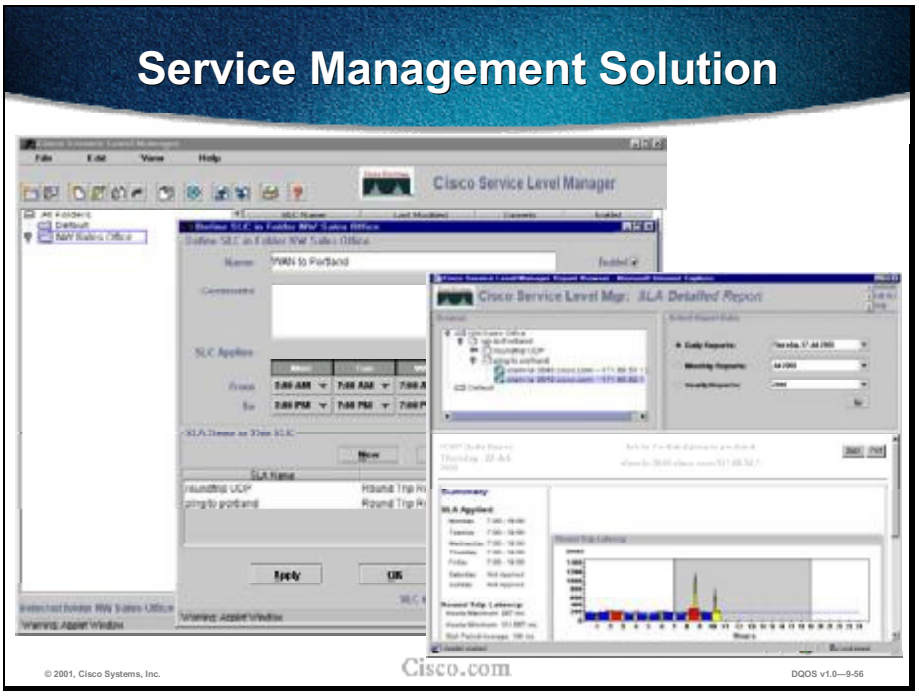
IPM provides real-time and historical graphical reports of response time between two network devices. It also proactively notifies the network administrator with an SNMP trap when response time exceeds predefined thresholds or when a link becomes unavailable.

IPM Performance Monitoring Graphs



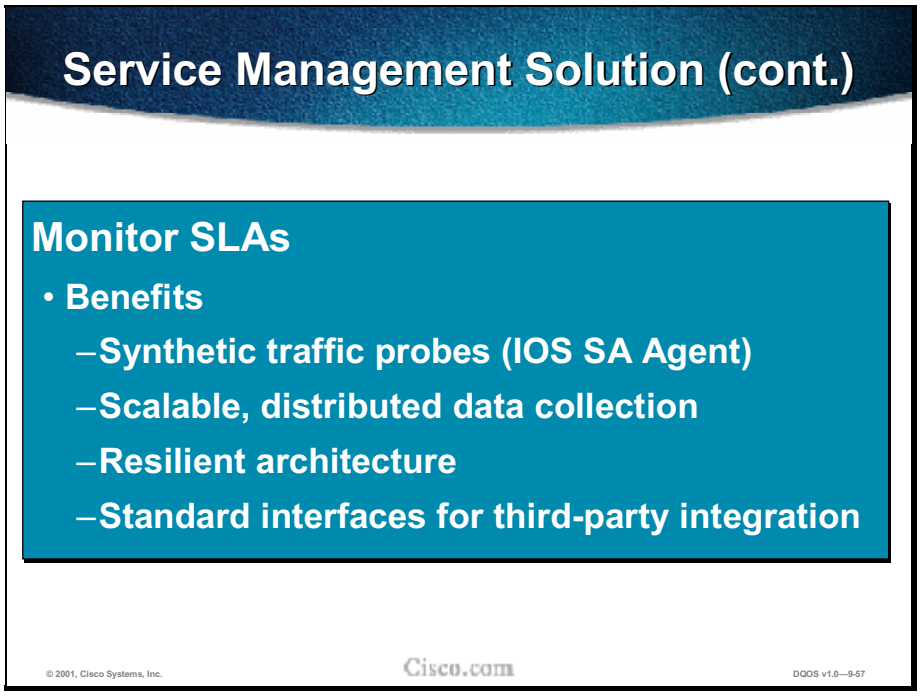
Cisco.com

Graphs for the entire route, or hop by hop, can quickly provide the administrator with data as to where congestion is occurring.



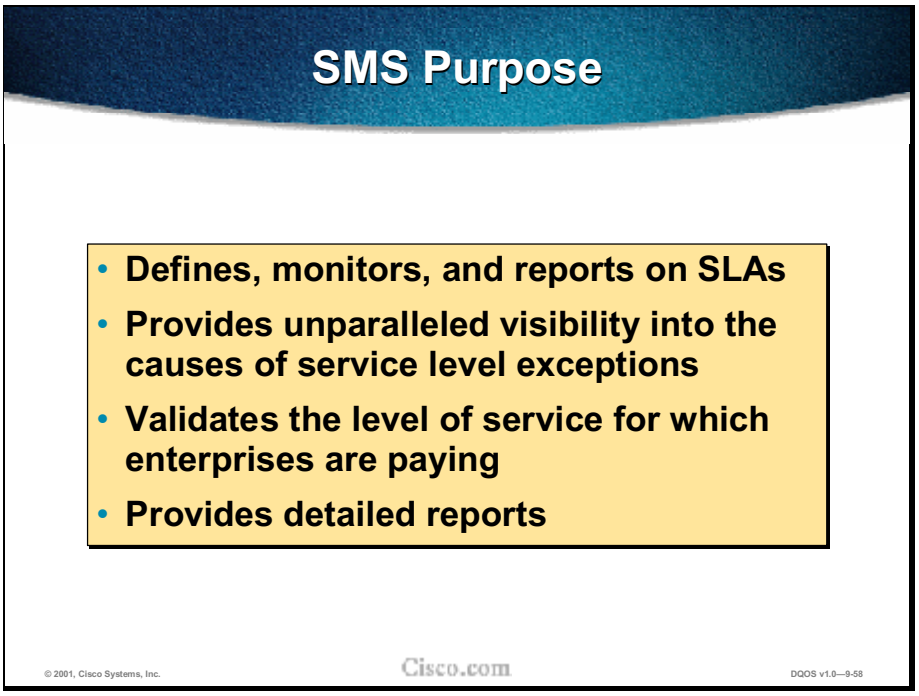
The Service Management Solution (SMS) is designed to generate management reports displaying average, aggregated statistics of the network's ability to support service level agreements.

CiscoWorks2000 Service Management Solution is different from the Internetwork Performance Monitor. While both CiscoWorks2000 Service Management Solution and the Internetwork Performance Monitor employ the Cisco IOS Service Assurance Agent, they address very different aspects of network management. The Service Management Solution is designed to generate management reports displaying average, aggregated statistics of the network's ability to support service level agreements. Internetwork Performance Monitor is designed as a network response time and availability troubleshooting application. Basically, IPM reports what is happening at a specific point in time, while SMS collects data over periods of time and then aggregates and summarizes the data for the administrator. IPM monitors, while SMS trends.

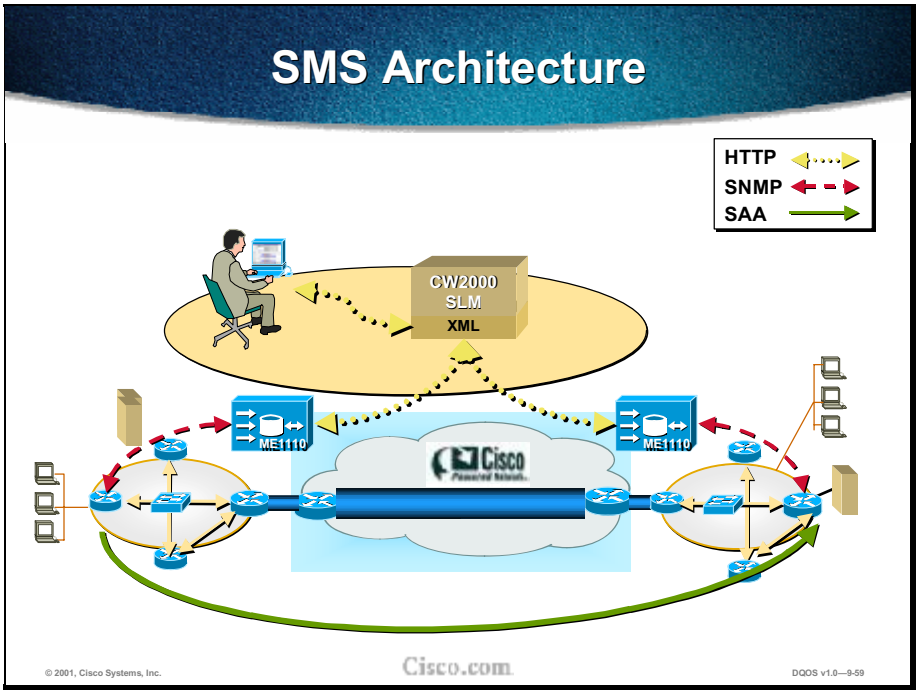


The CiscoWorks2000 Service Management Solution enables users to monitor and verify that their SLAs are being met. This allows enterprises to confirm that ISPs are providing committed service levels or for ISPs to differentiate their services from competitors. For users currently not deploying service level agreements, CiscoWorks2000 Service Management Solution provides a great starting point for establishing a baseline for the performance of the network, related to network response times, availability, and variability in network latency.

Through a combination of NT- or Solaris-based server software and Management Engine ME1110 hardware appliances, Service Management Solution interacts with the Cisco IOS Service Assurance Agent to monitor average latency, jitter, and lookup times for a variety of network services.

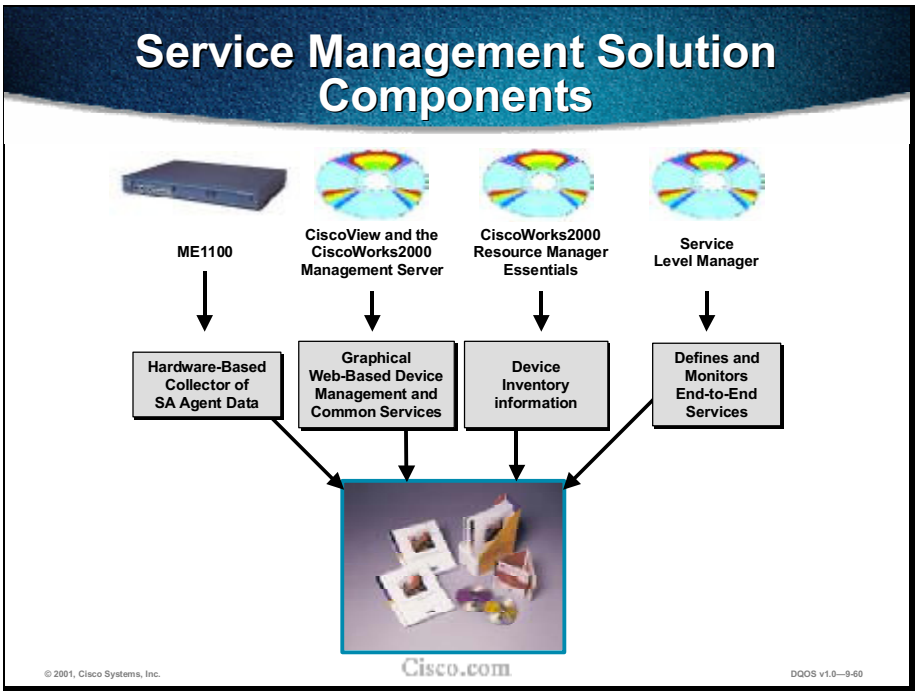


SMS is a valuable tool for enterprises to concisely define, monitor, and report on SLAs and a network's ability to support them. It integrates seamlessly with best-of-breed application and performance monitoring tools, providing unparalleled visibility into the causes of service level exceptions. It therefore protects the investments made by enterprises in their SLAs by validating that an enterprise is receiving the level of service for which it is paying, and it provides detailed reports showing ISP SLA performance.



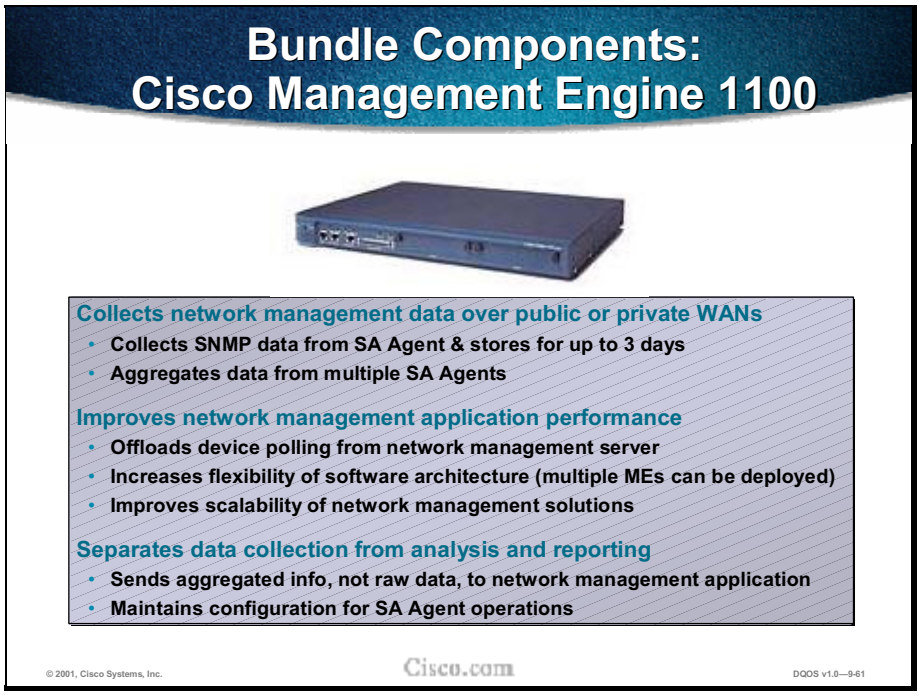
The end user accesses SMS via XML carried over HTTP. The SLM server communicates with Management Engines (ME1100s) and via XML/HTTP. This allows MEs (Management Engines) to be placed across WAN or on the other side of a firewall. MEs communicate with IOS SA Agent via SNMP, so they are best placed local to the routers on which they initiate and manage SA Agent (SAA) probes (although they do not need to be colocated).

SAA generates synthetic traffic probes to other SAA-enabled devices or IP end stations—for instance, jitter tests require an SAA responder in receiving device, while HTTP or ICMP Echo tests need only a web server or IP stack as a target.



There are four components of SMS. They are:

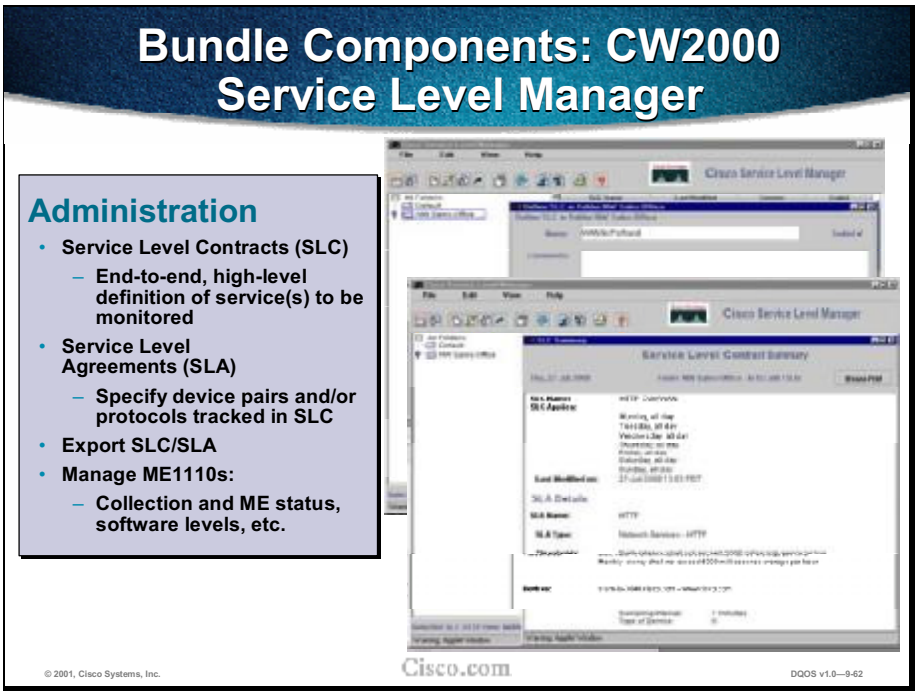
- ME1100—Hardware for collecting SA Agent data
- CiscoView and CiscoWorks2000 Management Server—Provides the graphical web-based device
- CiscoWorks2000 Resource Manager Essentials—Device inventory information
- Service Level Manager—Defines and monitors end-to-end services



CW2000 SLM Solution includes a management engine, the ME1110.

The Management Engine is a network component that provides scalability and distribution of network management functionality. Distribution can allow management applications to reach through firewalls or across NAT boundaries or simply localize network management traffic. The SLM server downloads its Java-based data collector to the ME1110 through an HTTP connection and retrieves data collection results through HTTP and XML. When the user configures SLC/SLA on the SLM server using a web browser, this request is passed on to the Management Engine. The Management Engine takes care of configuring the SA Agent using SNMP to ensure that the appropriate SLM metric data is gathered from the network elements. The SLM server periodically retrieves the results data from the Management Engine. The reports can then be viewed using a browser.

As time goes by and needs change, additional Management Engines can be deployed in the network to scale and/or localize management traffic.



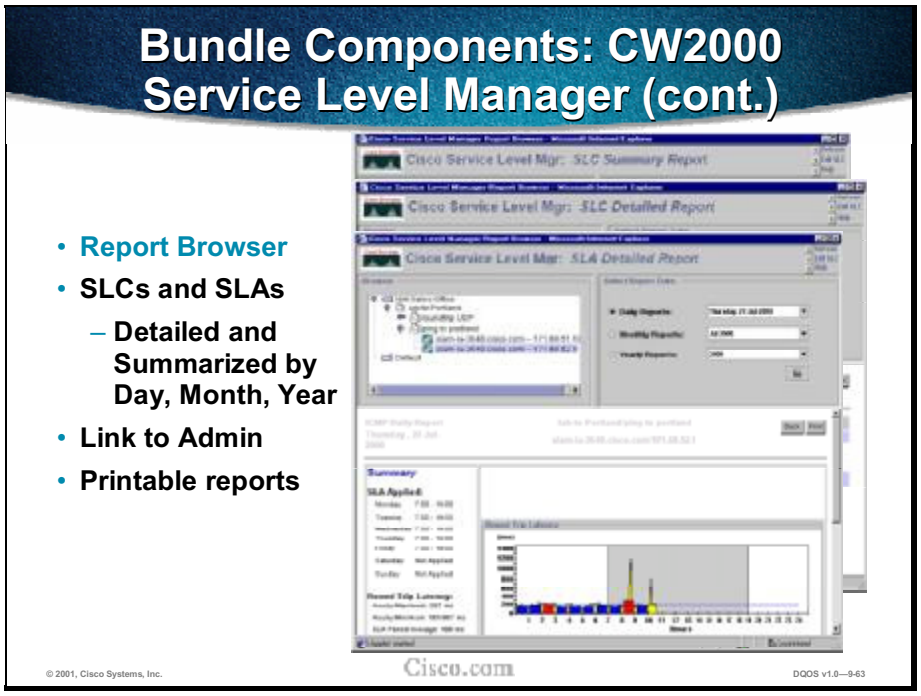
The Administration user interface allows users to define service level contracts and the SLA associated with them.

A key component is the Applies time, the time duration taken into account when violations and exceptions are determined. A response time may have been exceeded, but the SLA is not considered to be in violation if it is outside the Applies time.

SLA types supported in the first release are:

- **Latency** using UDP and ICMP; with the ability to set the ToS bit and change the payload size for the tests
- **Network** service, for looking at DNS resolve times and HTTP response times; for HTTP service, can see the time taken to do DNS lookup, TCP connection, and page download
- **VoIP,** where the forward and return jitter can be measured between two points in the network, with the ability to change the number of packets, payload, and interpacket gap

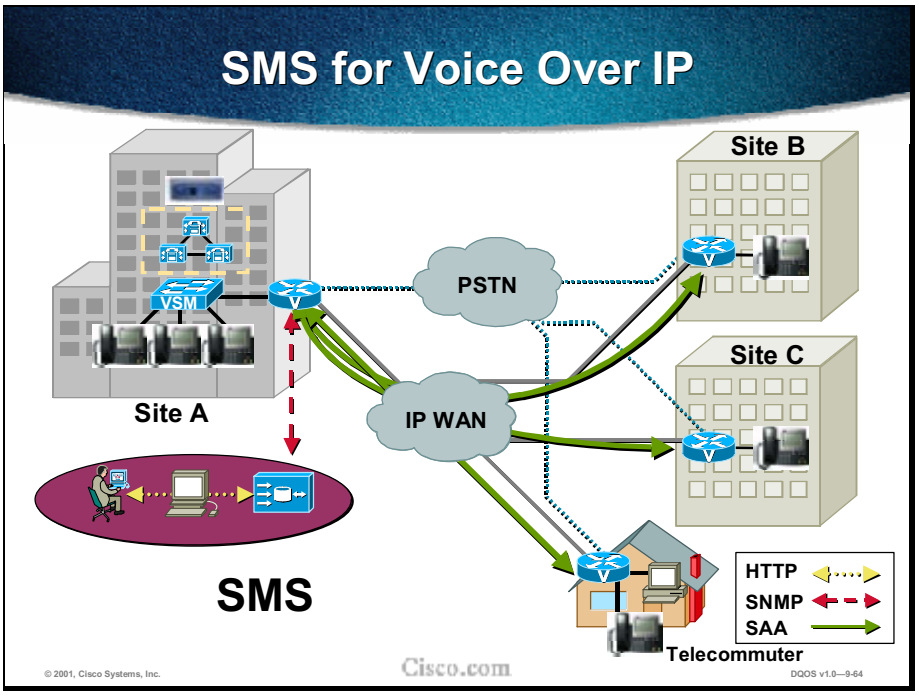
The export SLC/SLA functionality allows user to exchange configurations between multiple SLM systems.



The reports navigator provides a way to select reports at the SLC and SLA summary and details levels.

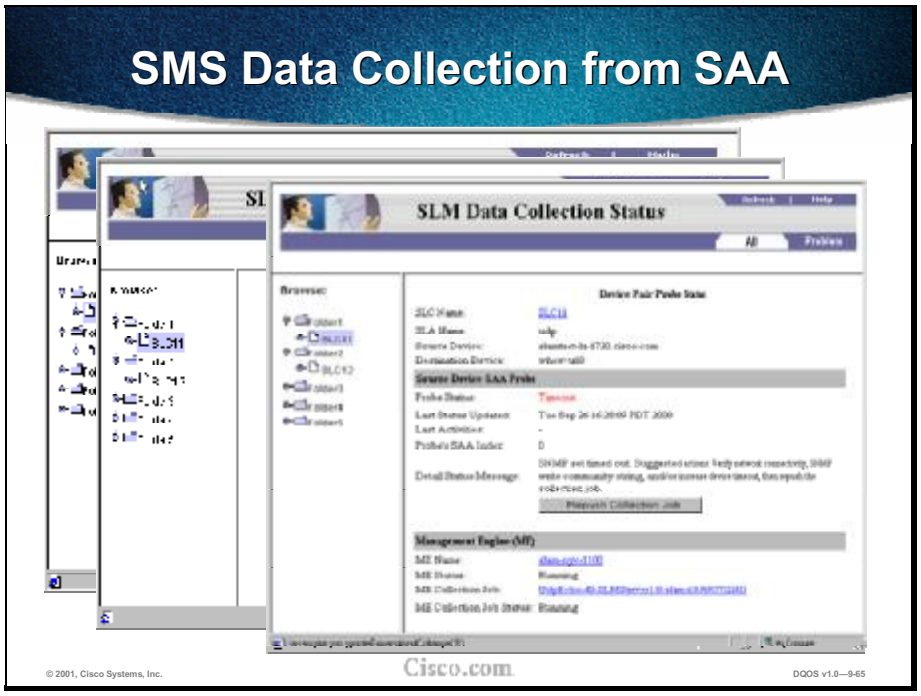
The summary reports are designed to help management see at a glance how well SLAs are being conformed to in general. The summary reports show the percentage of SLAs that have been violated or are in an exception state.

The network operators can use the summary reports to drill-down on the hour-by-hour detailed reports for further analysis.

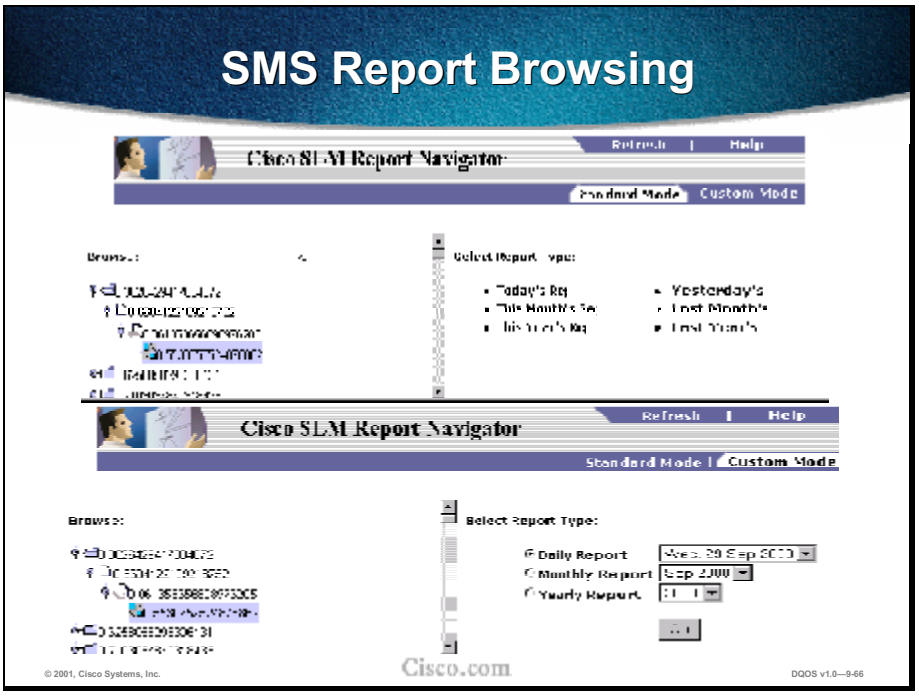


The SMS application is able to monitor Voice over IP. SMS can be initiated at a central site customer premises equipment (CPE) and monitor remote CPE routers. It can monitor VoIP SLAs through forward and backward jitter tests.

- RTR responder activated in CPE edge routers

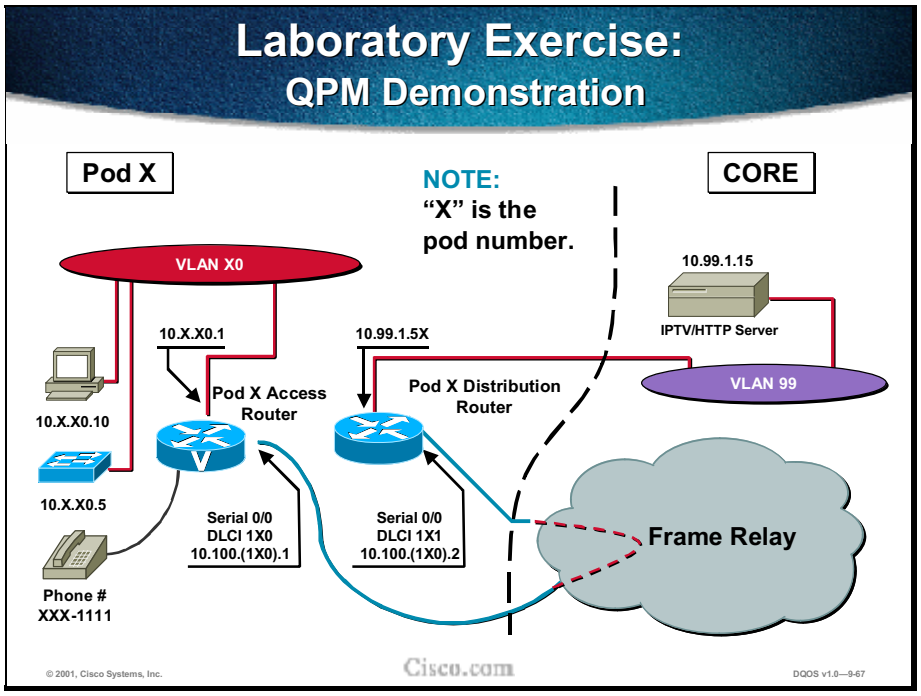


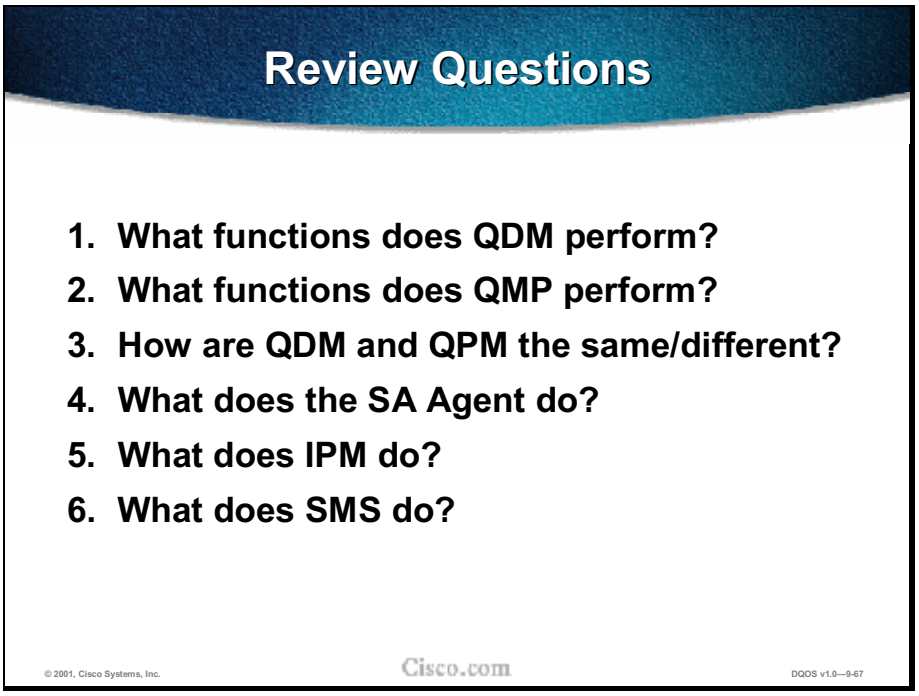
SMS provides improved views into the status and progress of SLA data collections. The All and Problems tabs allow the network administrator greater focus on key information and the ability to drill-down to the specific problem.



Standard report times provide faster access to most commonly used timeframes.

The All and Exception Only tabs provide greater focus on key information.


Laboratory Exercise:
QPM Demonstration
Pod X
CORE
NOTE:
"X" is the
pod number.
VLAN X0
10.99.1.15
IPTV/HTTP Server
10.X.X0.1
10.99.1.5X
Pod X Access
Router
Pod X Distribution
Router
VLAN 99
10.X.X0.10
10.X.X0.5
Serial 0/0
DLCI 1X0
10.100.(1X0).1
Serial 0/0
DLCI 1X1
10.100.(1X0).2
Frame Relay
Phone #
XXX-1111
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—9-67



Answer these questions:

1. What functions does QDM perform?
2. What functions does QMP perform?
3. How are QDM and QPM the same/different?
4. What does the SA Agent do? [Monitors network performance, network resources, and applications by measuring response times and availability]
5. What does IPM do?
6. What does SMS do?

Answers to the review questions appear in Appendix B.

Summary

Upon completing this module, you should be able to:

- **Utilize QoS Device Manager to monitor performance, establish baselines, and configure QoS policies**
- **Utilize QoS Policy Manager to configure advanced QoS policies, scale policy deployment, upload/verify/roll back policies, and deploy QoS policies by external time-based/event-based scripts**
- **Configure Cisco Service Assurance Agent to measure key SLA metrics and monitor network performance between local and remote devices**
- **Monitor and troubleshoot network performance with IPM and SMS**

10

QoS Design

Overview

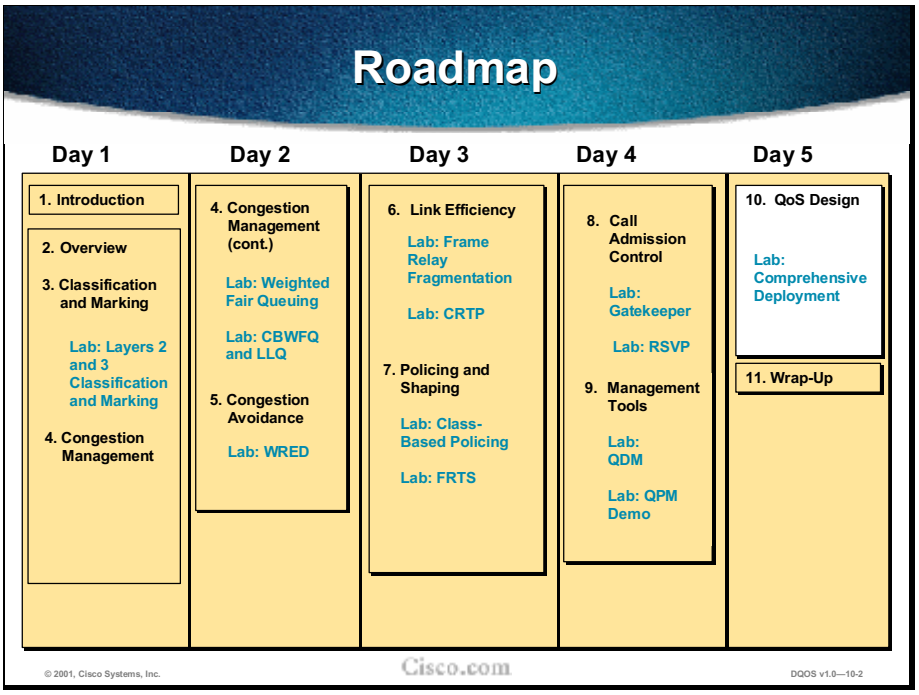
This chapter discusses the different quality of service (QoS) mechanisms explained in the prior chapters. It provides the recommended steps in deploying QoS and design guidelines for voice, video, and mission-critical traffic.

Objectives

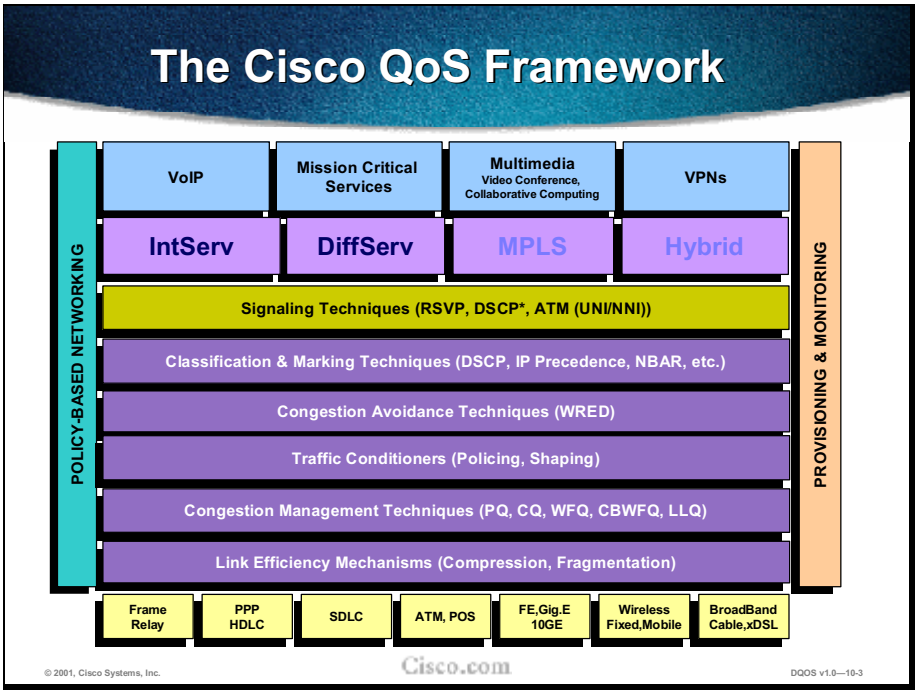
Upon completing this chapter, you will be able to:

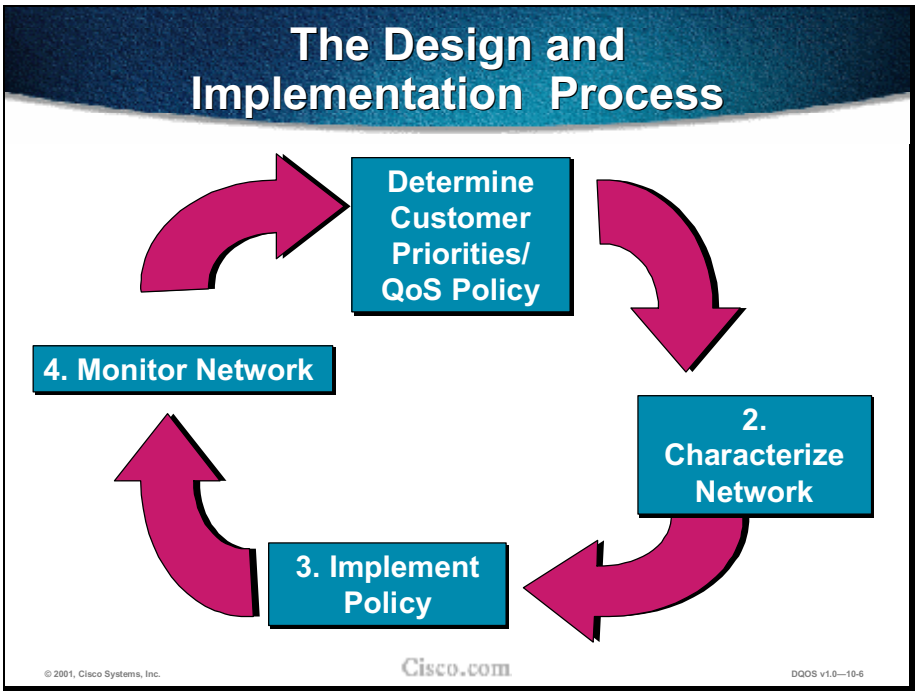
- Design a converged multiservice network to provide proper QoS for voice, video, and data traffic

Outline



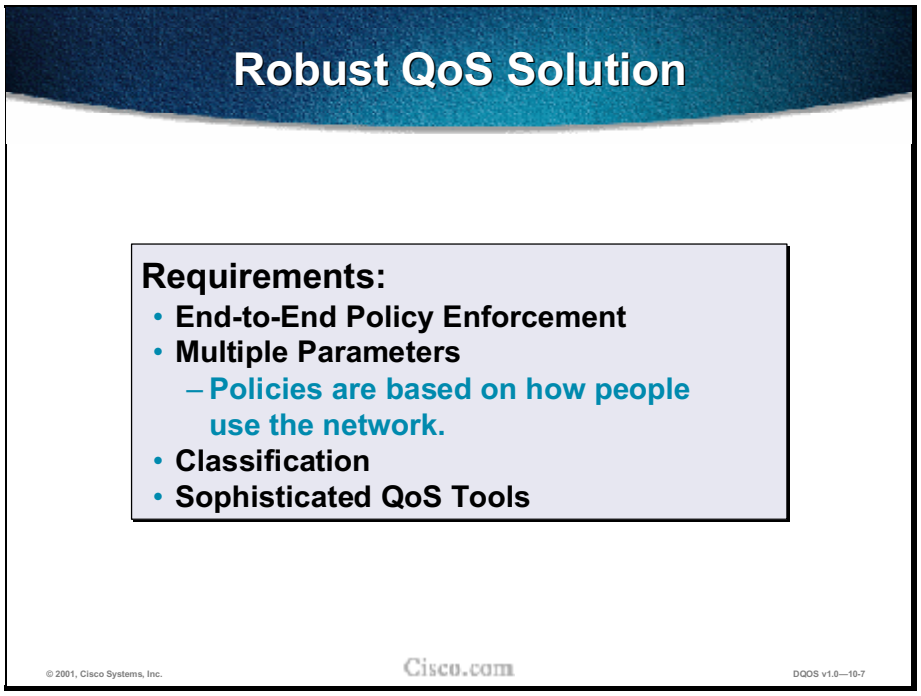
The figure shows the plan for the week. This chapter, QoS Design, contains a lab in which you will design and deploy a comprehensive QoS solution for a hypothetical company.





In the design process, the network administrator must:

1. Determine customer priorities/QoS policies
   - Find out what traffic is important to the customer and the service level desired for that traffic
2. Characterize the network traffic
   - Conduct analyses to collect information about current data, voice and/or video usage
3. Implement network policies
   - Define ACL
   - Define classes
   - Define policies for classes
4. Monitor network
   - Collect data on network performance; adjust as necessary
   - Use management tools: QPM, QDM, SA Agent, IPM



What characteristics does a robust QoS solution have?

- End-to-end policy enforcement—QoS must be applied end-to-end. Consequently, it must be platform, device, and media independent, operating at Layer 3 and above to ensure end-to-end functionality across multiple network devices (such as routers, switches, firewalls, access servers, and gateways) and link layers (for example, ATM, Frame Relay, or Ethernet).

- Multiple parameters—Policies are based on how people use the network. Devices must have the flexibility to apply and enforce QoS based on several parameters that can closely reflect network managers' defined policy. These parameters distinguish traffic flows based on IP or MAC address, application, user, time of day, or location within the network.

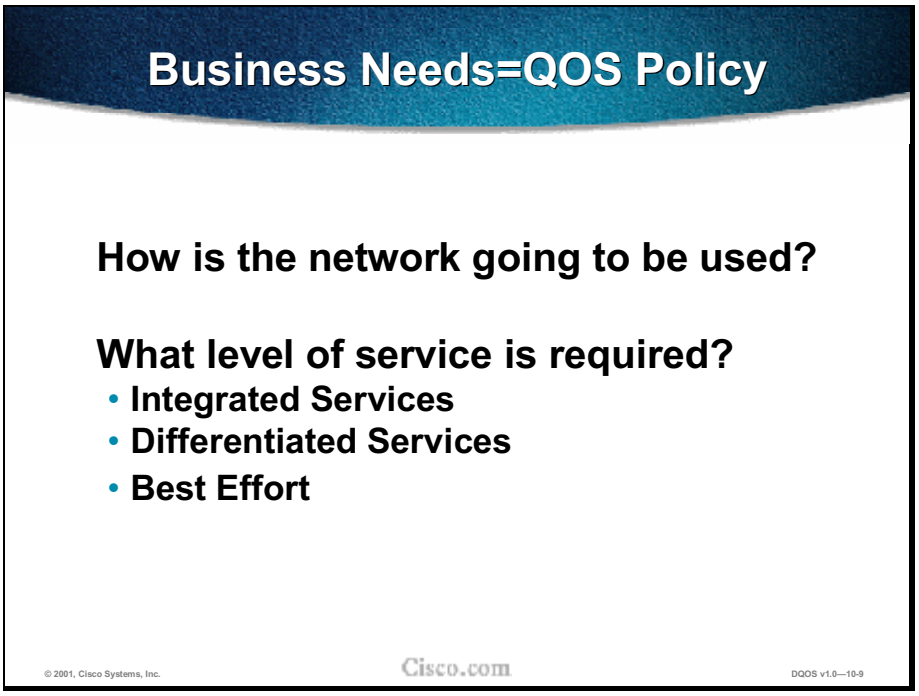
  Administrators define policies using a combination of these parameters. For instance, a simple policy might read, "No Webcast traffic allowed across a certain 56-K link to Branch Office A." A more sophisticated policy might read, "Allow video conference traffic within a defined LAN segment on Mondays between 3 and 5 P.M. for users A, B, C, and D only. At all other times and for all other users, disallow video conference."

- Classification—By definition, administrators need the ability to set different QoS levels for traffic as defined by the policy.

- Centralized control—A network-based policy enforcement more often results in a consistent policy deployment and enforcement.

- Sophisticated QoS tools—Because there are so many different network elements and so many parameters required to successfully deploy and implement a QoS policy end-to-end, the associated sets of QoS tools are necessarily complex. They must be fully featured to enable network managers to build the intelligent networks they need.


The Design Process:
Step 1—Determine
Priorities/Policy
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—10-8



Policies are based on how people use the network. Devices must have the flexibility to apply and enforce QoS based on several parameters that can closely reflect network managers' defined policy. These parameters distinguish traffic flows based on IP or MAC address, application, user, time of day, or location within the network.

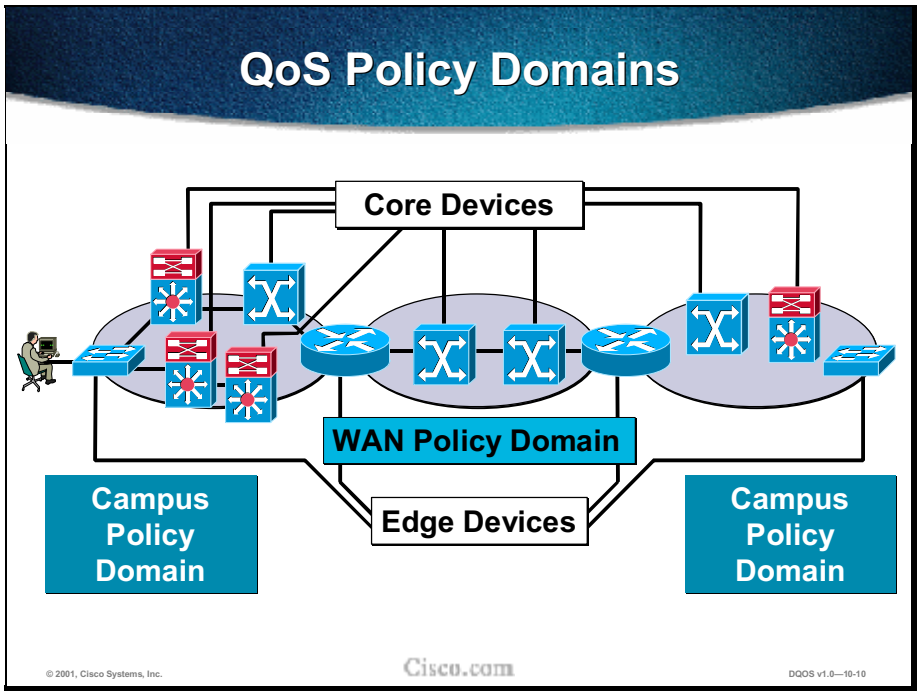
Administrators define policies using a combination of these parameters. For instance, a simple policy might read, "No Webcast traffic allowed across a certain 56-K link to branch office A." A more sophisticated policy might read, "Allow video conference traffic within a defined LAN segment on Mondays between 3 and 5 P.M. for users A, B, C, and D only. At all other times and for all other users, disallow video conference."

Which type of service is appropriate to deploy in the network ? Is it integrated services? Differentiated services? Or best effort? This depends on several factors:

- The application or problem the customer is trying to solve: Each of the three types of service is appropriate for certain applications. This does not imply that a customer must migrate to differentiated and then to guaranteed service (although many probably eventually will). A differentiated service—or even best-effort service—may be appropriate depending on the customer application requirements.

- The rate at which customers can realistically upgrade their infrastructures. There is a natural upgrade path from the technology needed to provide differentiated services to that needed to provide integrated services, which is a superset of that needed for differentiated services.

- The cost of implementing and deploying integrated service is likely to be more than that for a differentiated service.

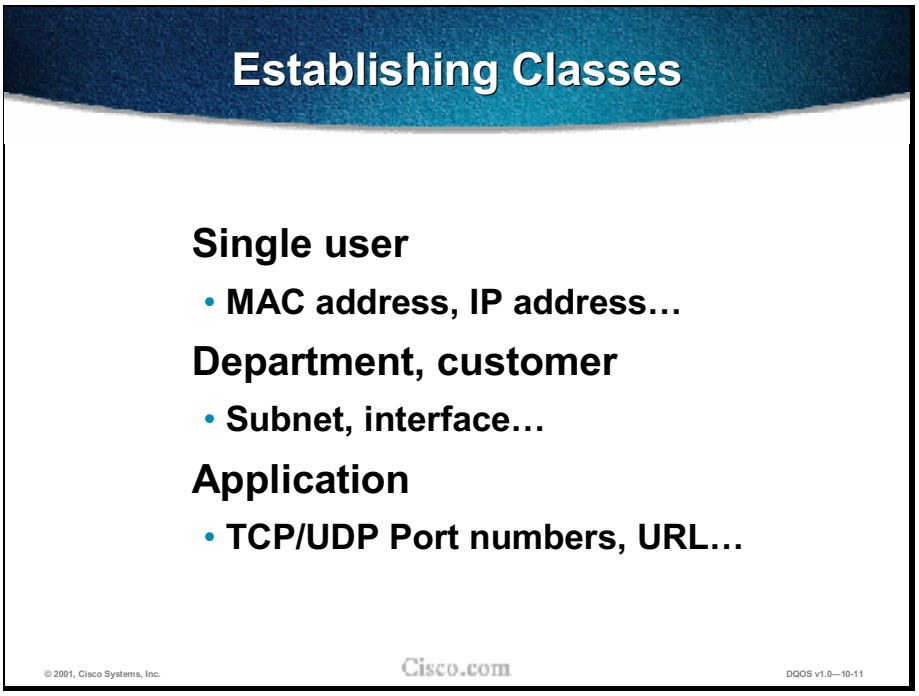


Administrators define QoS policy domains within the network to indicate which devices take up which tasks. The most common instance of a policy domain is a demarcation between a LAN and a WAN. The device (usually a router or access server) that links the LAN and WAN becomes a QoS policy domain-edge device, while those within the LAN and WAN backbones are core devices.

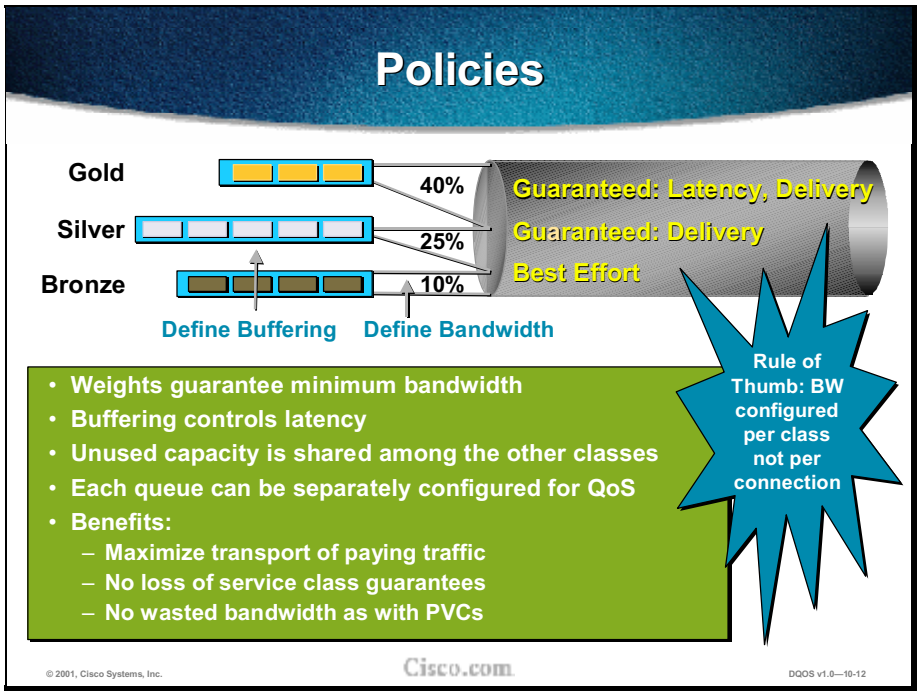
Within Cisco's QoS solution, the division of tasks is as follows: Edge devices handle the required activities to monitor, recognize, and classify traffic. Core devices are optimized for performance to expedite transport of high-priority traffic without congestion or delay.

QoS edge devices are responsible for:

- Admission control—Permits/denies application traffic
- Classification—Sets a priority level (high/medium/low)
- Proxy signaling—Signals IP Precedence/differentiated services or IP RSVP bandwidth allocation request along the delivery path
- Policing—Verifies that only the allocated bandwidth is used
- Traffic shaping—For reactive congestion management
- NBAR—For application recognition



When categorizing traffic into classes, it is important to note that, while categorization based on application type may be the most intuitive method, there are other ways to split traffic into classes. Some of those are listed in this figure.



As an example, let's take a closer look at class-based weighted fair queuing (CBWFQ) and see how this can be used to guarantee service levels and maximize bandwidth utilization.

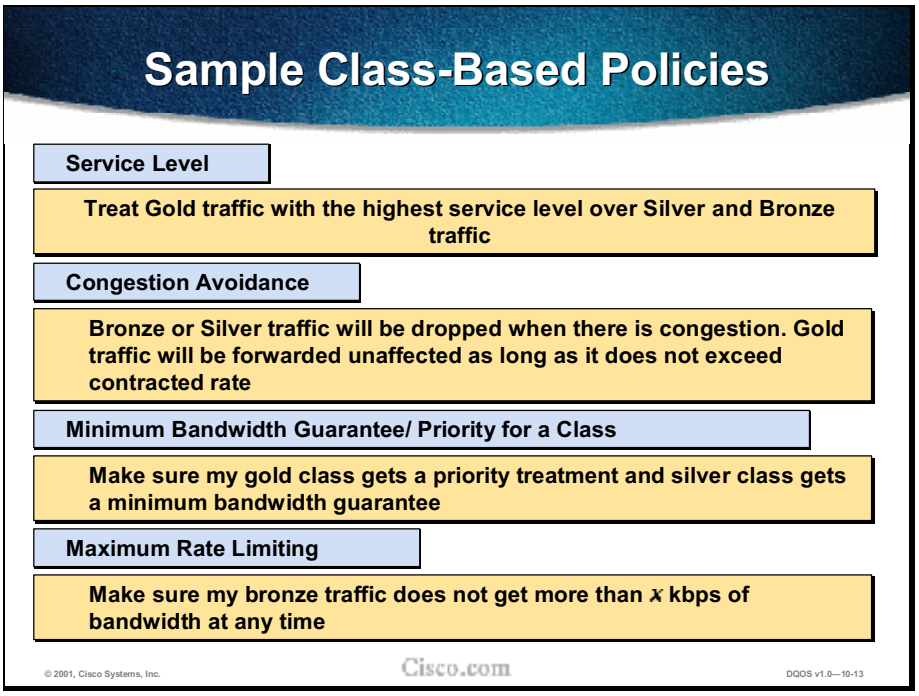
In this example, we have defined three service classes:

- Gold, with guaranteed latency and delivery
- Silver, with guaranteed delivery
- Bronze, a best-effort service

Bandwidth is configured per class, not per connection.

By separately allocating bandwidth and buffering space, we can tailor each class to the specific service needs. For example, the gold class could be used for voice traffic: A large bandwidth allocation ensures that sufficient bandwidth is available for all the cells in the voice queue, while a moderately sized buffer limits the potential cell delay. Since these shares are relative weights, allocating a large share to gold means that a minimum is guaranteed; if the gold class is underutilized, the bandwidth will be shared by the remaining classes in proportion to their weights. This ensures maximum efficiency and ensures that paying customer traffic will be sent if bandwidth is available.

Also, since Cisco's IP+ATM switches implement separate queues for each class (as shown on the previous figure), each service queue can be separately configured for ATM QoS. For example, ABR could be enabled for the silver class.



These are some key policies that need to be implemented within a network designed using a class-based approach.

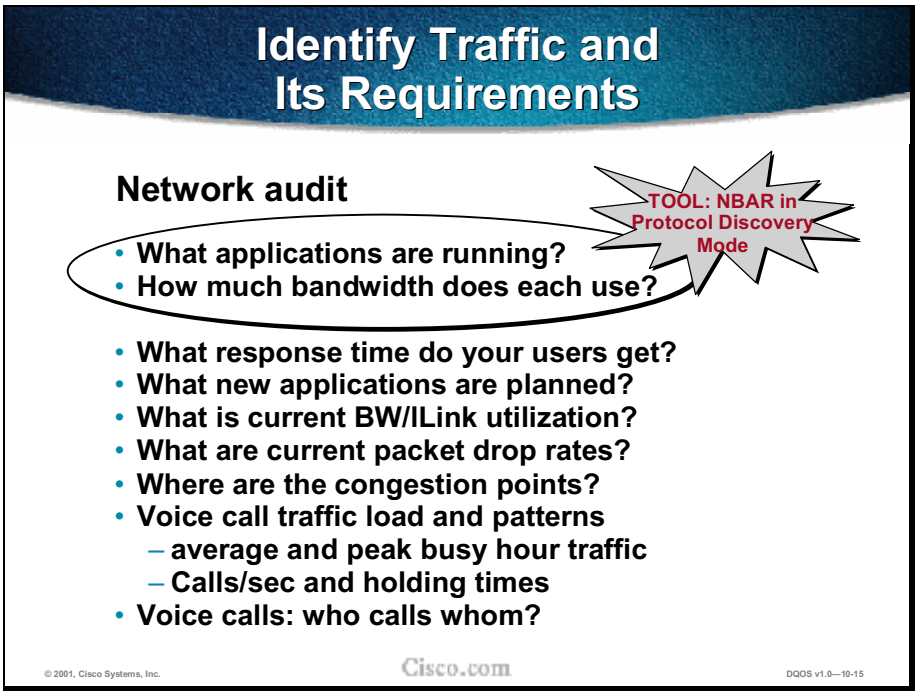
The gold traffic class is that portion of traffic that is extremely sensitive to delay. Voice over IP is a common gold-class application.

Congestion avoidance—When traffic volumes grow and a network starts to get congested, it must still give preference to high-priority flows, while encouraging low-priority flows to slow down. Defined by a QoS policy, a congestion strategy keeps throughput high and packet loss low.

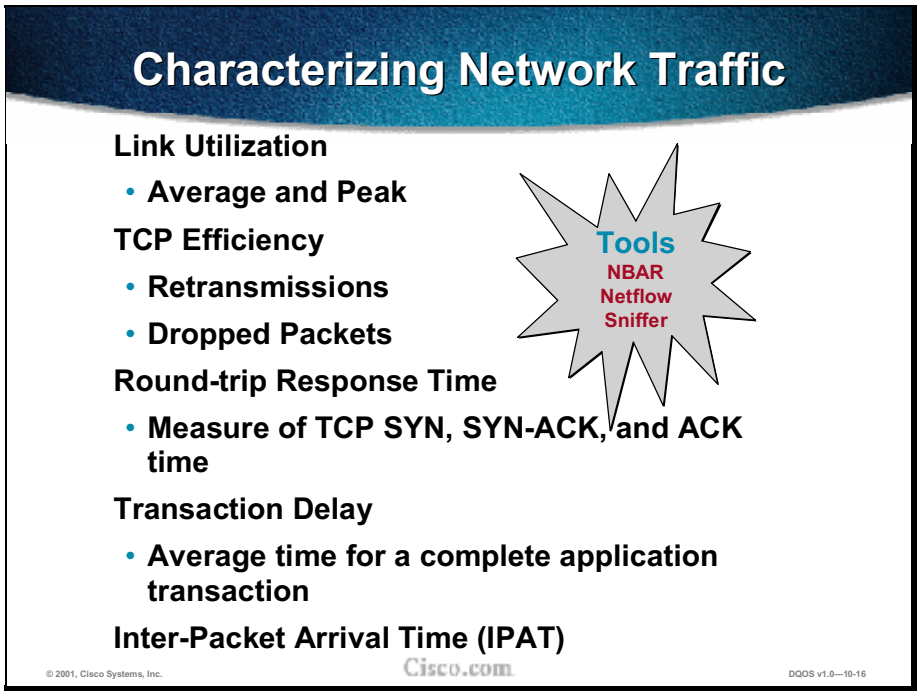
Minimum bandwidth guarantee/priority for a class—Traffic-limiting policies can define an upper range on the bandwidth allocated to selected traffic. Thus, the custom queue defines a minimum bandwidth, and the limiting policy defines an upper limit.

Maximum rate-limiting—A commonly overlooked policy in QoS design is to put a limit on the maximum limit on the best effort traffic in the network.

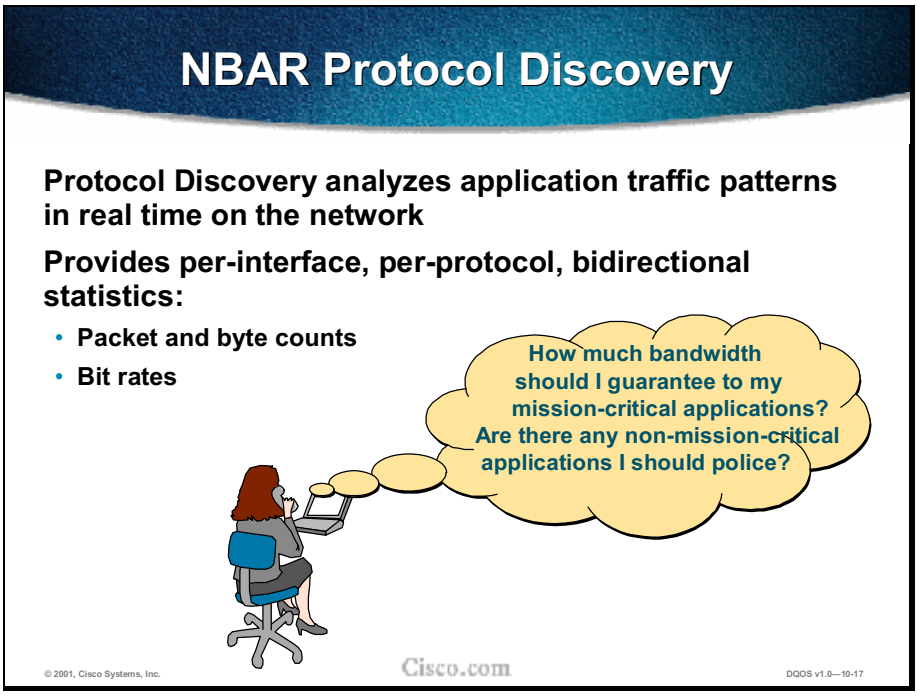

The Design Process: Step 2—
Characterize the Network
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—10-14



In the figure is a set of questions that must be answered in conducting a network audit.

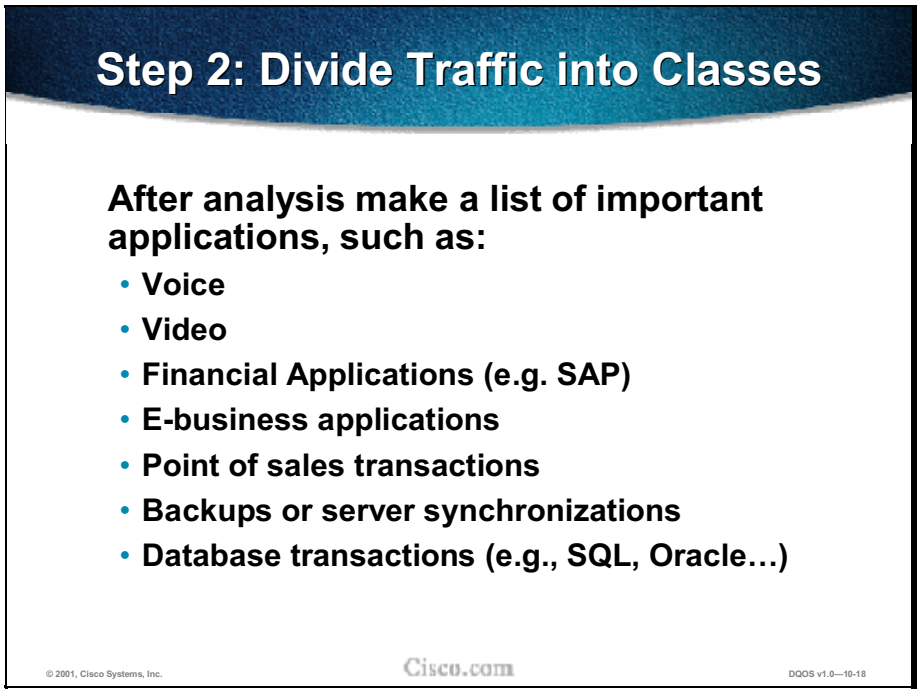


Characteristics of data traffic play a crucial role in performance analysis and design of communication networks. As the figure shows, there are several parameters determining an applications classification.



In Chapter 3, Classification and Marking, NBAR was looked at in some detail. NBAR includes a Protocol Discovery feature. Protocol discovery shows the mix of applications currently running on the network. This helps define QoS classes and polices, such as how much bandwidth to provide to mission-critical applications, and helps determine which protocols should be policed. The following per-protocol, bidirectional statistics are available:

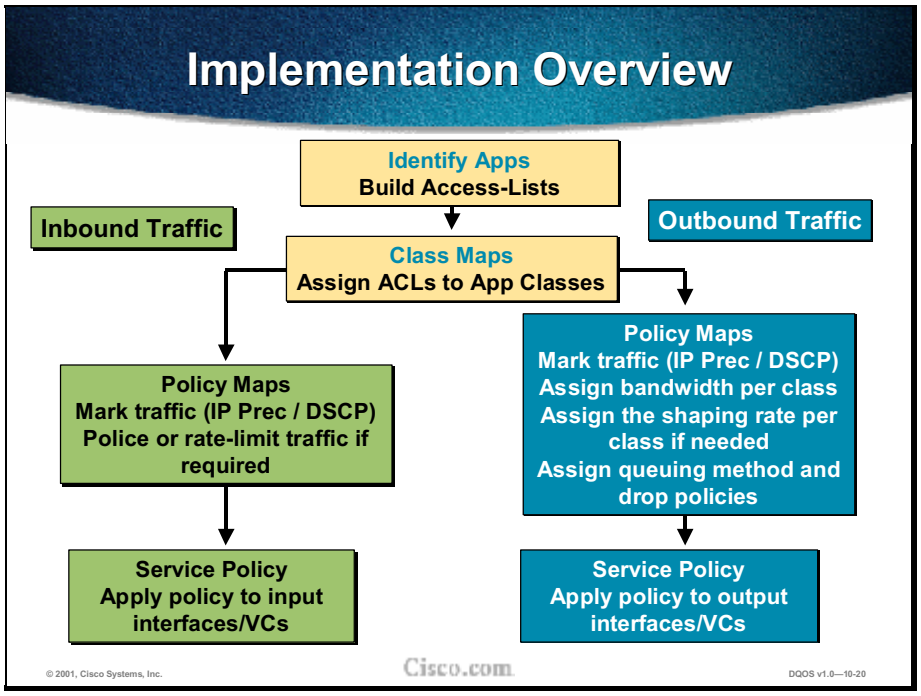
- Total number of input and output packets and bytes
- Input and output bit rates



After developing a list of all traffic types on the network (or planned for the network), the traffic types must be ordered in terms of their relative priority. This process is known as classification.

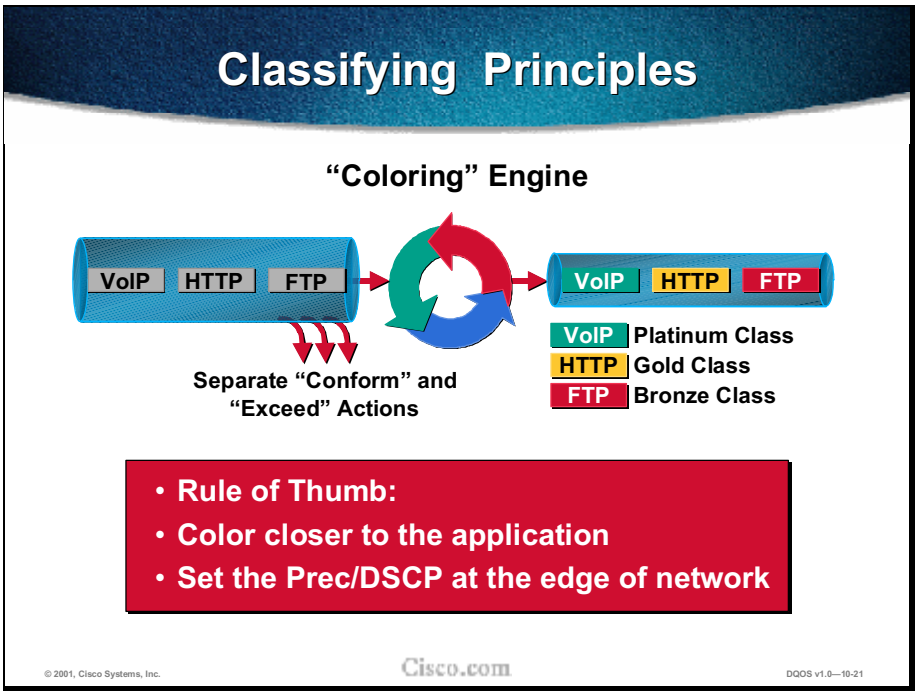
Some customers find it useful to build a matrix that includes:

- Application name
- Identification method (IP address, UDP/TCP ports, host, subnet)
- Value of the identification method
- Service class assigned



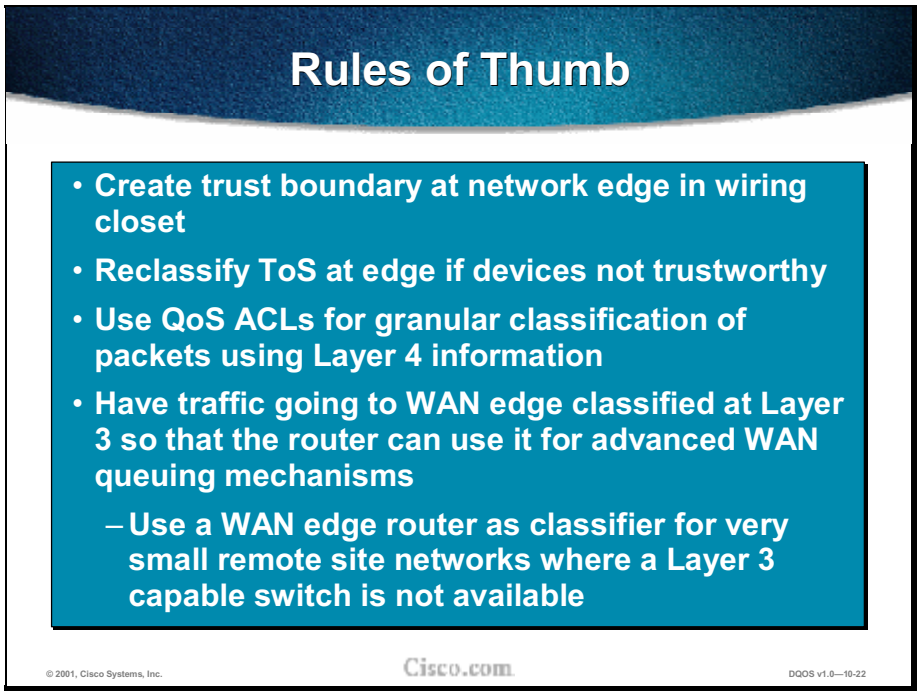
The basic goal of policies is to ensure that network bandwidth is used efficiently by working with QoS features to provide:

- Guaranteed bandwidth/low delay
- Bandwidth limits via policing
- Traffic shaping
- Packet coloring

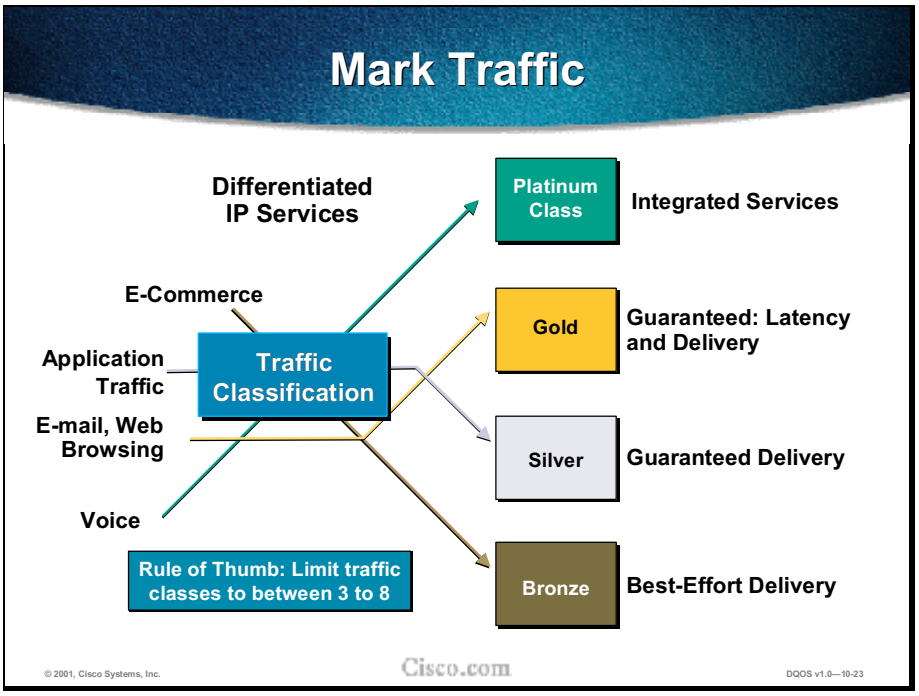


Classification, or "coloring," marks a packet or flow with a specific priority. This marking establishes a trust boundary that must be enforced.

Classification should take place at the network edge, typically in the wiring closet or within the IP phones or voice endpoints themselves. Packets can be marked as important by using Layer 2 class of service (CoS) settings in the User Priority bits of the 802.1p portion of the 802.1Q header or the IP Precedence/DSCP bits in the type of service (ToS) byte of the IPv4 header.



The notion of trusting or not trusting forms the basis for the trust boundary. Ideally, classification should be done as close to the source as possible. If the end device is capable of performing this function, the trust boundary for the network is at the access layer in the wiring closet. If the device is not capable of performing this function, or the wiring closet switch does not trust the classification done by the end device, the trust boundary may shift. How this shift happens depends on the capabilities of the switch in the wiring closet. If the switch can reclassify the packets, then the trust boundary remains in the wiring closet. If the switch cannot perform this function, then the task falls to other devices in the network going toward the backbone. In this case, the rule of thumb is to perform reclassification at the distribution layer. This means that the trust boundary has shifted to the distribution layer. It is more than likely that there is a high-end switch in the distribution layer with features to support this function. If possible, try to avoid performing this function in the core of the network.

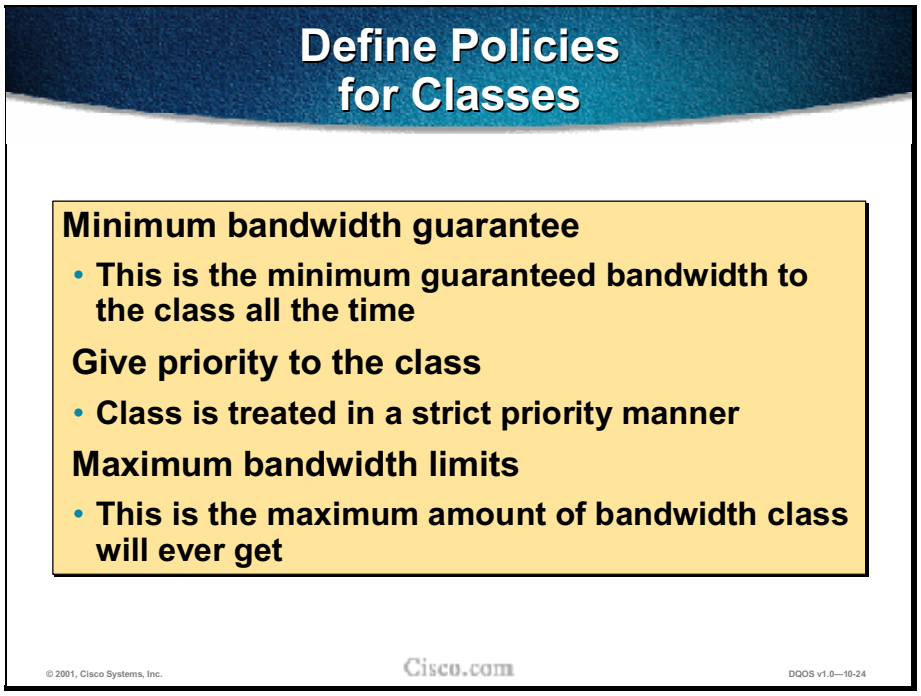


A good network design needs to keep traffic classification simple, and most networks will never go over three to eight service classes. It would be difficult to differentiate and successfully implement 64 or 120 different service classes.

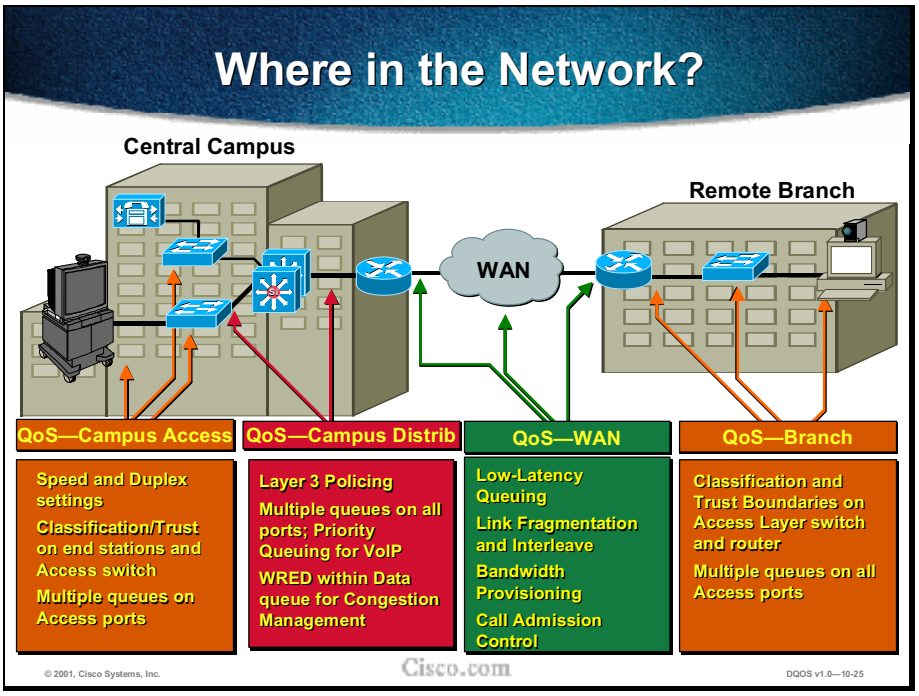
A gold service would guarantee latency and delivery for the transport of mission-critical business applications like packet telephony or SNA.

The silver class would guarantee delivery and be used for more general applications that are not as sensitive to delay like e-commerce.

The bronze class could be used to support small business and e-mail and other best-effort applications.



Once traffic has been classified into various classes, the next step in the design process is assigning policies to these classes. Policies include minimum and maximum amount of bandwidth that should be made available to a class and how to treat the class of traffic if congestion were to occur. All this can be implemented within a Cisco environment using priority levels assigned to the class of traffic.

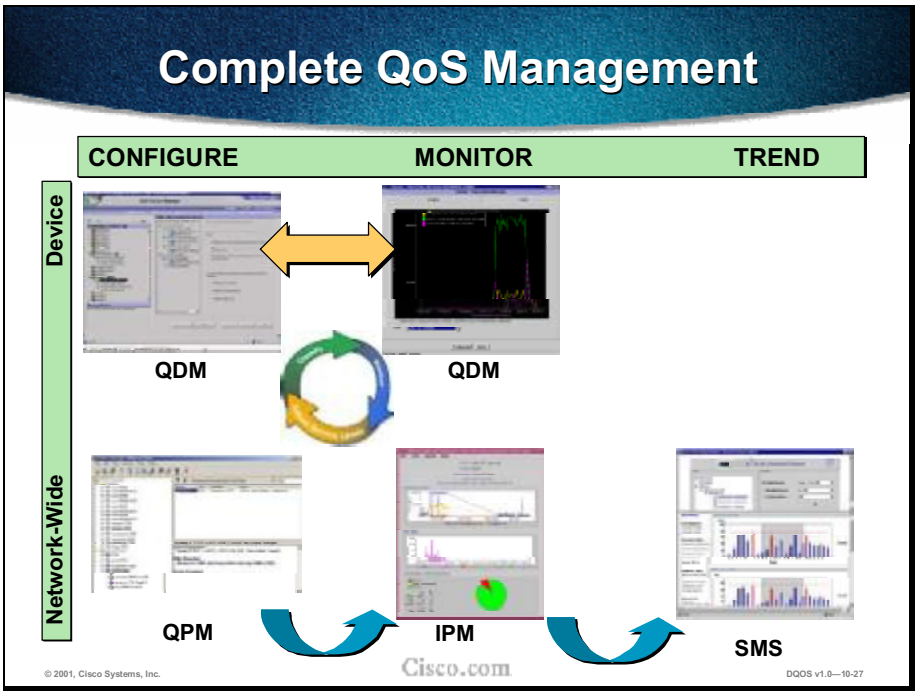


As the figure indicates, packets need to be marked as close to the source as possible. Admission control needs to be performed at the WAN ingress. Within the WAN, link efficiency and traffic conditioning tools may be deployed as required. Congestion avoidance and congestion management tools are used throughout the network.

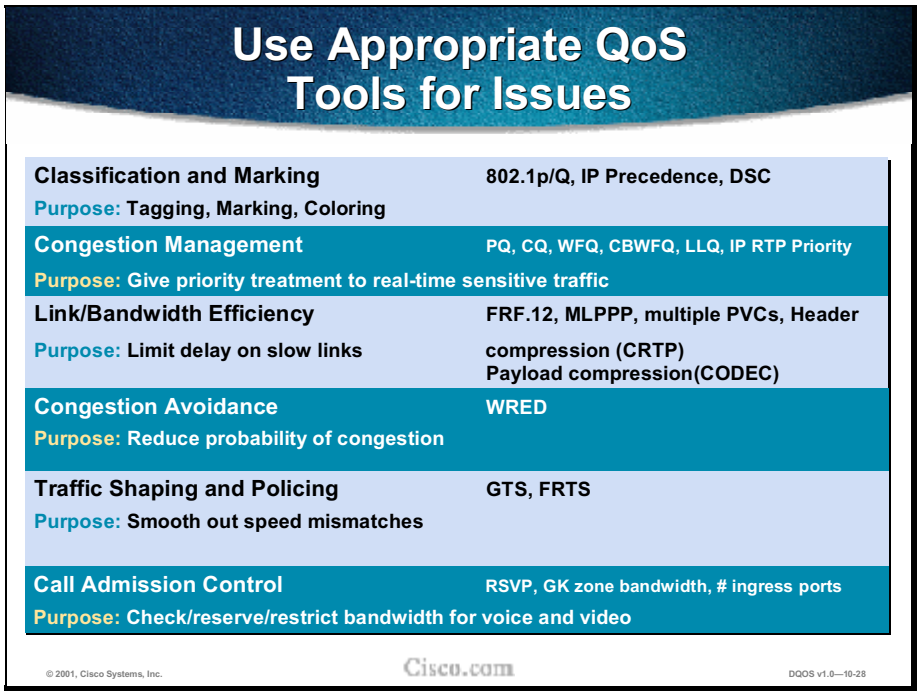
In order for QoS methods to be used within the network, traffic must be classified into higher and lower priorities. Each classification must then be marked so the network knows which QoS methods to apply. This process is completed at the ingress points to the network. Queuing and shaping methods can then be applied throughout the network.

The classification and marking work is usually done at the edge of the network, where speeds are lower. This is because it can be more CPU and memory intense. In general, at the edge we can use relatively complex access lists, flows, and other techniques to recognize traffic. In the network core, where speeds are higher, we keep things simpler, by using marked packets (simpler lookups) and CoS (several major categories of traffic rather than per-application or per-flow handling).

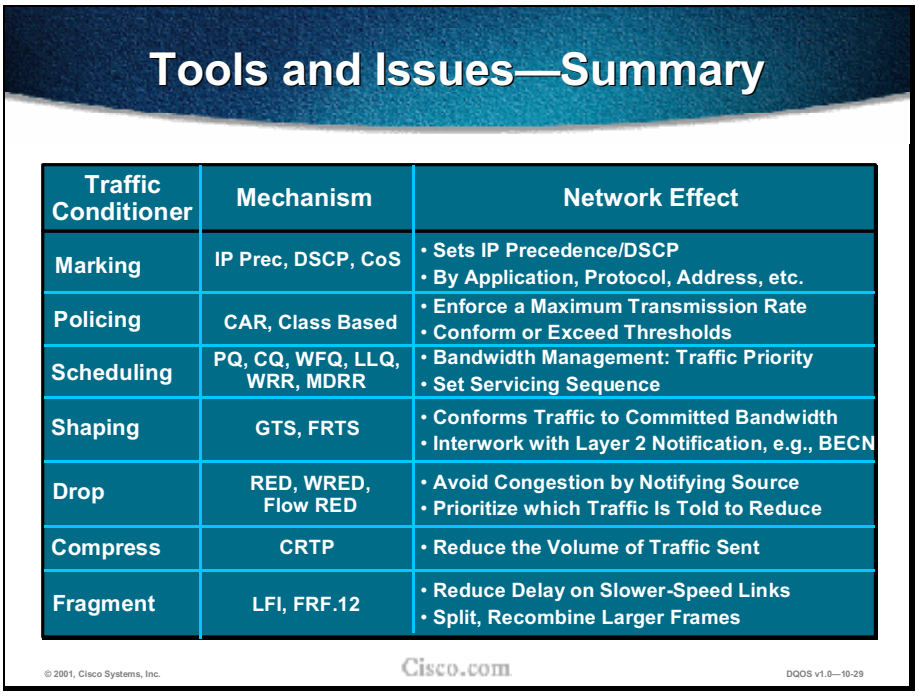

The Design Process:
Step 4—Monitor
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—10-26



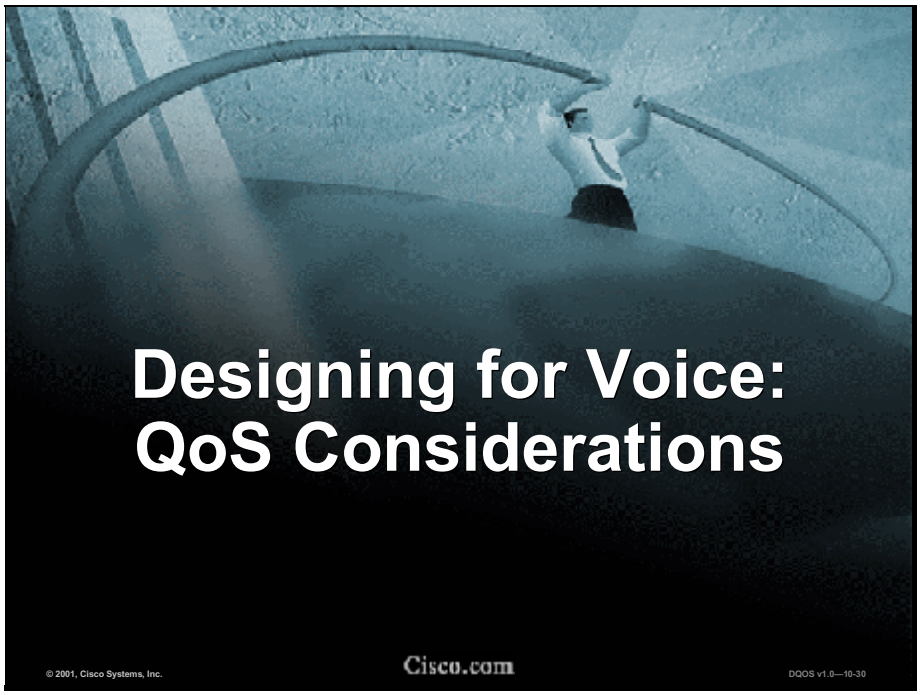
Setting QoS policies is not sufficient for successful deployment of QoS: A network administrator must monitor network traffic before and after policies are set to ensure that the desired effects are indeed occurring. Furthermore, long-term trending is also essential to ensure that as the enterprise grows and changes, service levels are still being met.

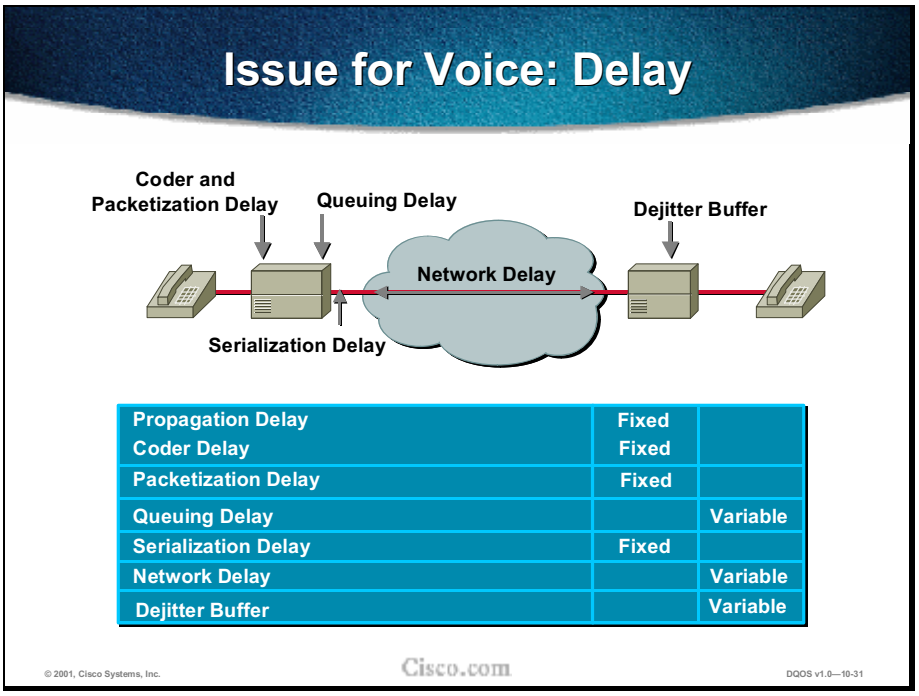


In monitoring end-to-end QoS, network administrators may need to adjust policies to provide optimal performance. The figure above and on the next page summarizes some of the more common quality issues.

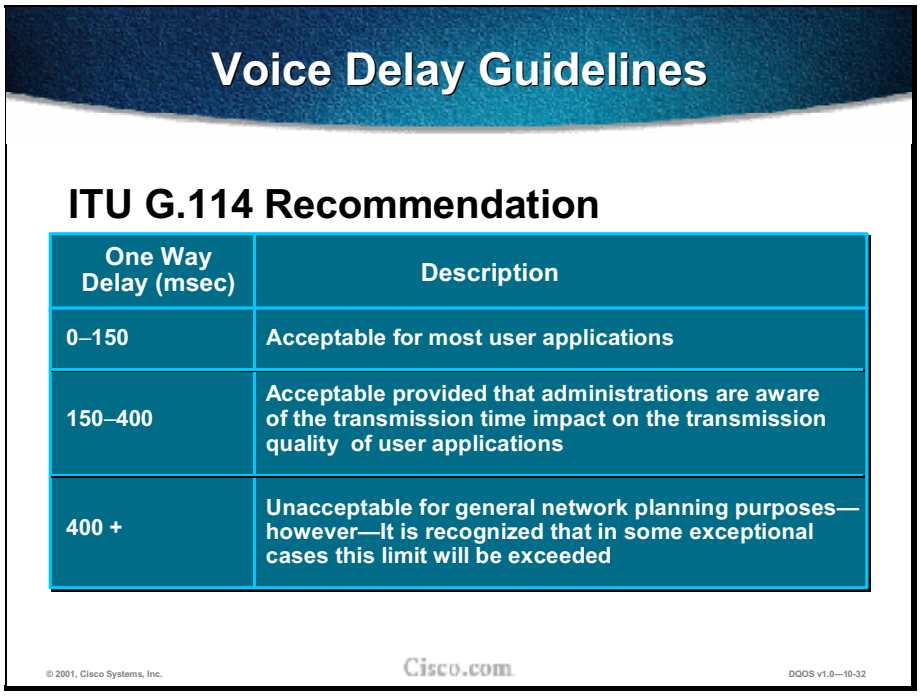


The above table summarizes the various QoS tools, as well as their functions and effects.



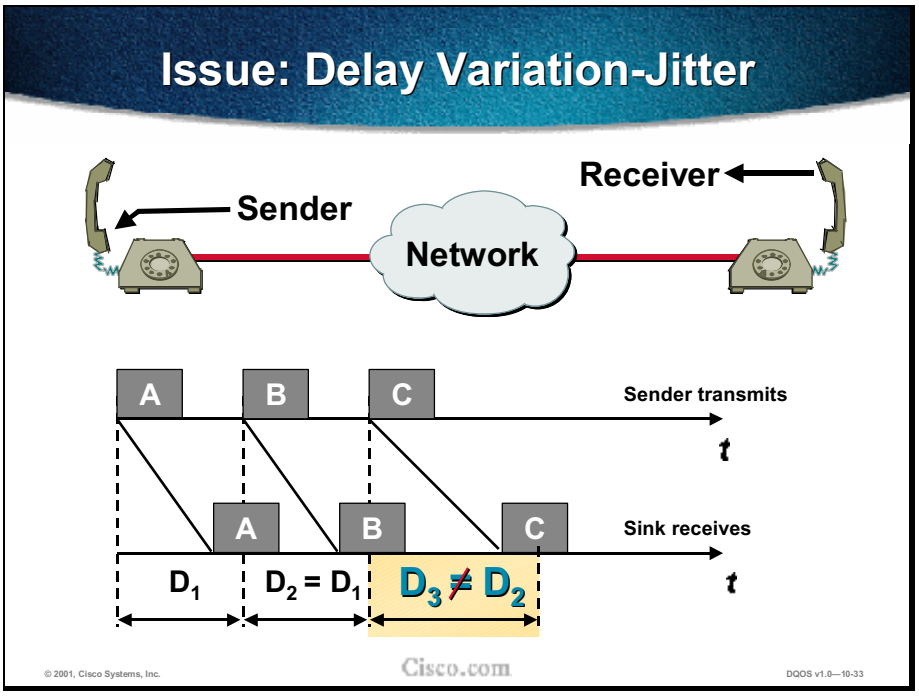


The figure illustrates the various components that make up one-way or round-trip delay. There are some fixed components associated with the overall delay, such as propagation delay, CODEC (if used) delay, packetization delay, and serialization delay. In addition, there are some variable delays associated with packet transport that can be optimized using proper design techniques. These delays include queuing, network, and dejitter (if used) delays.

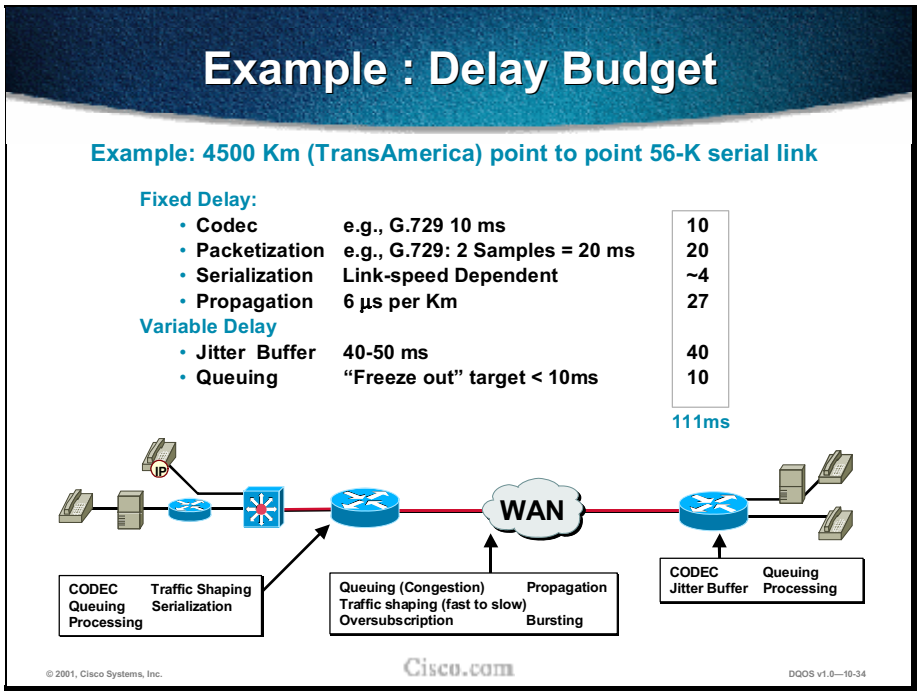


How much delay is too much? This chart is the ITU recommendation.

When designing any voice solutions, target a one-way delay budget of 150 milliseconds. This is a practical number. You may find that your customers are more, or less, tolerant, depending on your existing environment and network goals.



Congested egress queues and serialization delays on network interfaces can cause variable packet delays. Without priority or low latency queuing (LLQ), queuing delay times equal serialization delay times as link utilization approaches 100 percent. Serialization delay is a constant function of link speed and packet size. The larger the packet and the slower the link clocking speed, the greater the serialization delay. While this is a known ratio, it can be considered variable because a larger data packet can at any time enter the egress queue before a voice packet. If the voice packet must wait for the data packet to serialize, the delay incurred by the voice packet is its own serialization delay plus the serialization delay of the data packet in front of it. Using Cisco LFI techniques, serialization delay can be configured to be a constant delay value.

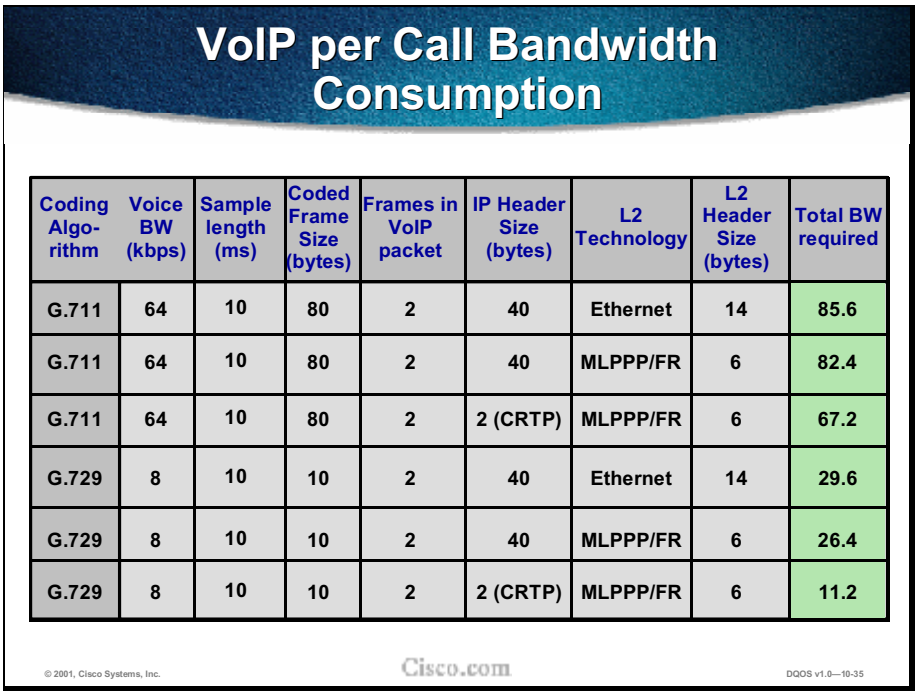


To calculate a delay budget, you must look at the fixed and variable delay components discussed so far.

In summary, fixed delay includes propagation, serialization, and processing delays.

Variable delay includes queuing delay and the dejitter buffer delay.

The coder delay for G.729 voice compression is 10 milliseconds (ms) per side. Packetization delay is the rate at which a packet is filled. This is typically governed by the speed at which voice samples are played out. Queuing delay is variable and is cumulative based on the number of devices in the network. Serialization delay is the time it takes to play out the voice packet(s) onto the 56-K trunk. The propagation delay is calculated using a 6-ms-per-km conversion. Assuming that you have your adaptable dejitter buffer set to a nominal playout value of 40 to 50 ms, you induce that much delay on the playout. Thus you have 61 ms of fixed delay and 50 ms of variable delay, getting you to 111 ms plus the network delay from the network. This puts you in good shape for being likely to deliver quality voice calls over this setup.

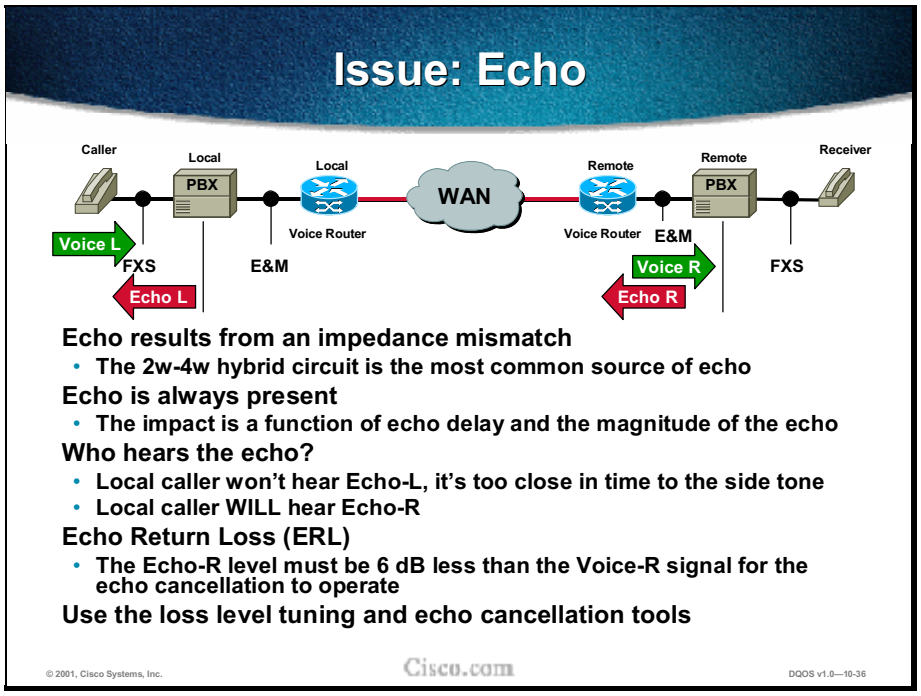


Bandwidth planning considerations are a must when running compressed voice across the WAN.

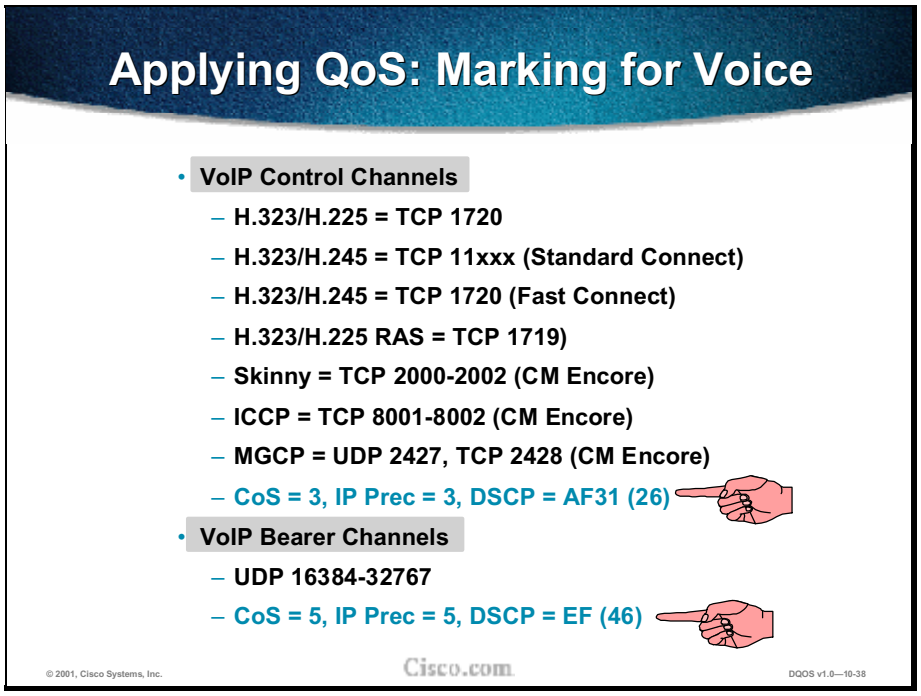
Compressed voice, for example, G.729, consumes only 8 Kb from a voice payload perspective. This is an 8-to-1 savings from traditional standard G.711 (64-Kb PCM transport model).

However, for every packet you have 40 bytes of IP overhead, so the actual 8 Kb does not reflect the effective end bandwidth consumed on a WAN. Depending on the packet size, the payload header and the packets per second at which the router or IP phone transmits will actually change, or your end bandwidth consumption will vary. For example, a Cisco router running G.711 with a 160-byte payload at 50 packets per second will consume 80 kbps.

The G.729 mentioned before with 8 kbps of voice payload will actually consume 26.4 Kb of WAN bandwidth. Also note that not included are the link layer header sizes. These can vary per media—Frame Relay, Multilink PPP, and so on. Header sizes, even Ethernet, will boost your overall consumed bandwidth.



Echo is hearing your own voice in the telephone receiver while you are talking. When timed properly, echo is reassuring to the speaker; if the echo exceeds approximately 25 milliseconds, it can be distracting and cause breaks in the conversation. In a traditional telephony network, echo is normally caused by a mismatch in impedance from the 4-wire network switch conversion to the 2-wire local loop and controlled by echo cancellers. In voice packet-based networks, echo cancellers are built into the low-bit rate CODECs and are operated on each DSP. Echo cancellers are limited by design by the total amount of time they will wait for the reflected speech to be received, which is known as an echo trail. The echo trail is normally 32 milliseconds.



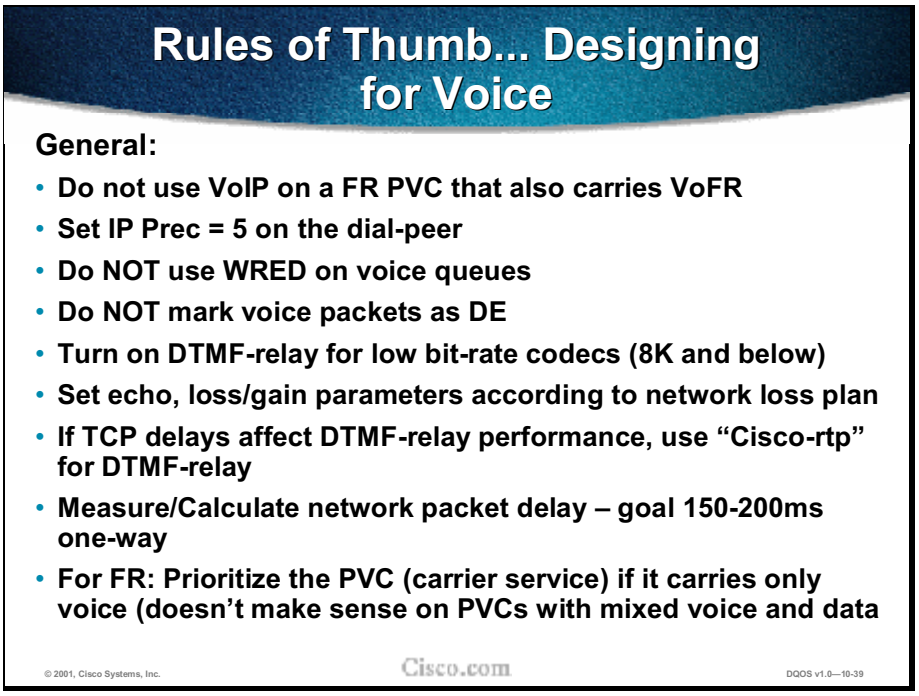
Sending VoIP control signals as best effort can result in high variations in delay and quality for user services, including Delay to Dial-Tone (DDT). During periods of heavy network congestion, DDT times can reach the multiple-second range.

Voice call setup should not be dependent on the number of FTP, HTTP, and SMTP streams currently active in the data network. Therefore, VoIP control sessions should be marked with a higher classification than best-effort data traffic. However, these control sessions *should not* be marked the same as the VoIP bearer channels (DSCP=EF). There are no compelling reasons to add VoIP control traffic to the priority queue (in fact, there are many reasons not to).

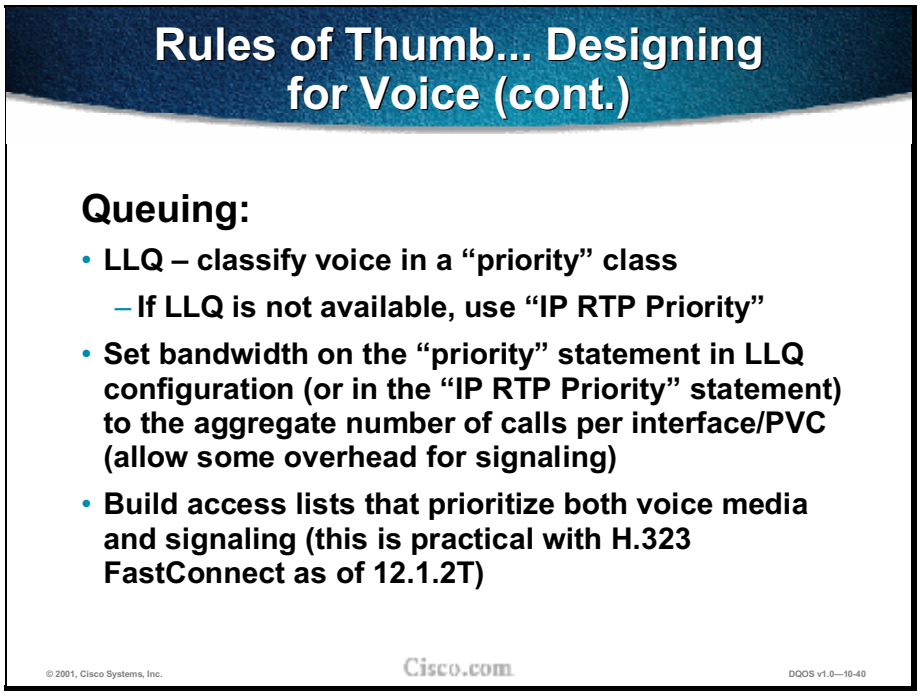
A solution is to mark Voice over IP control traffic as assured forwarding 31 (AF31), as opposed to Voice over IP bearer traffic marked as expedited forwarding (EF) and Video over IP bearer plane traffic as AF41. The DSCP value AF31 correlates to IP Precedence 3 for network elements that do not yet comply with the DSCP specifications.

Voice over IP Bearer Plane Traffic Classification

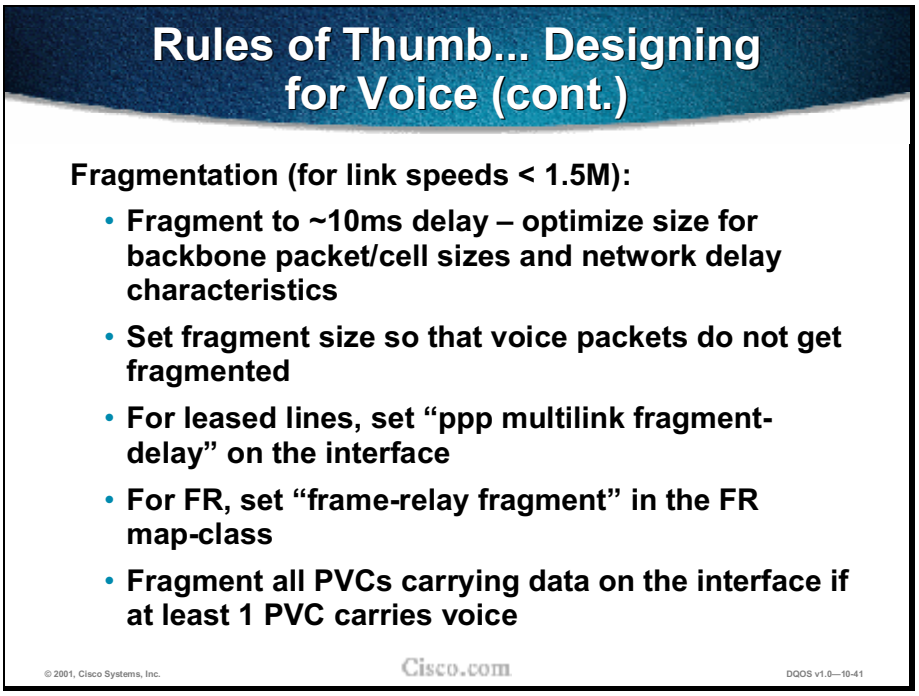
Voice has very strict requirements with regard to packet delay and packet loss sensitivity. The ITU recommends a one-way maximum delay of 150 ms for a voice conversation. While it has been proven that 200 ms of one-way delay is acceptable, this is still a very strict delay budget. Perhaps more important, variable network delay, defined as jitter, can induce voice buffer overruns on the receiving VoIP endpoint that results in degraded voice quality.



The next few slides highlight and summarize some design considerations when implementing a voice/data integrated network.



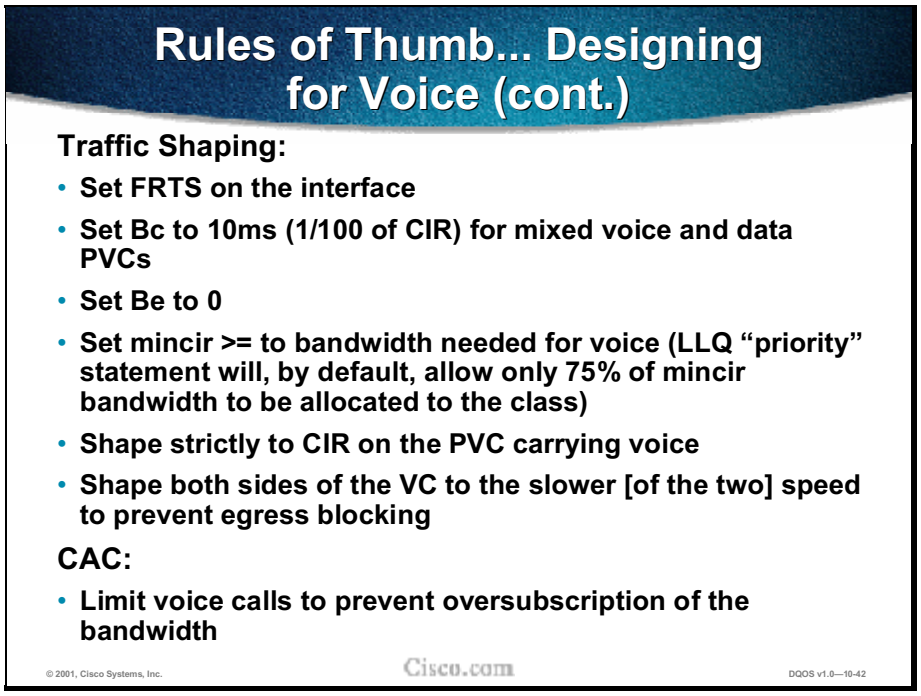
- Use 12.0.7T or later with LLQ on interfaces, or 12.1.2T or later with LLQ on FR PVCs.
- IP RTP Priority and LLQ cannot be applied to the same interface or VC at the same time.



As indicated before, fragmentation and interleaving mechanisms should be deployed on slow links to optimize end-to-end QoS. Illustrated are some of the design considerations when fragmenting in a voice/data integrated environment.

FRF.12 is the functional equivalent of LFI over MLP, where it will fragment large frames at Layer 2, interleave the voice packets, and actually reassemble that large frame on the other end of the circuit.

ATM, inherently using its 53-byte cell, also has the ability of interleaving. In many cases, ATM has an interface that is higher than the T1 and interleaving may not be required. However, you may be dealing with low-speed PVCs that may need some sort of link efficiency mechanisms. In the ATM segmentation and reassembly (SAR) process, if you actually take a 1500-byte frame and turn it into cells, they need to be in sequence throughout the life of that transmission; in some cases two PVCs may be needed.



FRTS allows line speed in bursts so that the average over time works out to committed information rate (CIR).

Bursting Guidelines:

Single PVC—Limit bursting to CIR

- The safest: Guaranteed what you pay for
- Fragment if link on either side of the network is slow speed

Single PVC—Mark data discard eligible

- Your data dropped first upon network congestion

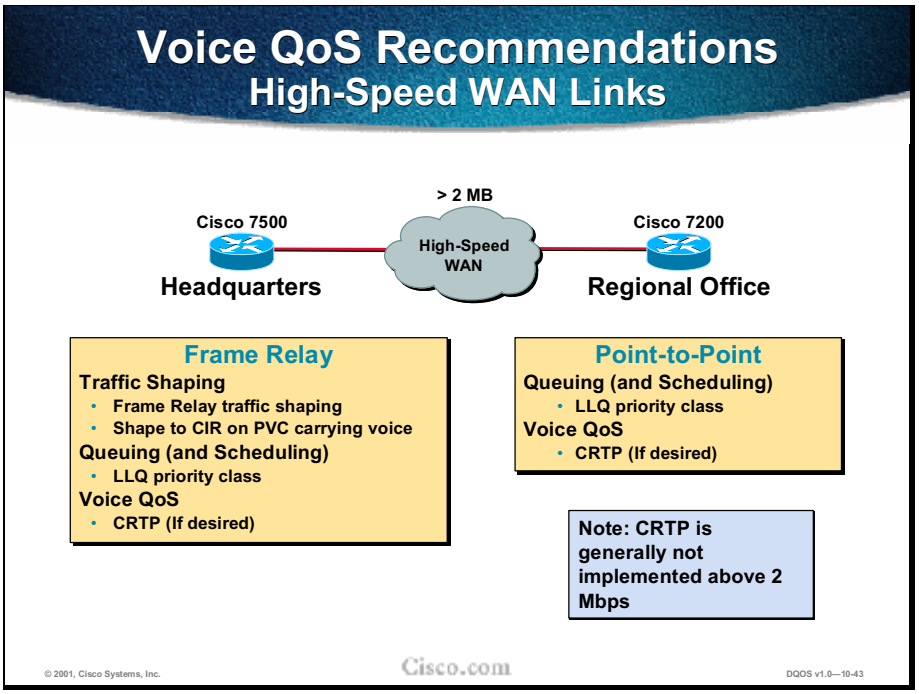
Single PVC—Utilize BECNs, ForeSight, or ABR

- Only invoked when congestion has already occurred
- Round-trip delays: Congestion indication must get back to source

Dual PVCs—One for voice and one for data

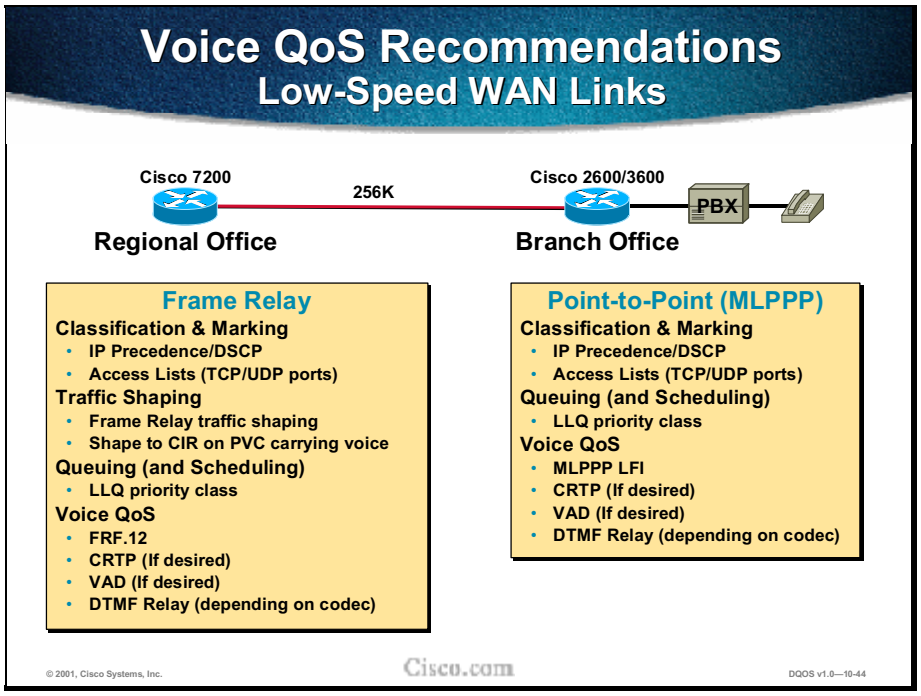
- One for data (may burst), one for voice (keep below CIR)
- Must perform PVC prioritization in Frame cloud (Cisco WAN gear does)
- Fragmentation rules still apply for data PVC
- IP routing complicated if VoIP over FR

CAC is used to ensure that voice is separated from voice.

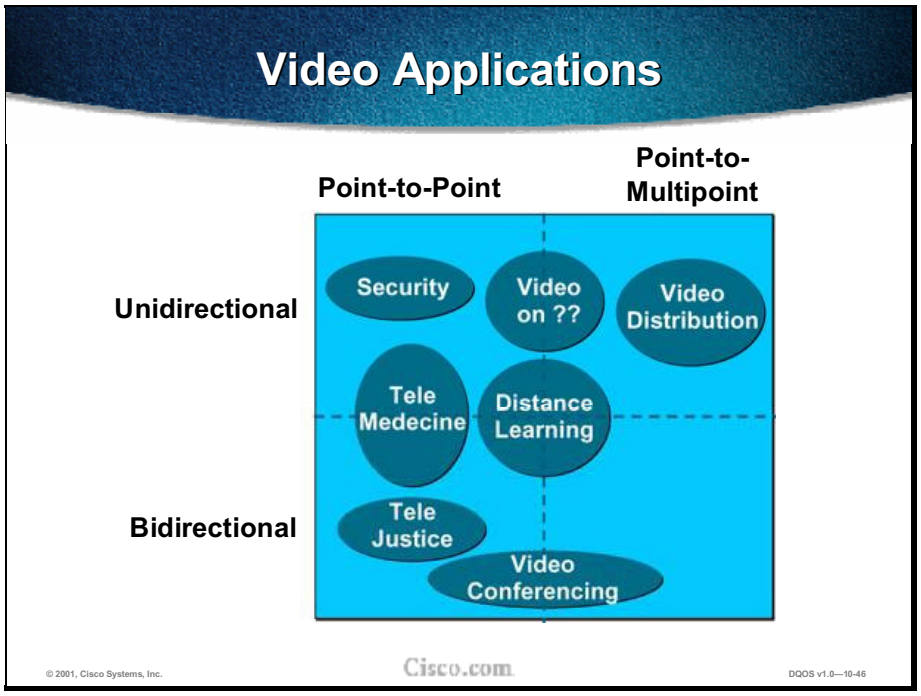


Both PPP and Frame Relay T1 links require use of priority queuing in order to grant voice the strict priority it requires. If voice bandwidth is an issue, CRTP to compress the large RTP header will help. Because the T1 line is high speed, it is not necessary to fragment large data packets to interleave with voice packets.

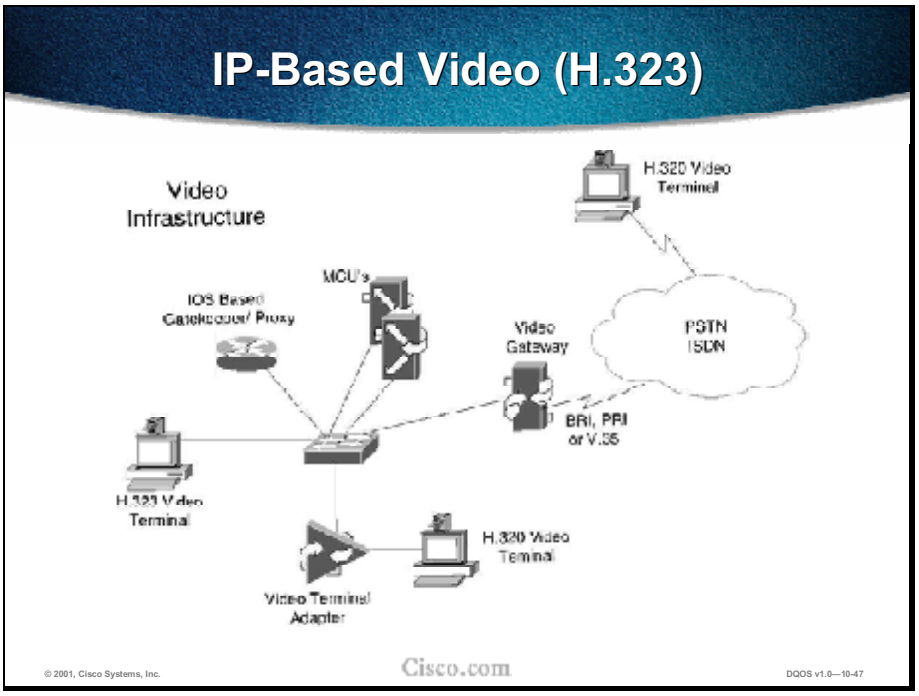
For any VoIP network, compression reduces the bandwidth necessary for each voice call, thus increasing the number of calls that can be sent over a given link.



In addition to the tools used for high-speed links, FRF.12 or LFI may be needed to reduce latency resulting from large packets, and the benefits of CRTP (bandwidth and latency reduction) become more apparent.



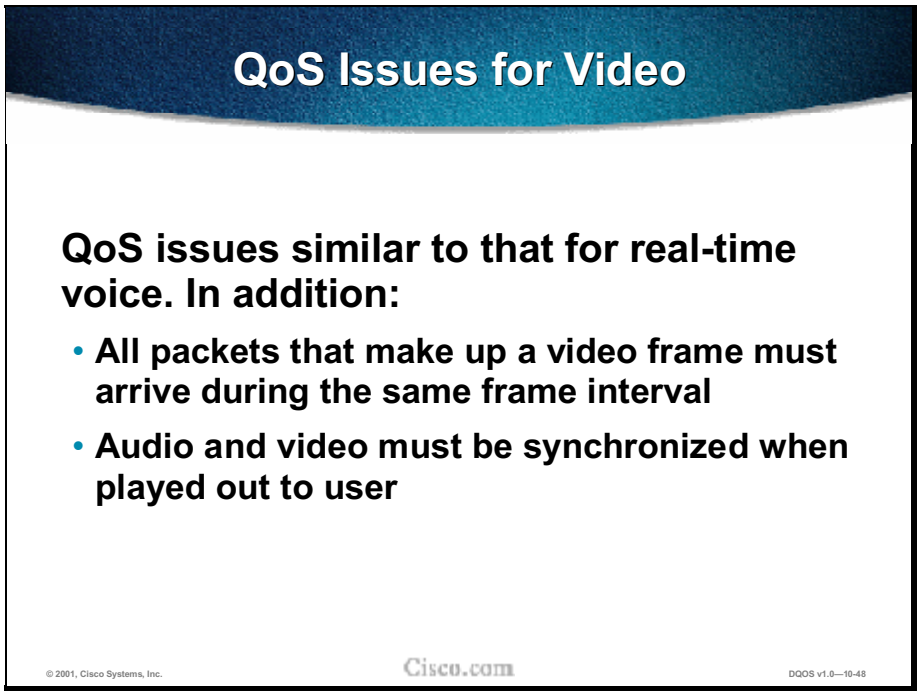
The point-to-multipoint technology allows the transmission of programs to large audiences without placing an unnecessary burden on the server or the network. However, the network infrastructure must support multicast routing and multicast group protocols. Unidirectional video traffic is noninteractive while bidirectional traffic involves interactive transactions between source and destination.



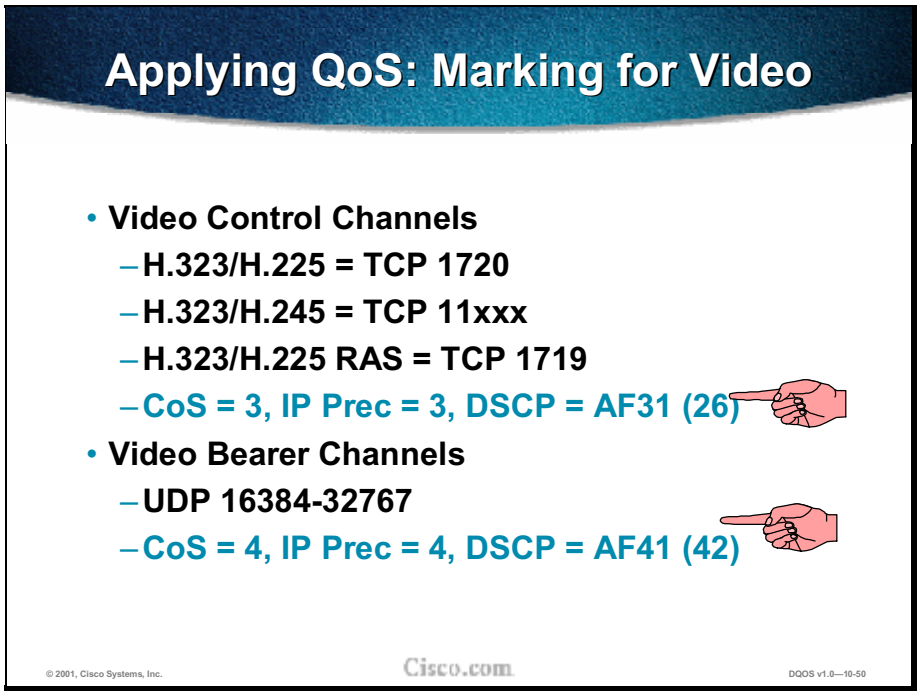
The above figure illustrates an H.323 network.

There are four components that make up an H.323 videoconferencing network. These four components are:

- Video terminals
- Gatekeepers
- Gateways
- Multipoint conference units (MCU)



Video has many of the same issues that voice has. In addition, video cannot tolerate bursty data transmission, loss of packets, delay in two-way situations, or variation in delay.

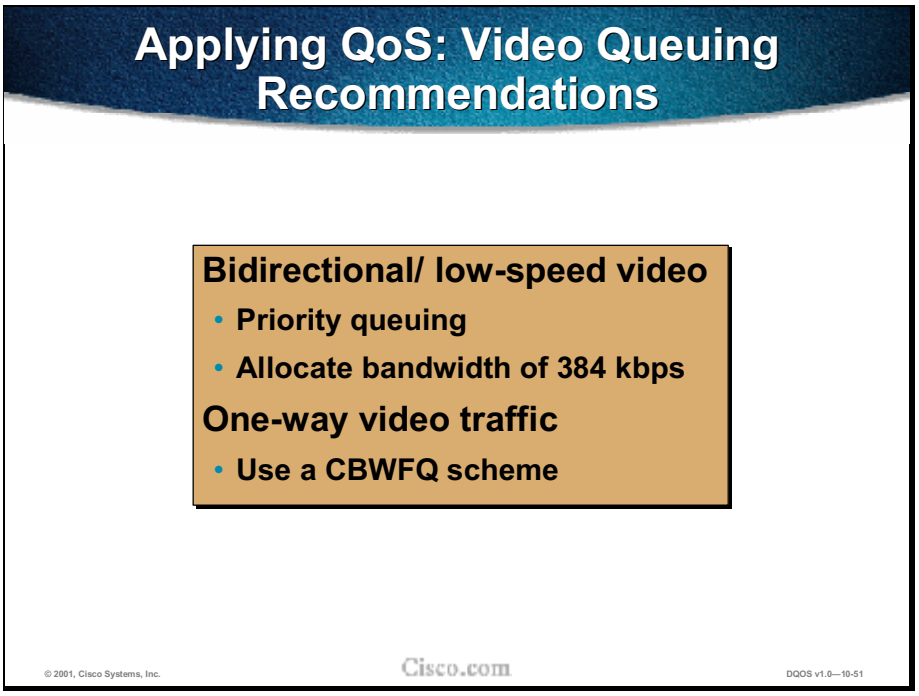


Video over IP Bearer Plane Traffic Classification

While Video over IP does have consistent and demonstrable maximum one-way delay values and minimum packet loss ratios, these values are much lower than for Voice over IP. For example, a typical one-way delay value for an MPEG-2 video stream is in the neighborhood of 400 to 450 ms before video quality is adversely affected. Because of the higher delay tolerances, video packets should not be placed into the voice-oriented EF priority queue on network elements. Another, more compelling reason for not mixing voice and video packets is the sheer size of the Video over IP packets. Almost all video endpoints will transmit video packets using the maximum link MTU size. Each defined priority queue in a network element is small to absolutely minimize the amount of delay incurred by an EF-tagged packet as it traverses the node. If large video packets are placed into small priority queues, the absolute end-to-end delay incurred by a small VoIP packet will increase, because it may enter the queue behind a larger packet.

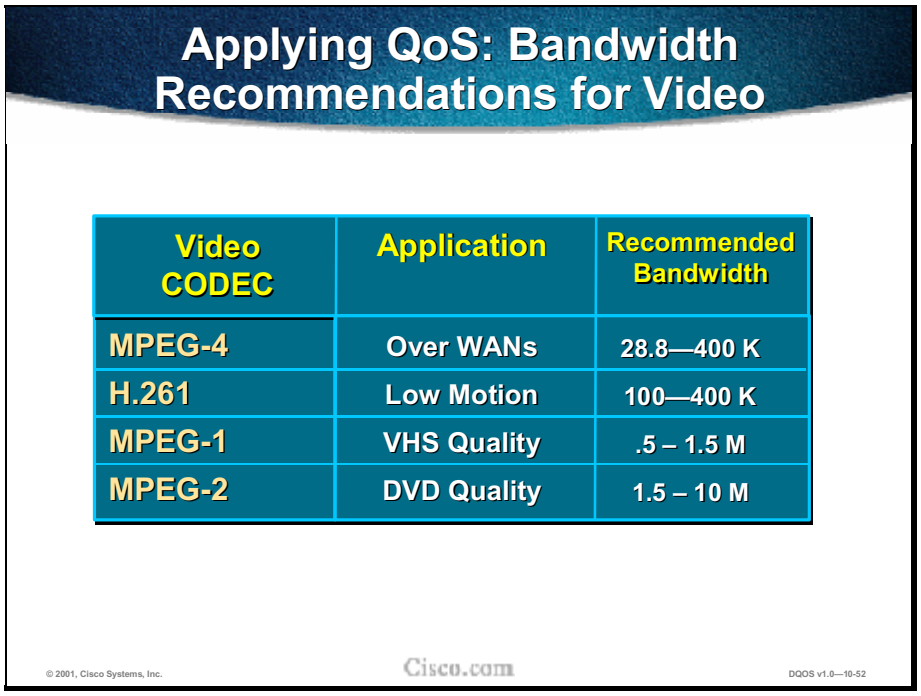
The solution to this dilemma, barring the creation of a second EF class, is to use AF41 DSCP value to classify video over IP packets. AF41 provides the highest classification within the assured forwarding classes, while also specifying the lowest packet-drop percentages. The DSCP value AF41 correlates to IP Precedence 4 for network elements that do not yet comply with the DSCP specifications.

Note	An interim step of marking both Video over IP control plane traffic and bearer plane traffic with a DSCP value of AF41 is acceptable because video control traffic overhead is very low in proportion to the actual bearer plane traffic.



On low-speed links with bidirectional video, queue video traffic with priority queuing capabilities, and allocate a bandwidth of 384 kbps. Traffic in excess of 384 kbps would be dropped if the interface becomes congested. Also use an admission control mechanism to ensure that this value is not exceeded.

One-way video traffic, such as IP/TV, uses a CBWFQ scheme because the delay tolerances are much higher. Also overprovision the class for low latency.



This figure relates the different video compression standards to everyday applications and the recommended bandwidth.

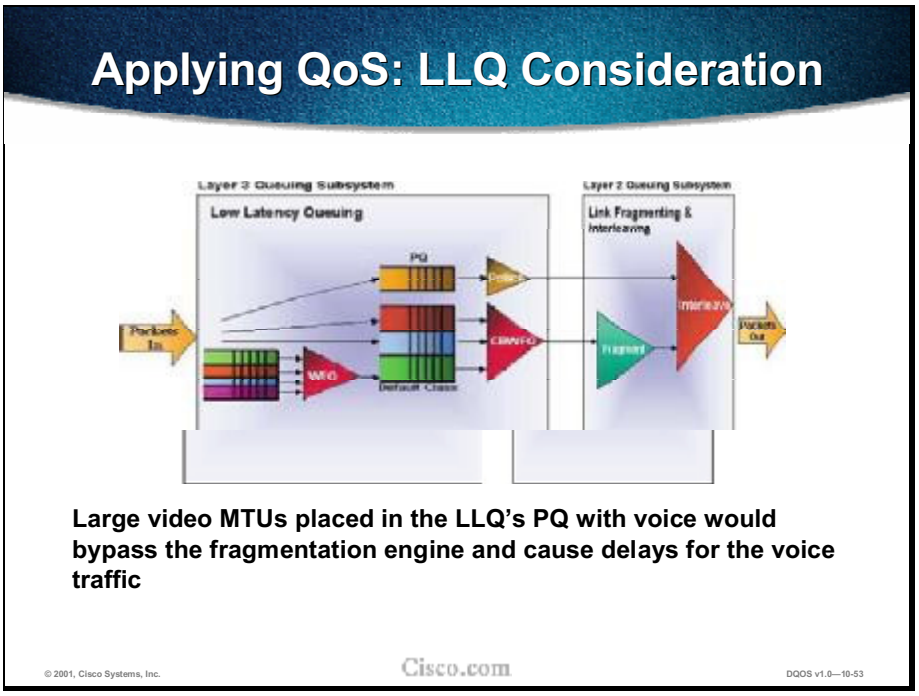
MPEG-1 is ISO standard 11172.

MPEG-2 is ISO standard 13818.

MPEG-4 is designed for low bandwidth and low-speed WAN environments.

H.261 was developed for videoconferencing and uses CIF resolution of 352 x 288 and QCIF resolution of 176 x 144.

H.261 is supported across multiple workstation platforms. With H.261 a program can be viewed from a workstation running Mac OS or UNIX OS in addition to Windows.

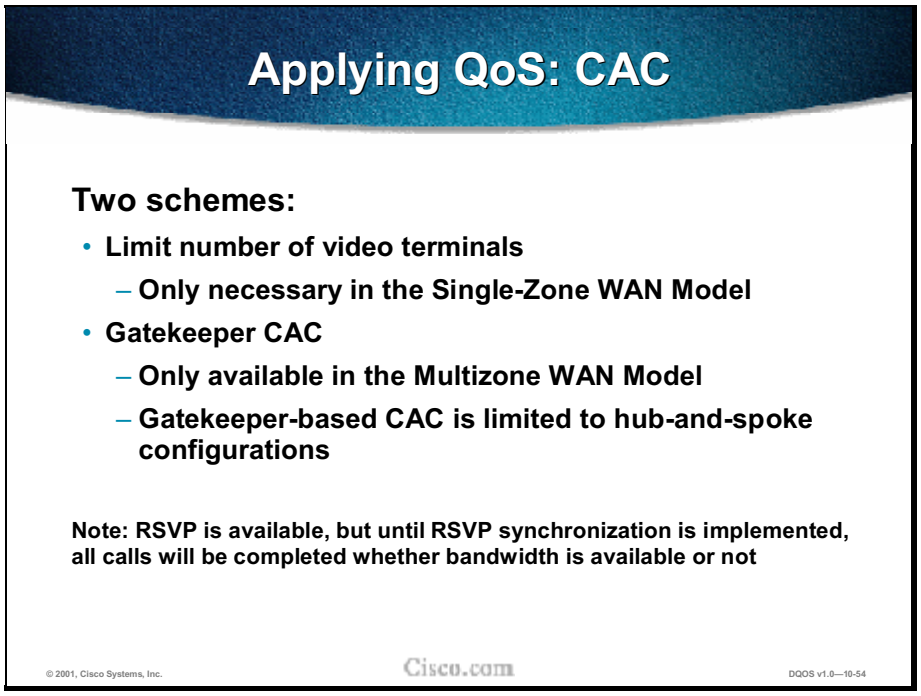


LLQ provides a strict priority queue (PQ) for video and voice traffic and weighted fair queues for other classes of traffic. LLQ, also called priority queuing/class-based weighted fair queuing (PQ/CBWFQ), is a superset of and more flexible than previous WAN quality of service offerings, in particular Real-Time Transport Protocol (RTP) prioritization and priority queuing/weighted fair queuing (PQ/WFQ).

With RTP prioritization and PQ/WFQ, traffic that matches a specified User Datagram Protocol (UDP)/RTP port range is considered high priority and allocated to the PQ. With low latency queuing for Frame Relay, you set up classes of traffic according to protocol, interface, or access lists and then define policy maps to establish how the classes are handled in the priority queue and weighted fair queues.

Queues are set up on a per-permanent virtual circuit (PVC) basis: Each PVC has a PQ and an assigned number of fair queues. The fair queues are assigned weights proportional to the bandwidth requirements of each class; a class requiring twice the bandwidth of another will have half the weight. Oversubscription of the bandwidth is not permitted. The command line interface (CLI) will reject a change of configuration that would cause the total bandwidth to be exceeded. This functionality differs from that of weighted fair queuing (WFQ), in which flows are assigned a weight based on IP Precedence. WFQ allows higher-precedence traffic to obtain proportionately more of the bandwidth, but the more flows there are, the less bandwidth is available to each flow.

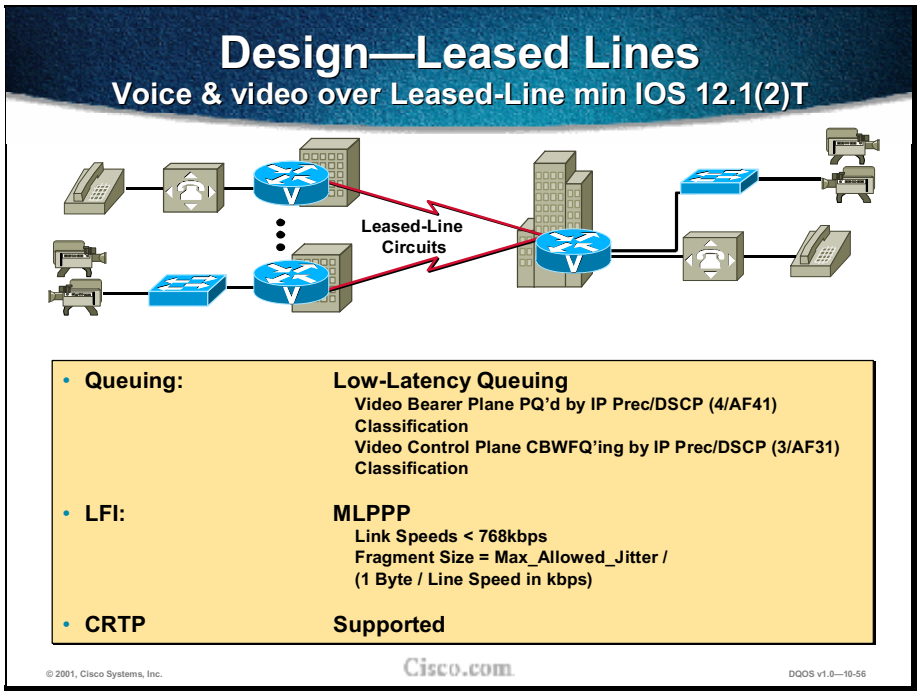
The PQ is policed to ensure that the fair queues are not starved of bandwidth. When you configure the PQ, you specify in kbps the maximum amount of bandwidth available to that queue. Packets that exceed that maximum are dropped. There is no policing of the fair queues.



Call admission control (CAC) is required to ensure that the network resources are not oversubscribed. Calls that exceed the specified bandwidth are rejected to ensure video quality. There are only two schemes for providing CAC for video calls over the WAN:

- Limited number of video terminals: Limiting the number of video terminals for CAC is only necessary in the single-zone WAN model. With the lack of a gatekeeper at remote sites, the only way to control the amount of bandwidth used for video across the WAN is to physically limit the number of video terminals at remote sites. The priority queue at each site must then be provisioned for the maximum possible data rate of all the video endpoints at any given site.

- Gatekeeper CAC: This is only available in the multizone WAN world. The gatekeeper allows administrators to set bandwidth limits for interzone, intrazone, or on a per-session basis. This allows administrators the ability to set an interzone bandwidth limit, provision a priority queue for the same bandwidth, and ensure the integrity of that queue. Note that today, gatekeeper-based CAC is limited to hub-and-spoke configurations.



The network-side hardware for the remote office connecting into a leased-line circuit is typically a standard Cisco low-speed serial interface. The following quality-of-service and link-optimization services available through IOS are recommended:

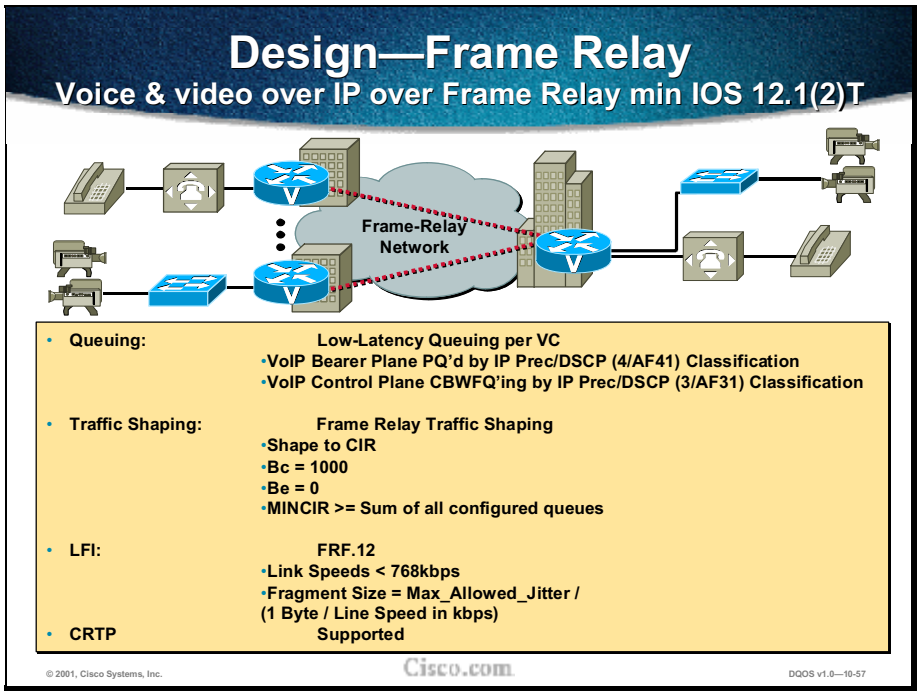
- DSCP
- IP RTP Priority: IP RTP Priority provides a strict-priority queuing method for RTP traffic.
- WFQ/CBWFQ: Weighted fair-queuing and/or class-based WFQ are recommended for all other traffic traversing the leased-line circuit. CBWFQ allows traffic to be classified into broad classes, which can then be guaranteed a certain amount of bandwidth across the circuit.
- MLPPP should be utilized for PPP circuits less than 768 kbps. For circuits greater than 768 kbps, HDLC encapsulation can be utilized since it is more efficient than PPP. LFI is not necessary for higher-speed circuits.
- CRTP

The choice of a compression algorithm to use should be based upon the voice quality desired and the amount of bandwidth available for voice transport.

- VAD: Voice-activity detection optimizes link utilization by not sending packets when there is no voice traffic.

- Fax relay services can be used to provide optimization for fax traffic.

At the head end, a variety of serial interfaces can be used. Individual circuits can be terminated on standard Cisco low-speed serial interfaces. The circuits can also be aggregated by the carrier and brought in on a multichannel T1/E1 or T3/E3 interface to lower overall costs and ease maintenance.



The network-side hardware for the remote office connecting into a Frame Relay cloud is typically a standard Cisco low-speed serial interface. The following quality-of-service and link-optimization services available through IOS are recommended:

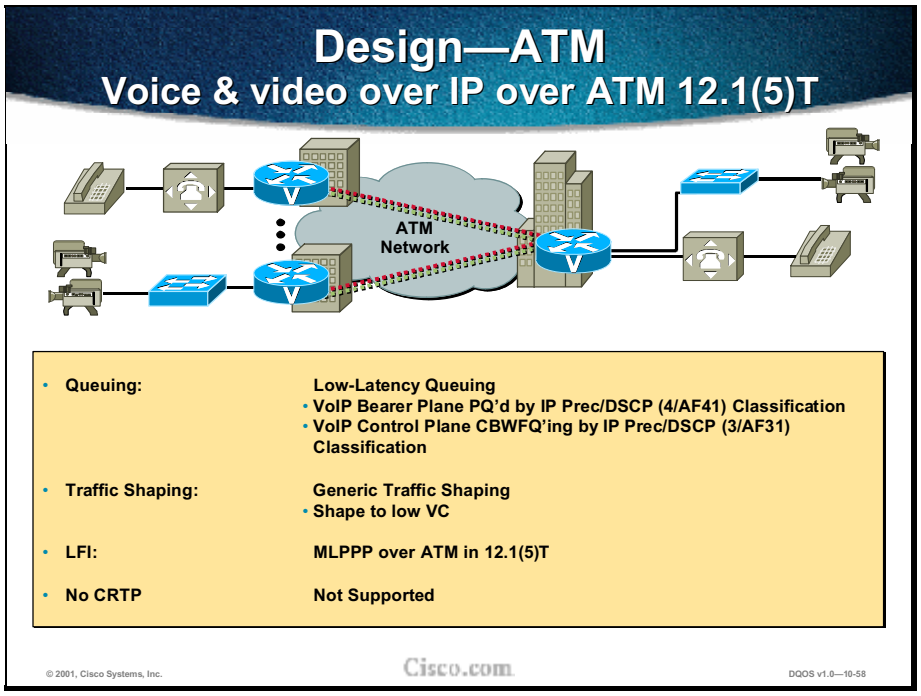
- DSCP
- IP RTP Priority: IP RTP Priority provides a strict-priority queuing method for RTP traffic.
- WFQ/CBWFQ: Weighted fair-queuing and/or class-based WFQ are recommended for all other traffic traversing the Frame Relay PVC. CBWFQ allows traffic to be classified into broad classes, which can then be guaranteed a certain amount of bandwidth across the circuit.
- Standards-based FRF.12 fragmentation is the recommended link fragmentation and interleaving method to implement on low-speed Frame Relay circuits.
- In order to guarantee voice quality, Frame Relay traffic shaping should be set to CIR. Likewise, there should be no oversubscription at the headend.
- CRTP

Cisco supports a wide range of voice compression algorithms including G.729, G.723.1, G.732, and G.71, among others. The choice of a compression algorithm to use should be based upon the voice quality desired and the amount of bandwidth available for voice transport.

- VAD: Voice-activity detection optimizes link utilization by not sending packets when there is no voice traffic.

- Fax relay services can be used to provide optimization for fax traffic.

At the headend, a variety of serial interfaces can be used. Individual circuits and/or PVCs can be terminated on standard Cisco low-speed serial interfaces. The circuits and/or PVCs can be aggregated by the carrier and brought in on a multichannel T1/E1 or T3/E3 interface or HSSI interface to lower overall costs and ease maintenance.



The network side hardware for the remote office connecting into an ATM cloud is typically a low-speed DS-1 ATM interface. The following quality-of-service and link-optimization services available through IOS are recommended:

- DSCP
- CBWFQ within a VC: Class-based WFQ is recommended for traffic traversing the ATM virtual circuit. CBWFQ allows traffic to be classified into broad classes, which can then be guaranteed a certain amount of bandwidth across the virtual circuit.
- For PVC speeds less than 768 kbps, separate PVCs for voice and data traffic are recommended. Although the ATM SAR function fragments all traffic to 53-byte cells, there is no interleaving function, which guarantees prioritization of voice cells over data cells within a single PVC.

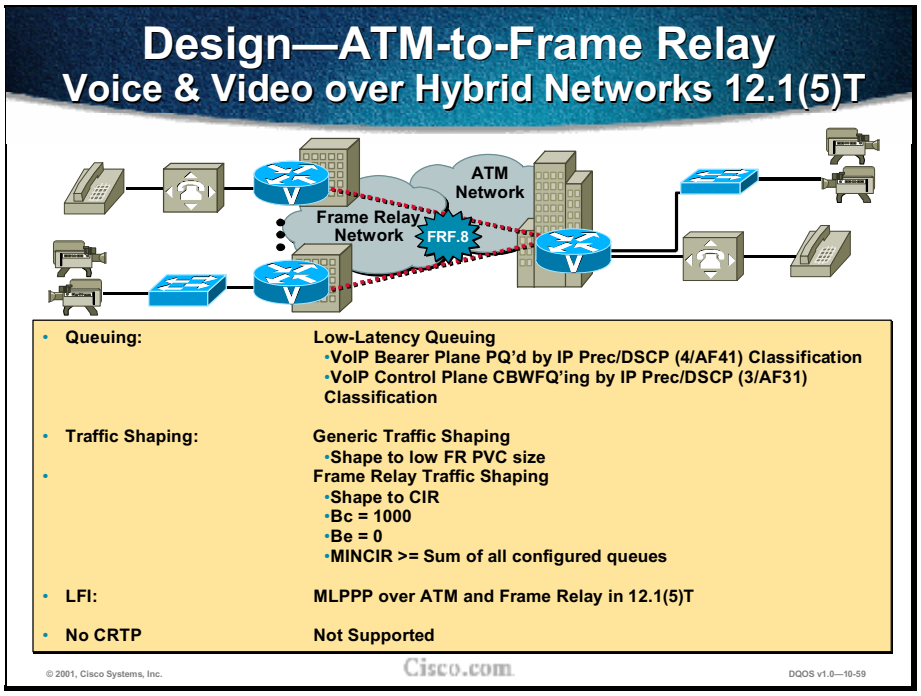
- In order to guarantee voice quality, traffic shaping should be set to the guaranteed rate of the ATM service. Likewise, there should be no oversubscription at the headend.

- Cisco supports a wide range of voice compression algorithms including G.729, G.723.1, G.732, and G.71, among others. The choice of a compression algorithm to use should be based upon the voice quality desired and the amount of bandwidth available for voice transport.

- VAD: Voice-activity detection optimizes link utilization by not sending packets when there is no voice traffic.

- Fax relay services can be also used to provide optimization for fax traffic.

At the headend, a variety of interfaces can be used. Individual circuits and/or PVCs can be terminated on DS-1 ATM interfaces. The circuits and/or PVCs can be aggregated by the carrier and brought in on a DS-3 or OC-3 ATM interface to lower overall costs and ease maintenance.



In this solution the remote side consists of Frame Relay service, while the headend side consists of ATM service. The service provider provides the FRF.8 service interworking between Frame Relay and ATM. The network-side hardware for the remote office connecting into a Frame Relay cloud would typically be a standard Cisco low-speed serial interface. The following quality-of-service and link-optimization services available through IOS are recommended:

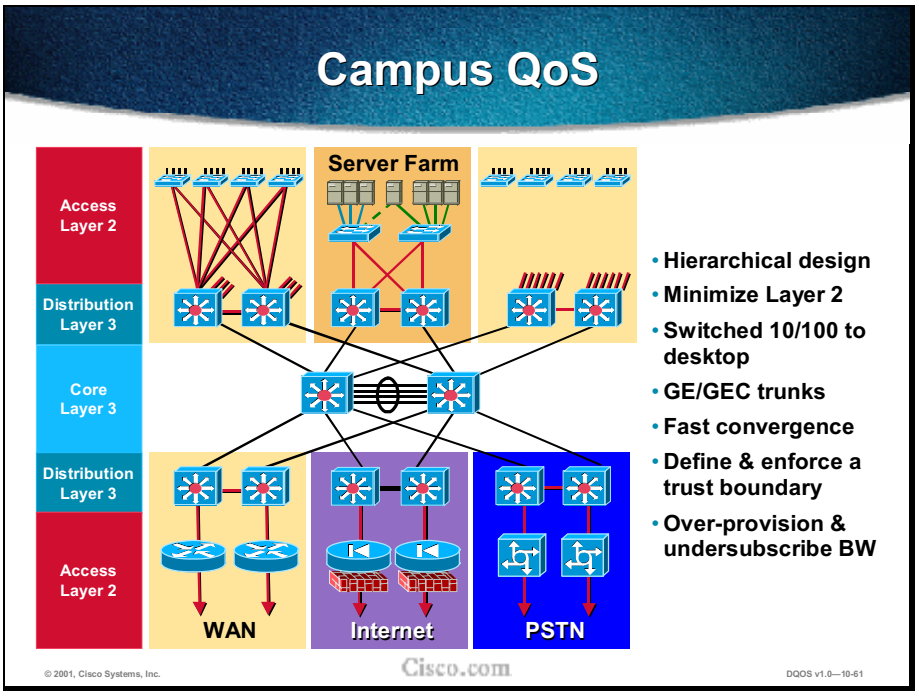
- DSCP
- CBWFQ within a VC—Class-based WFQ is recommended for traffic traversing the virtual circuit. CBWFQ allows traffic to be classified into broad classes, which can then be guaranteed a certain amount of bandwidth across the virtual circuit.
- For PVC speeds less than 768 kbps, separate PVCs for voice and data traffic are recommended. Although the ATM SAR function fragments all traffic to 53-byte cells, there is no interleaving function which guarantees prioritization of voice cells over data cells within a single PVC.
- In order to guarantee voice quality, traffic shaping should be set to the guaranteed rate of the Frame Relay and/or ATM service. Likewise, there should be no oversubscription at the headend.

- Cisco supports a wide range of voice compression algorithms including G.729, G.723.1, G.732, and G.71, among others. The choice of a compression algorithm to use should be based upon the voice quality desired and the amount of bandwidth available for voice transport.

- VAD—Voice-activity detection optimizes link utilization by not sending packets when there is no voice traffic.

- Fax relay services can be also used to provide optimization for fax traffic.

At the headend, a variety of interfaces can be used. Individual circuits and/or PVCs can be terminated on DS-1 ATM interfaces. The circuits and/or PVCs can be aggregated by the carrier and brought in on a DS-3 or OC-3 ATM interface to lower overall costs and ease maintenance.

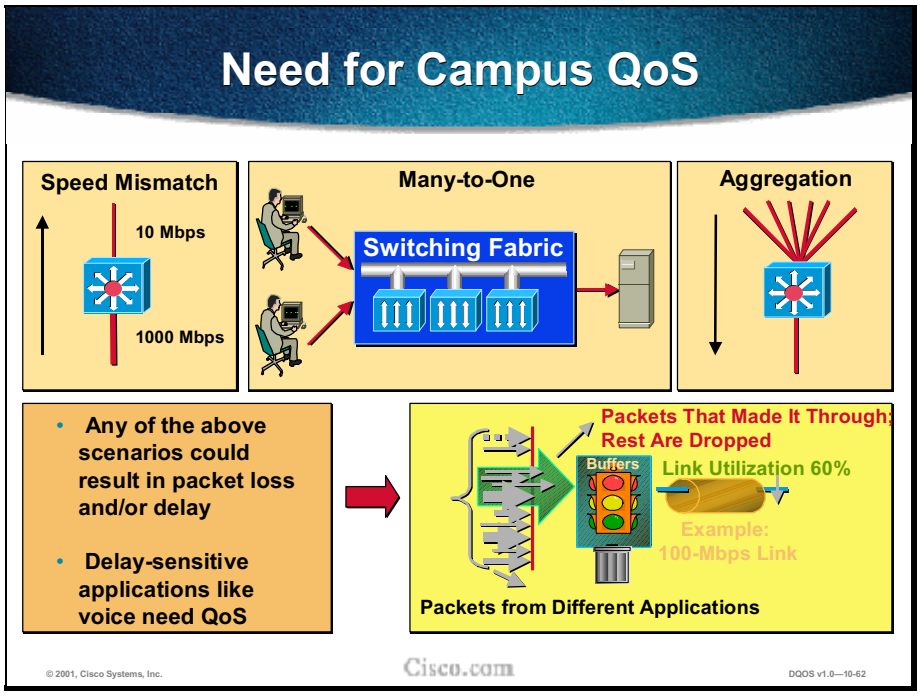


This graphic illustrates a typical network of today.

Many campus networks today are overprovisioned and underutilized; therefore, congestion management tools are of little value in these networks. However, campus networks are subject to oversubscription (congestion), just as low-speed WAN links are. Eventually, some form of queuing will be needed in the campus as demand for bandwidth increases.

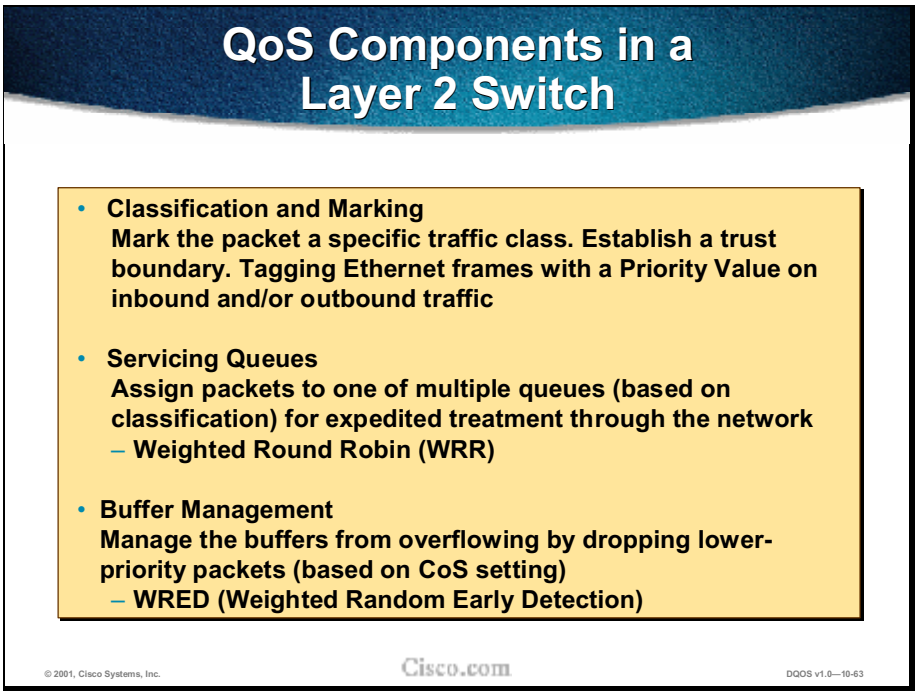
The 10/100 Ethernet to the desktop has to be switched. As new business client/server applications such as enterprise resource planning and supply-chain management are deployed on top of existing applications, network loads get heavier and heavier, forcing users to wait for files to download or screens to update. Users might mistakenly assume that a newly installed application is slow, when in reality a legacy network based on shared hubs is simply not up to the task of handling the extra load placed on it.

As shared devices, hubs share a fixed amount of bandwidth among connected users. Switches provide dedicated 10-Mbps or 100-Mbps bandwidth per port to individual users or servers. Less contention among users for bandwidth means fewer collisions, resulting in enhanced application performance without costly wiring or network interface card changes.



Why is quality of service (QoS) needed in the campus? Until recently, conventional wisdom stated that QoS would never be an issue in the enterprise campus because of the bursty nature of data traffic and the capability to withstand buffer overflow and packet loss. When applications such as voice and video, which are sensitive to loss and delay, began to traverse the data network, network designers gradually came to understand that buffers, and not bandwidth, are the issue in the campus. Buffers can fill instantaneously. When this occurs, packets can be dropped when attempting to enter the interface buffer. For applications like voice, which are extremely drop intolerant, this results in voice-quality degradation.

Another reason is speed mismatches. Where Gigabit Ethernet traffic flows to Fast Ethernet, oversubscription of the common link is likely. For instance, a Catalyst 6000 with 384-kbps Fast Ethernet ports on it could well be oversubscribed. QoS tools are required to manage these issues to minimize loss, delay, and delay variation for voice and video.



There are three main mechanisms used for QoS in Catalyst switches.

- Has the ability to honor or rewrite a CoS/ToS/DSCP value in a frame on its way into or out of the switch.

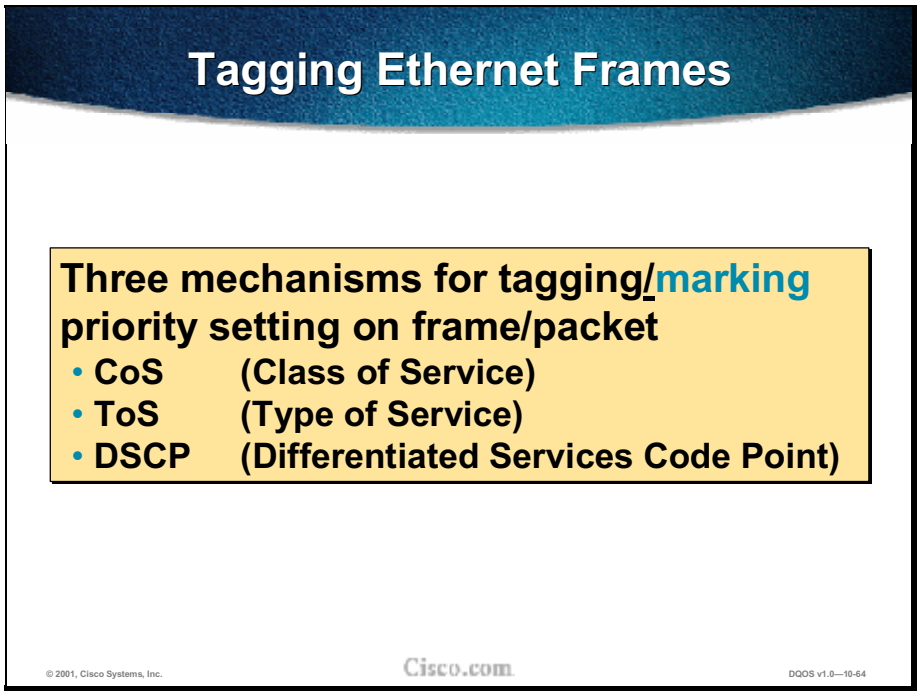
- Services the queues by forwarding eitherHigher-priority packets before lower-priority packets

  This technique is used on the Catalyst 3500XL (Tx queues) and Catalyst 6XXX Version 1A line cards (Rx and Tx queues). This operates in a similar fashion to priority queuing on Cisco IOS routers. In this mode, higher-priority queues must be empty before any frames in a lower-priority queue are serviced.

  – Higher-priority packets at a more frequent rate than lower priority packets

  This technique is referred to as weighted round robin (WRR), which allows the user to configure how much of the bandwidth a different queue has access to. For example, with two queues, you could configure one queue to get two-thirds of the bandwidth. WRR is available on the Catalyst 6XXX version 1A line cards. These line cards have three Tx queues, one strict-priority queue and a high and normal queue. The high and normal queue use WRR to service traffic between them. WRR is also available on the new Catalyst 4000 Layer 3 line card.

- Buffers management by dropping lower-priority packets (based on CoS settings). The mechanism used to perform this function is weighted random early detection (WRED). WRED monitors thresholds in the queue and starts dropping certain packets once a threshold has been exceeded. Different products have different configurable thresholds.



Class of Service

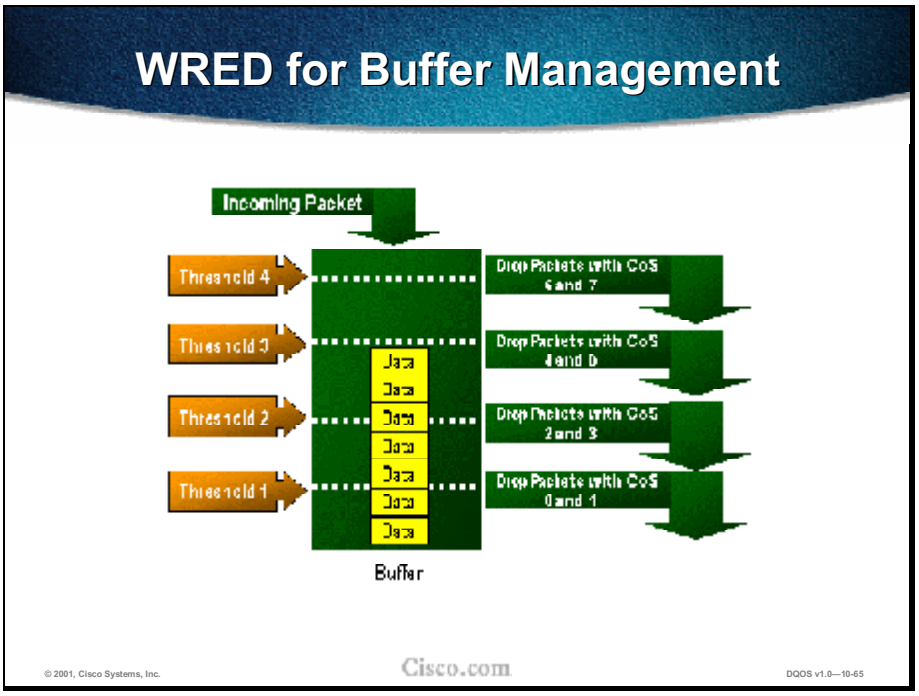
Class of service (CoS) refers to a tag that is inserted into the Ethernet frame. These tags form part of the IEEE 802.1Q tag or the ISL header. In the four-byte field (32 bits) are three bits that form part of the IEEE 802.1p specification, allowing for an Ethernet frame to be tagged with a priority. The three bits translate into eight priority settings (that is, 2 to the power of 3 = 8). Thus the values that are assignable to an Ethernet frame are 0 to 7 (0 being lowest, 7 being highest). In Windows 2000 machines running an Ethernet adapter supporting IEEE 802.1Q, users are able to set their own CoS values for Ethernet frames leaving that PC.

Type of Service

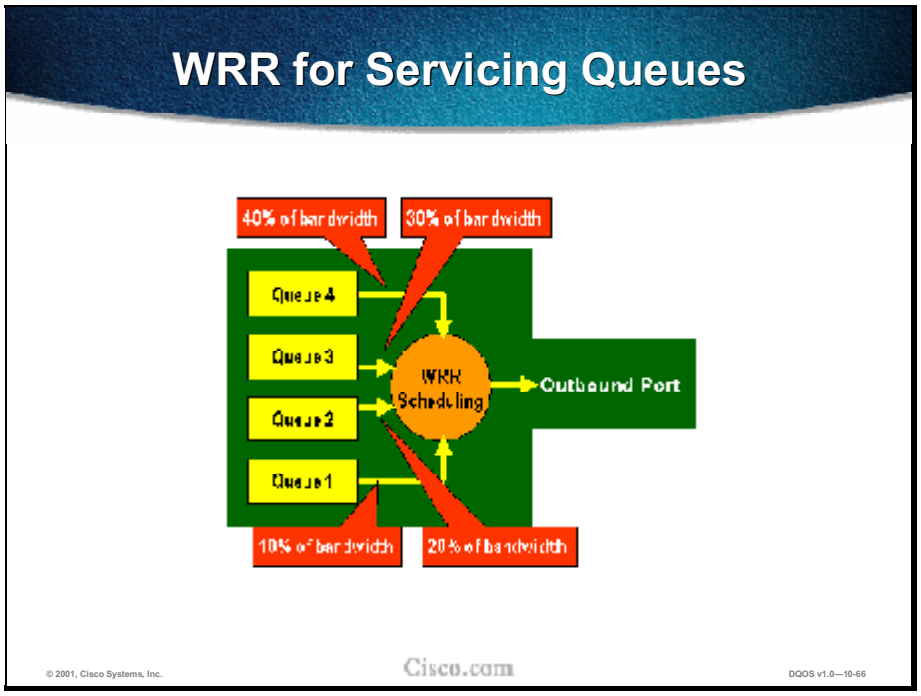
The type of service (ToS) field in the IPV4 header is a one-byte field. The first three bits of this field represent the IP Precedence, which allows for an IP packet to be tagged with a priority of 0 to 7. The main issue with using ToS is that if the IP Packet traverses a Layer 2 switch network, the IP Precedence value is lost inside the data field of the Ethernet frame. The Layer 2 switch cannot read the ToS field, hence no preference is assigned to higher-priority traffic. To maintain QoS across a Layer 2 network, there must be some way to map IP Precedence to CoS. This is a facility available in the Catalyst 5000, 6000, and 6500.

Differentiated Services Code Point

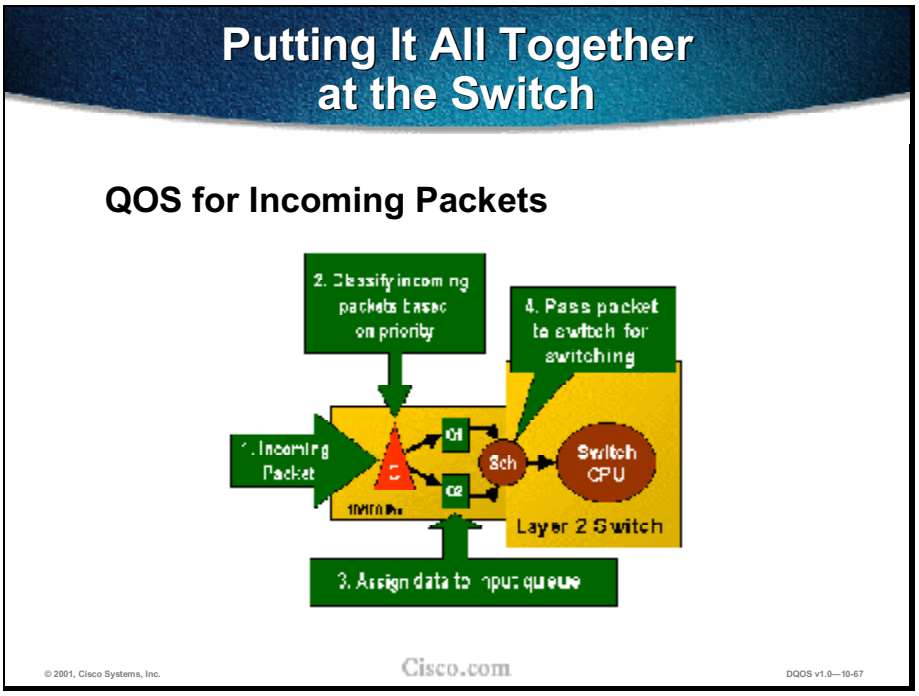
Differentiated services code point (DSCP) is a new QoS initiative from the standards body to provide more granular priority definitions. Like IP Precedence, DCSP can get lost when the DSCP marked IP packet traverses a Layer 2 switch network (as it is imbedded in the data portion of the Ethernet packet); thus, like the example above, no preferential treatment is given to a high-priority (DSCP-tagged) packet. With the Catalyst 6000, it gives the user the ability to map a DSCP value to a CoS value. As there are 64 DSCP values and only eight CoS values, typically a number of DSCP values are assigned to a singular CoS value. For instance, DSCP values 0, 1, 2, 3, 4 , 5, 6, and 7 are mapped to CoS value 0 and so on.



At times of traffic congestion, buffers start to fill up. When this happens, the switch can either let the buffers overflow or drop packets to keep the buffers from overflowing. WRED is a way for the switch to intelligently discard lower-priority traffic, keeping higher-priority traffic in the queue. The way it does this is to use the CoS value and thresholds. The weighted part of WRED looks at the CoS value of the Ethernet frame, and once a certain threshold has been reached, frames with certain CoS values are dropped. Thresholds are typically configurable by the user. Different switches implement a different number of thresholds, so check the manual as to what your catalyst switch offers. Catalyst 5000 and 6000 line cards that support WRED also allow the user to configure which CoS values get mapped to which thresholds. In the diagram above, Threshold 4 has CoS values 6 and 7 mapped to it. A user could quite easily configure CoS 5 to be mapped to threshold 4 if required.



WRR is a technique used by some Catalyst switches to apportion larger amounts of bandwidth to higher-priority traffic as traffic is scheduled outbound. Essentially when a port has more than a single queue, WRR can offer benefits to outbound priority-tagged traffic.



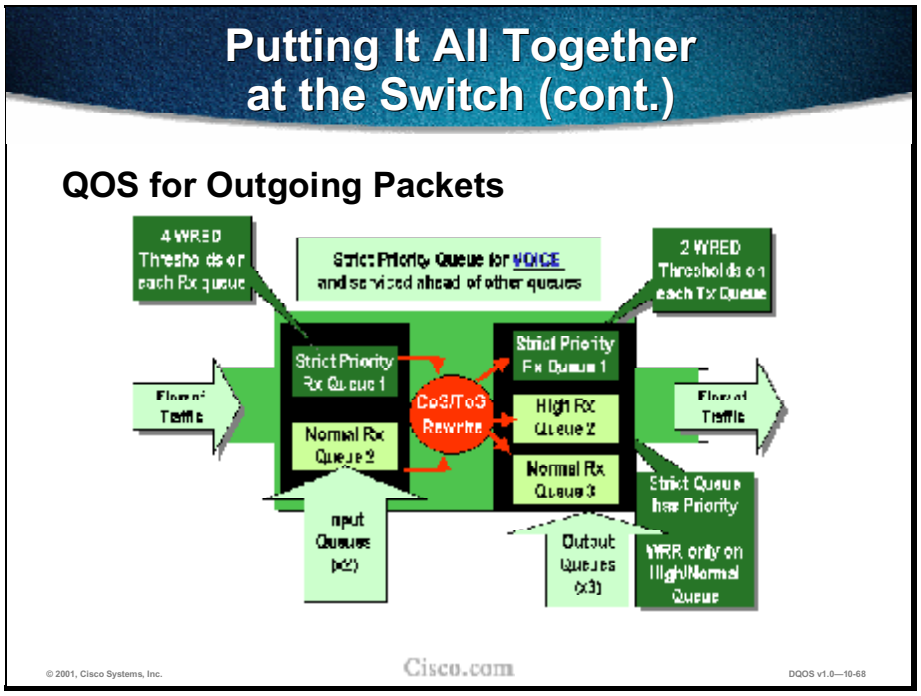
Depending on the level of QoS support in a switch, incoming packets can have the following QoS capabilities performed on them. The diagram above logically represents some of the features you may find in a Layer 2/3 switch.

Some switches allow for an incoming packet to be classified based on its priority. Most Layer 2 switches use the CoS setting for this classification process; however, some with an understanding of Layer 3 can inspect the ToS or DSCP setting. After inspecting this setting, the switch can perform the following functions:

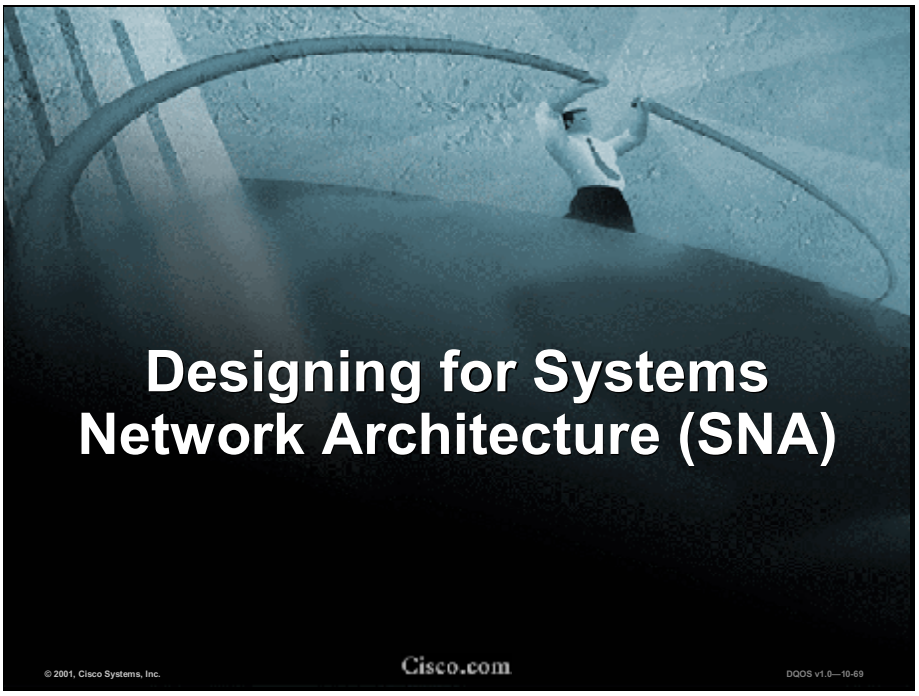
1. Leave the CoS setting or change it to another value

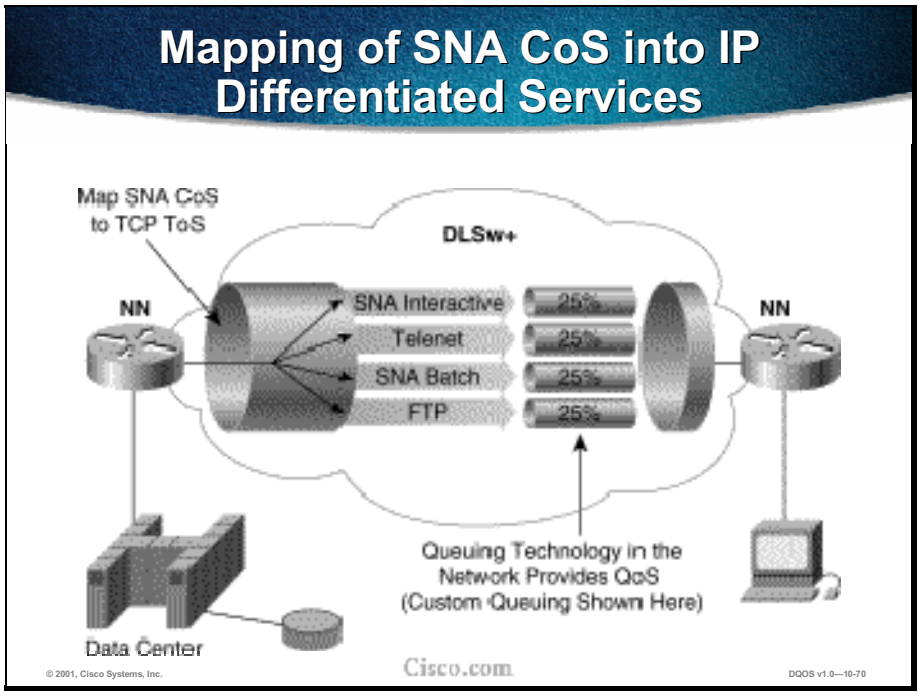
2. Use this CoS value and map it to a queue or a threshold in a queue

A scheduling component then services the queues by passing the higher-priority packets to the switch CPU for service. This example is best seen by the new Version 1A line cards for the Catalyst 6000. Each 10/100 port has two receive (Rx) queues, one deemed a strict-priority (SP) queue and the other a normal queue. The SP queue is used for high-priority traffic (such as VoIP) and must be empty before traffic from the other queue is serviced. The non “A” line cards for the Catalyst 6000 have a single Rx queue. as do line cards for the Catalyst 5000. The Catalyst 4000 has no Rx queues since each line card has nonblocking input into the switch fabric.



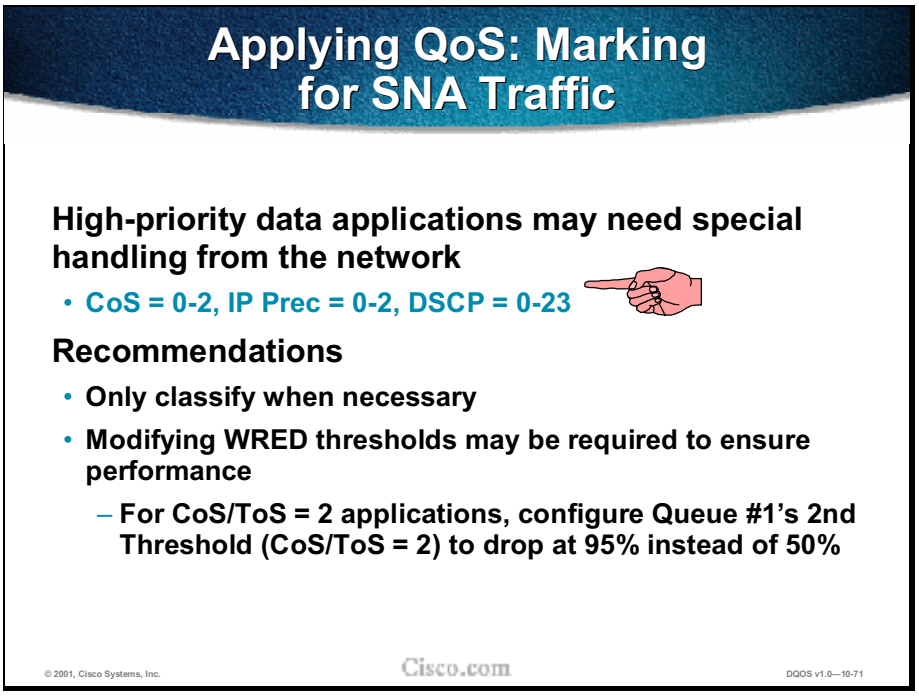
QoS for outgoing traffic on a switch can take advantage of all three QoS mechanisms (that is, CoS/ToS/DSCP, WRED, and WRR).The best example of what we can do with outbound QoS is with the new Version 1A line cards on the Catalyst 6000. There are three Tx (transmit) queues, one deemed a strict-priority queue (SP) and two other queues (high and normal). The SP queue is serviced like a priority queue; it is emptied before the other two queues are serviced. When it is empty, WRR is used to service the high and normal queue. Each Tx queue has two thresholds that are used by WRED to manage buffers from overflowing. Outbound traffic can also have their CoS/ToS/DSCP values adjusted based on ACLs defined in the switch. Map statements can also be made to map a CoS value to ToS/DSCP or vice versa.





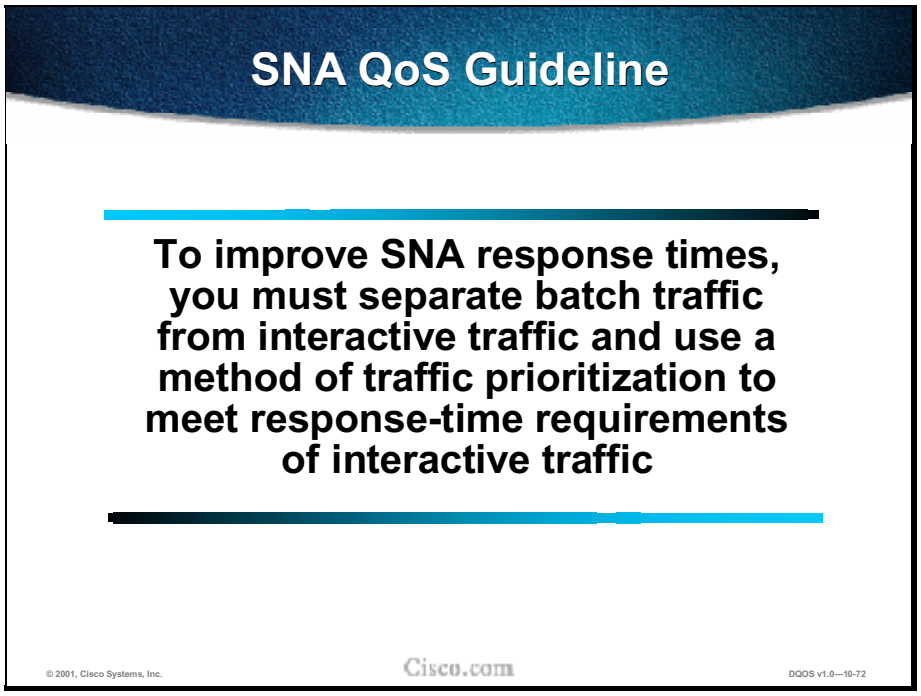
SNA ToS in conjunction with data-link switching plus (DLSw+) allows mapping of traditional SNA class of service (CoS) into IP differentiated service. This feature takes advantage of both QoS signaling and pieces of the architecture. DLSw+ opens four TCP sessions and maps each SNA ToS traffic into a different session. Each session is marked by IP Precedence. Cisco's congestion control technologies (CQ, PQ, and WFQ) act on these sessions to provide a bandwidth guarantee or other improved handling across an intranet. This provides a migration path for traditional SNA customers onto an IP-based intranet, while preserving the performance characteristics expected of SNA.

Thus, traditional mainframe-based, mission-critical applications can take advantage of evolving IP intranets and extranets without sacrificing the QoS capabilities historically provided by SNA networking.



Data Traffic Classification

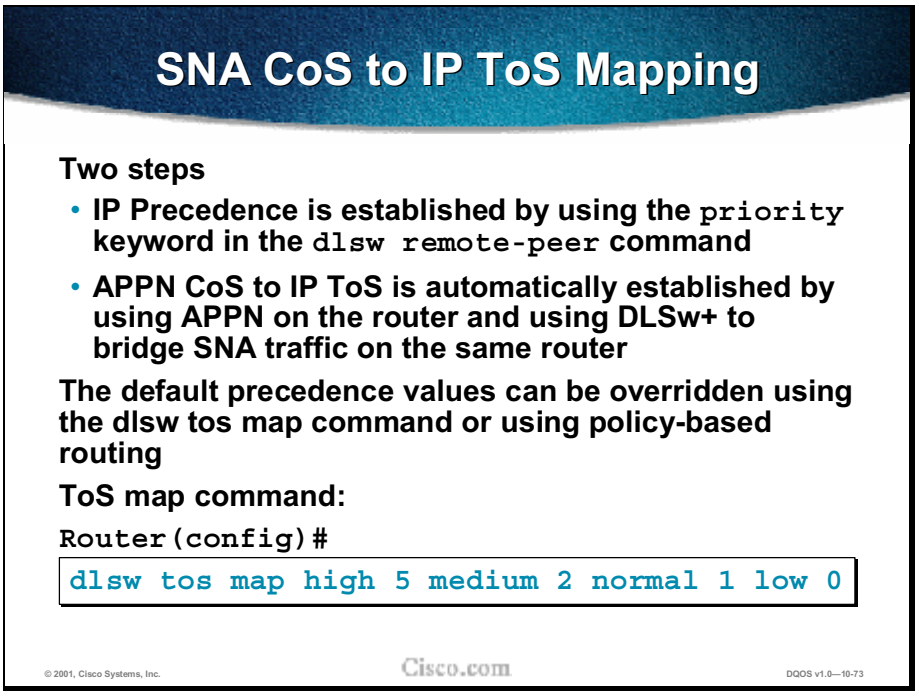
By default, the vast majority of data application traffic should be classified with a DSCP value of "best effort," or 0. However, the case can arise when enterprise business policies dictate the need to differentiate data traffic flows for enhanced network service. When this is required, all elements of the application traffic patterns should be analyzed to determine the maximum packet delay, minimum packet loss, and maximum possible bandwidth requirements. When this is completed, the data traffic can be classified with one of the assured forwarding (AF) classes, preferably AF11, AF12, AF13, AF21, AF22, or AF23. This ensures that the data traffic will not interfere with the Voice and Video over IP traffic policies.



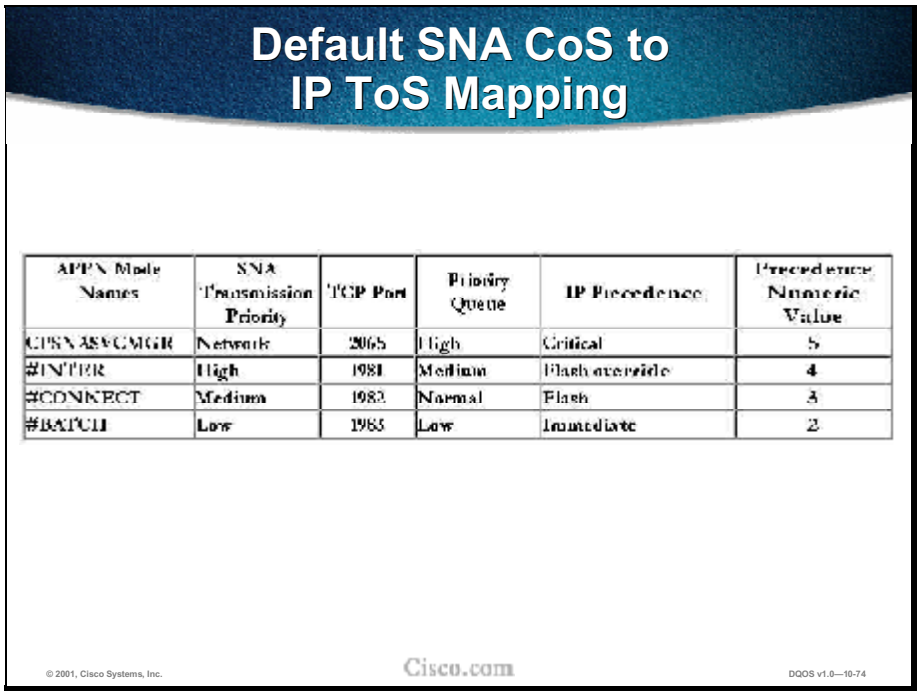
To preserve SNA response times, interactive SNA traffic must be separated from batch traffic and the interactive traffic must be prioritized. For example, if IND$FILE is present in a network, then even if SNA is high priority, the batch SNA traffic will negatively impact the response times for interactive SNA traffic.

Where dependent logical units (LUs) are used, the only means the router has to separate traffic is the SNA network addressable unit (NAU), which is otherwise referred to as the LU address or the LOCADDR. Architecturally SNA refers to this class of LUs as “dependent” because of their dependency on the VTAM SSCP. All LU type 2, all LU type 0, and some LU type 6.2 applications are dependent. The separation of batch from interactive is further complicated if LU 6.2 parallel sessions is used, because the NAU or LU address is dynamically allocated. Parallel session support is referred to as “independent” LUs because they do not require the assistance of the VTAM SSCP; it is APPN peer-to-peer communications.

SNA class of service (CoS) to DLSw+ type of service (ToS) resolves this problem. This works by mapping SNA CoS to IP ToS (or Precedence).



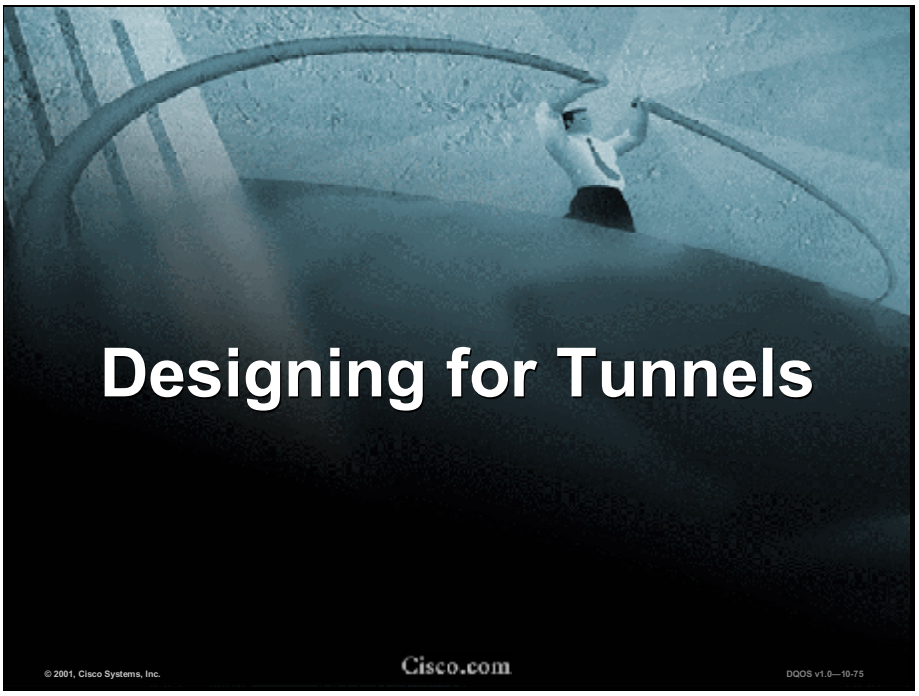
The other method of mapping SNA CoS to IP ToS is when Advanced Peer-to-Peer Networking (APPN) is being routed and uses the DLSw+ VDLC interface as a link layer connection. This is not the same as bridging APPN over DLSw+. In this situation, the router is a full APPN network node routing SNA. The APPN network node participates fully in the session establishment process, where it has easy access to session priority. When the network node accesses link layer services, the APPN CoS (APPN transmission priority) is mapped to a DLSw+ port. The table lists the default mappings. There are conveniently four APPN transmission priorities and four DLSw+ priority ports as a result of the DLSw+ **priority** keyword.

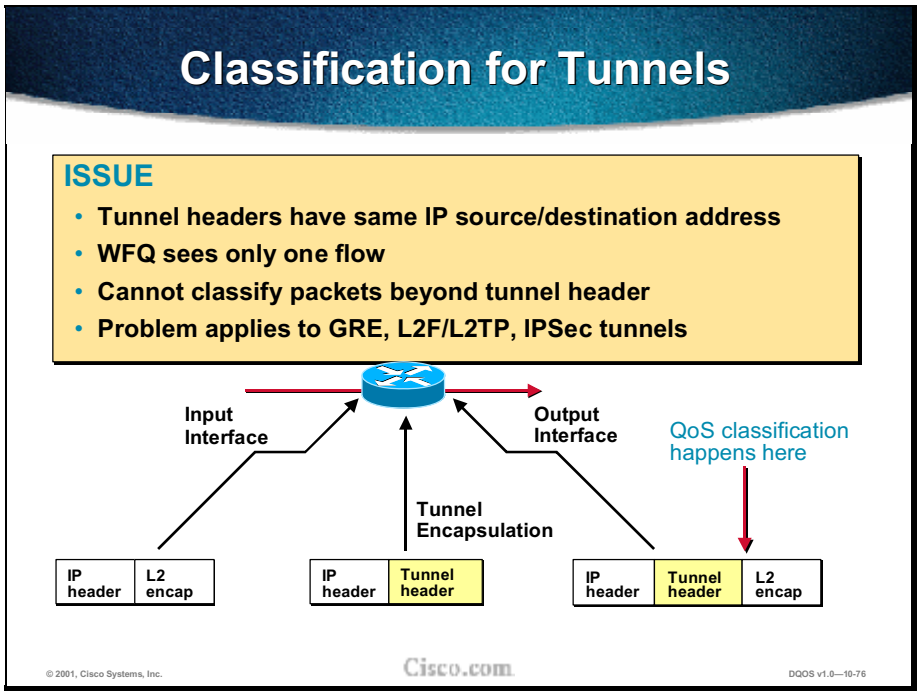


SNA CoS to DLSw+ ToS can be used in conjunction with Dependent LU Requester (DLUR) to migrate old dependent logical units (LUs) to APPN. However, to map SNA CoS to DLSw+, ToS requires the use of APPN and DLSw+ on the same router and APPN in the enterprise network. Features to separate SNA batch and interactive traffic without using APPN CoS to DLSw+ ToS exist in Cisco IOS. Using the LU address, the router can redirect batch traffic from a particular LU (for example, LOCADDR 04) to a specific DLSw+ TCP port (1980) and redirect interactive traffic (LOCADDR 02) on a different DLSw+ TCP port (2065). Once the traffic is separated on different TCP flows, it is simply a matter of using any one of the various queuing techniques to establish packet scheduling priorities based on the DLSw+ TCP ports.

Special considerations must be made if IND$FILE is present in a network. IND$FILE is a very old, very inelegant method of transferring data. If IND$FILE is in use, there is no way to stop a user from starting a file transfer on any SNA 3270 session that could just as well be interactive. The best advice for customers trying to preserve SNA response times is to recommend that they discontinue the use of IND$FILE and move to a more advanced method of file transfer.

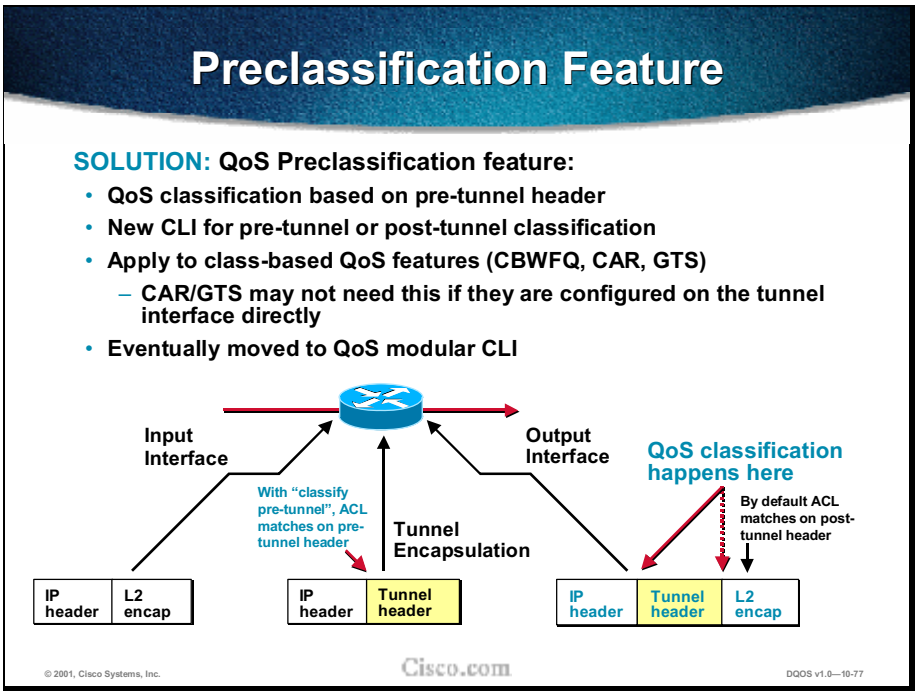
Currently, there is no way to change these default mappings. For example, the mode CPSNASCVMG is assigned network transmission priority that is mapped to TCP port 2065, the mode #INTER is given high priority that is mapped to port 1981 at IP Precedence flash override, #CONNECT is given medium priority, and #BATCH is given low priority. In addition **bind** commands and IPM (pacing messages) are given their network transmission priority by APPN.





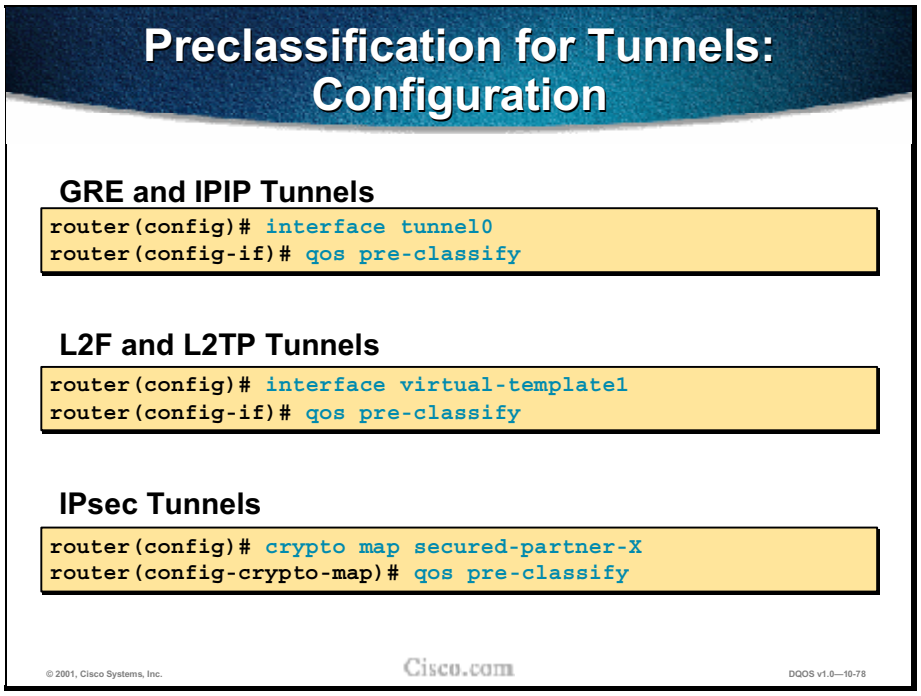
When packets are encapsulated by tunnel or encryption headers, QoS features are unable to examine the original packet headers and correctly classify the packets. Packets traveling across the same tunnel have the same tunnel headers, so the packets are treated identically if the physical interface is congested.

With the growing popularity of VPNs, the need to classify traffic within a traffic tunnel is gaining importance. QoS features have historically been unable to classify traffic within a tunnel.

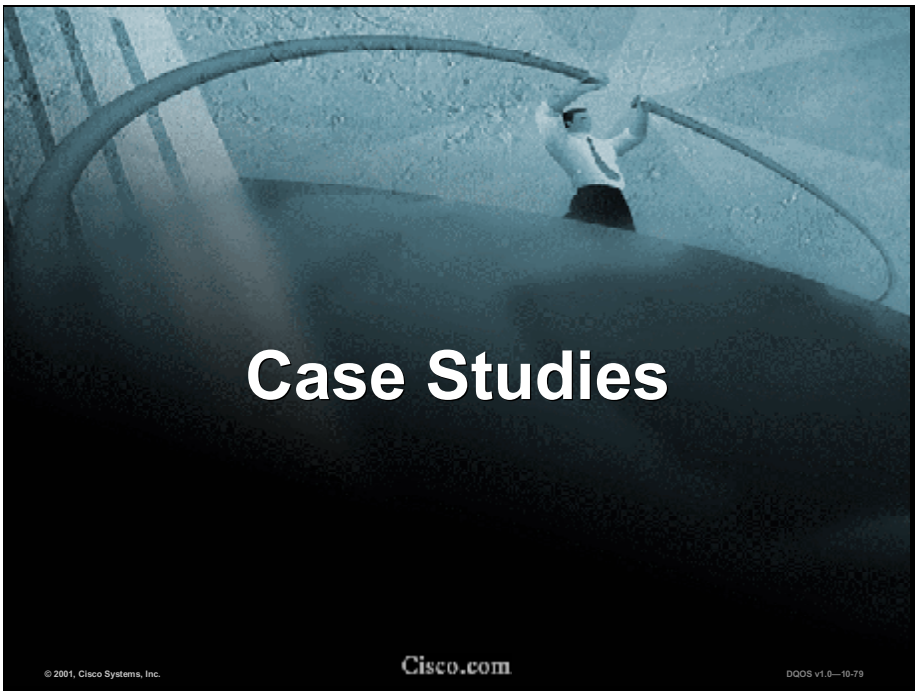


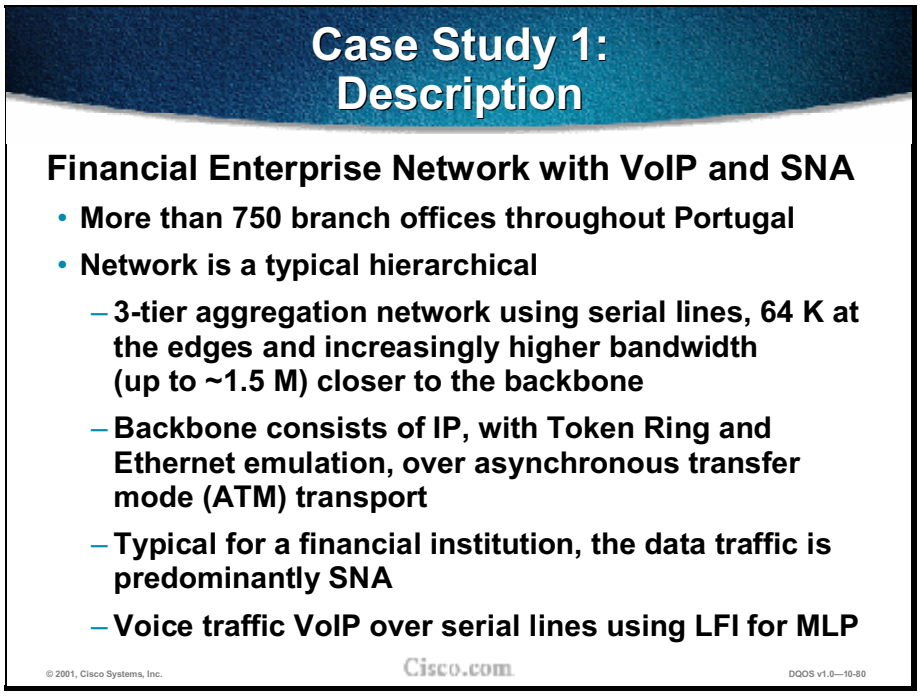
With the introduction of the quality of service for Virtual Private Networks (QoS for VPNs) feature, packets can now be classified before tunneling and encryption occur. The process of classifying features before tunneling and encryption is called preclassification.

The QoS for VPNs feature is designed for tunnel interfaces. When the new feature is enabled, the QoS features on the output interface classify packets before encryption, allowing traffic flows to be adjusted in congested environments. The end result is more effective packet tunneling.



The **qos pre-classify** command is applied to a tunnel or VPN interface. This enables the DSCP/TOS field of the tunnelled packet to be copied to the DSCP/TOS field of the outer packet.



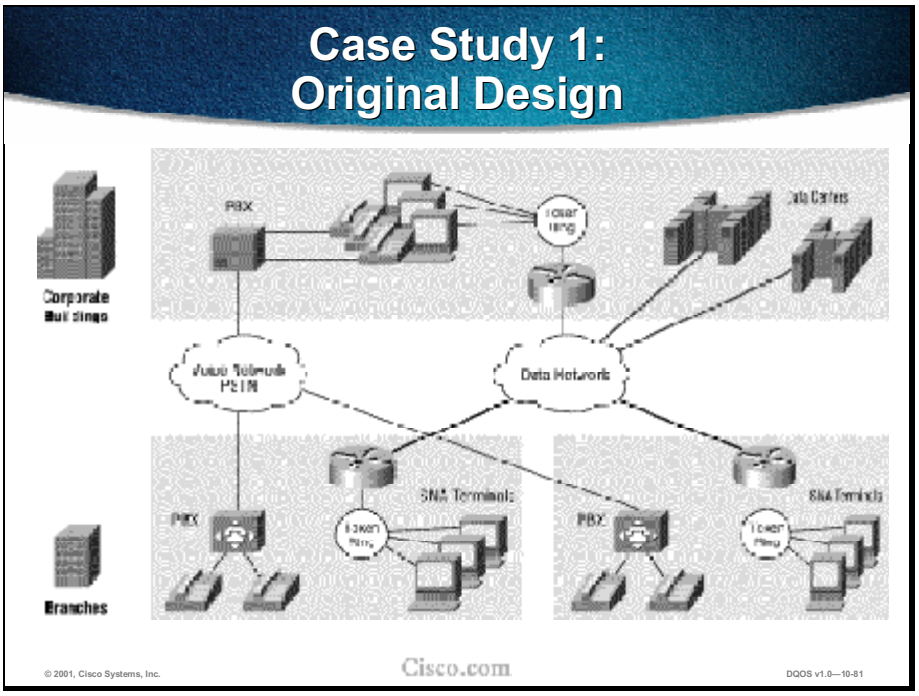


Introduction

Headquartered in Lisbon, Portugal, Grupo Mundial-Confiança (GMC) is a financial group comprising an insurance company and four banks (Banco Pinto & Sotto Mayor—BPSM, Banco Totta & Açores—BTA, Crédito Predial Português—CPP, Chemical), all old, established firms within Portugal. Together, this firm has more than 750 branch offices throughout Portugal, approximately half of which are outside the major urban areas of Lisbon and Porto.

Its network is a typical hierarchical, three-tier aggregation network using serial lines, low bandwidth (64 K) at the edges, and increasingly higher bandwidth (up to ~1.5 M) closer to the backbone. The backbone consists of IP, with Token Ring and Ethernet emulation, over asynchronous transfer mode (ATM) transport. Also typical for a financial institution, the data traffic is predominantly systems network architecture (SNA). Voice traffic is Voice over IP (VoIP) over serial lines using multilayer point-to-point protocol (MLPPP) for fragmentation and interleaving.

This case study can be found at :
http://www.cisco.com/warp/public/cc/pd/rt/2600/profiles/mfien_bc.htm



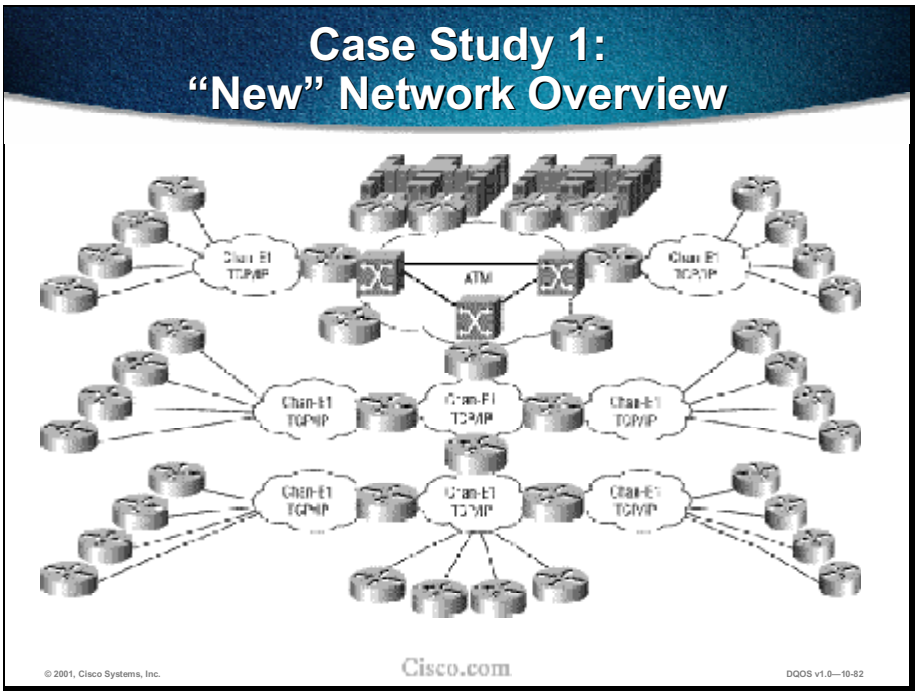
Original Design: Separate Voice and Data Networks

The diagram shows a general view of the “old” network topology, where phones and data terminals from the same work area were connected to two separate networks.

Design Objectives

In presales meetings, the customer expressed a need for a new generation data/voice/video network that would meet the following objectives:

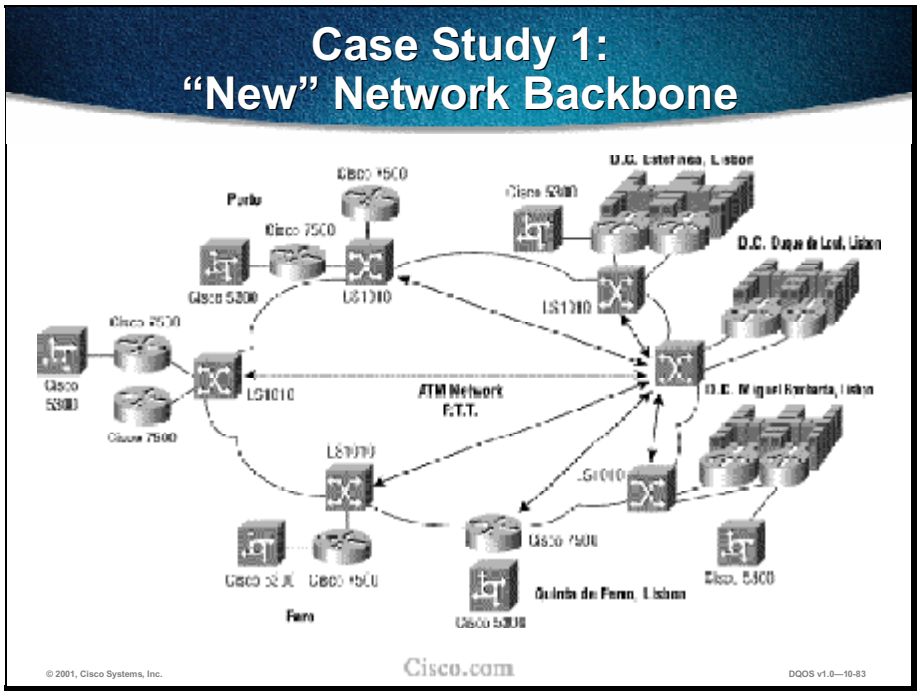
- The new network should be standards based and able to host different flavors of networking protocols/products. The initial protocols required were TCP/IP and SNA. SNA must be fully controlled (response times and bandwidth dedication) and the introduction of any additional protocol should not impact SNA performance.
- Migration of the network should not impact the current SNA/37xx/NewBridge MUX infrastructure.
- It should have full connectivity between branch offices (TCP/IP and SNA).
- It should have full voice and fax connectivity between branch offices (VoIP).
- It should provide higher availability and performance.
- Network management should be integrated and centralized.
- It should have direct inward dialing (DID) functionality.
- Mean Opinion Score (MOS) of 4 is required (considering private switched telephone network [PSTN] as 5, which is the level of quality the customer is accustomed to).



Steps in Implementation and Cutover

The migration of the old network to the new network happened in several stages. Traffic was stabilized after each major step. Installation of the network took place over several months, but was intensely focused in the fourth calendar quarter of 1998.

Data traffic was actively supporting the business, and the network had to be cut over with minimal disturbance to the old network, requiring significant reconfiguration of old and new network elements at various times. Voice was slightly easier in that the old and new networks could run in parallel, and voice on the new network could be turned up when convenient without impacting business traffic.



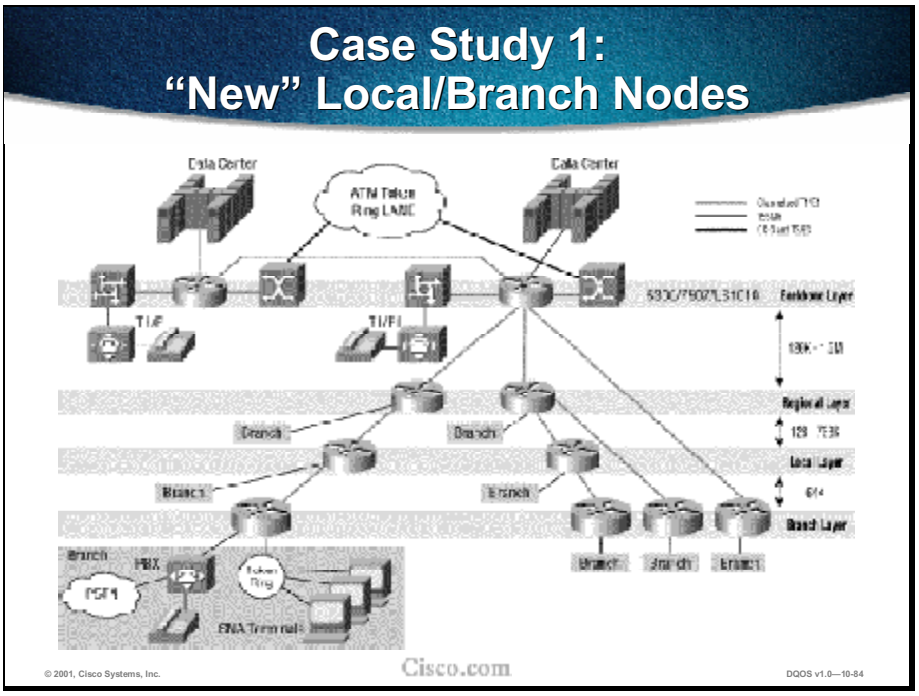
Steps in Implementation and Cutover (continued)

1. Install the new data network, backbone first, followed by city branch locations, then rural locations.
2. Relocate and consolidate the five data centers into one location—redeploy the Cisco 7500s and reconfigure the old network to access the new paths to the data centers.

Backbone

The figure above shows the backbone of the network covering four major geographical regions, connecting into the data center location in the Lisbon area. The connectivity in the backbone is ATM from the local service provider (SP), with leased-line channelized E1 connections for backup. The ATM bandwidth between the regions is approximately 5 Mbps. Each location has dual Cisco 7500s, an LS1010 for ATM connectivity, and a Cisco 5300 to service traffic from the corporate building via two E1s to the PBX.

The backbone combines an ATM infrastructure with a router infrastructure, as shown in figure above. This preserves the ATM backbone for voice and SNA traffic while the multiprotocol traffic (TCP/IP) is being routed by the overlapping router infrastructure. From there the SNA traffic is propagated to the data center by DLSw and is terminated at the Cisco 7500s by Channel Interface Processors (CIPs). The LS1010s are connected to the public ATM network via virtual path (VP) tunneling over ATM constant bit rate (CBR) VPs with different bandwidths. The Cisco 7500s are connected to the LS1010 with 155M interfaces, aggregating the traffic from the other network layers.

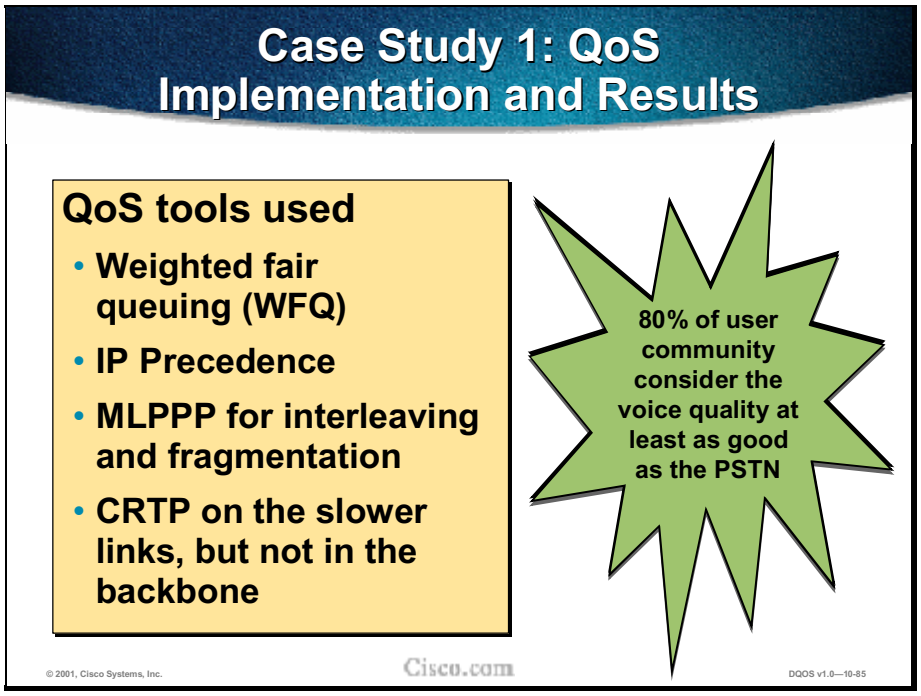


Steps in Implementation and Cutover (continued)

3. Deploy a pilot of 20 VoIP nodes with a flat dial plan (install BRI modules into Cisco 2600/3600s already running data).
4. Install BRI voice hardware in remaining Cisco 2600/3600s, but do not turn up voice.
5. Implement gatekeeper/gateway (GK/GW) design and change dial plans of active nodes.
6. Ramp up voice on all nodes within the GK/GW design.

Regional, Local, and Branch Nodes

The over 750 branches are divided into three aggregation tiers: regional, local, and branch. The branch layer routers are Cisco 2612s and in denser city areas frequently connect directly into the backbone (Cisco 75xxs); in the more geographically dispersed areas, they connect in via a local layer or via both a local and regional layer Cisco 3640.

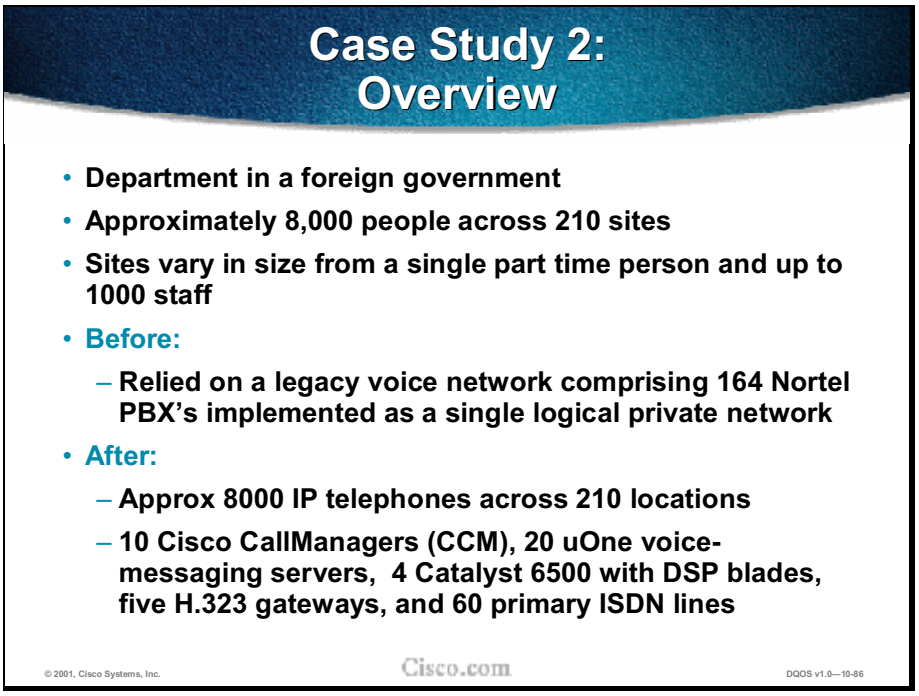


Voice QoS Implementation

Both good voice quality and adequate SNA response times are important in this network. The following QoS mechanisms were implemented to prioritize voice and SNA interactive traffic over SNA batch and other IP traffic:

- Weighted fair queuing (WFQ)
- IP Precedence:
  - VoIP: Highest
  - Interactive SNA: High
  - Other traffic: Low
- MLPPP for interleaving and fragmentation
- real-time transport protocol [RTP] header compression (CRTP) on the slower links, but not in the backbone (connections into the Cisco 7500)

Initial survey results from the user community indicate that more than 80 percent consider the voice quality at least as good as the PSTN.

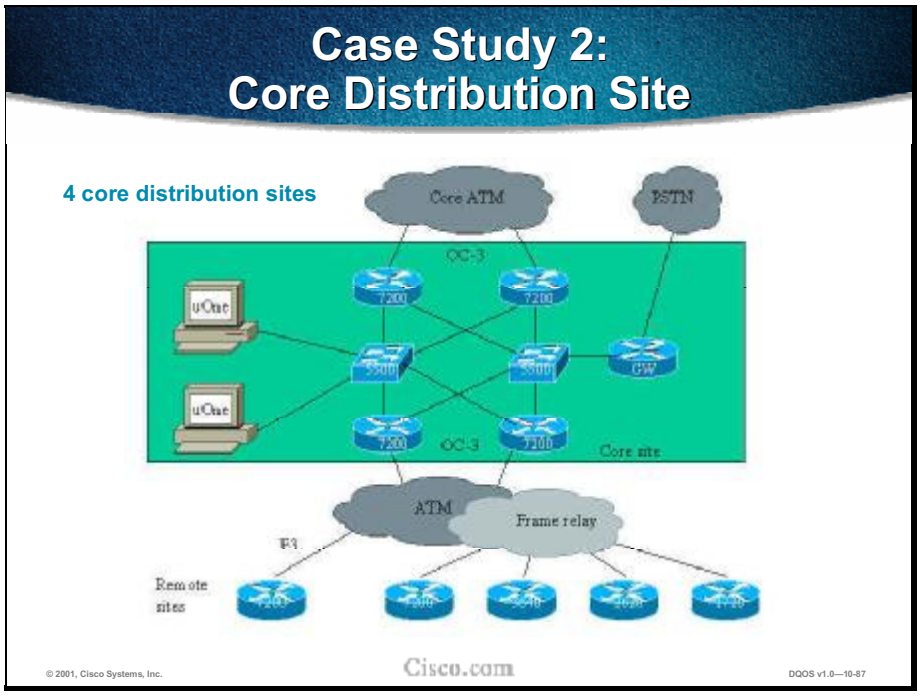


Introduction

The Ministry of Social Policy (MoSP) is a New Zealand national government department responsible for social benefits, community services, and youth affairs. MoSP employs approximately 8,000 people across 210 sites. These sites are spread across the entire country and vary in size from a single part-time person up to 1,000 staff members.

Before moving to an AVVID solution, MoSP relied on a legacy voice network that comprised 164 Nortel PBXs implemented as a single logical private network.

After the AVVID deployment, the network comprises approximately 8,000 IP telephones across some 210 locations throughout New Zealand. Ten Cisco CallManagers (CCM), 20 uOne voice-messaging servers, four Catalyst 6500 with DSP blades, five H.323 gateways, and 60 primary ISDN lines were deployed.

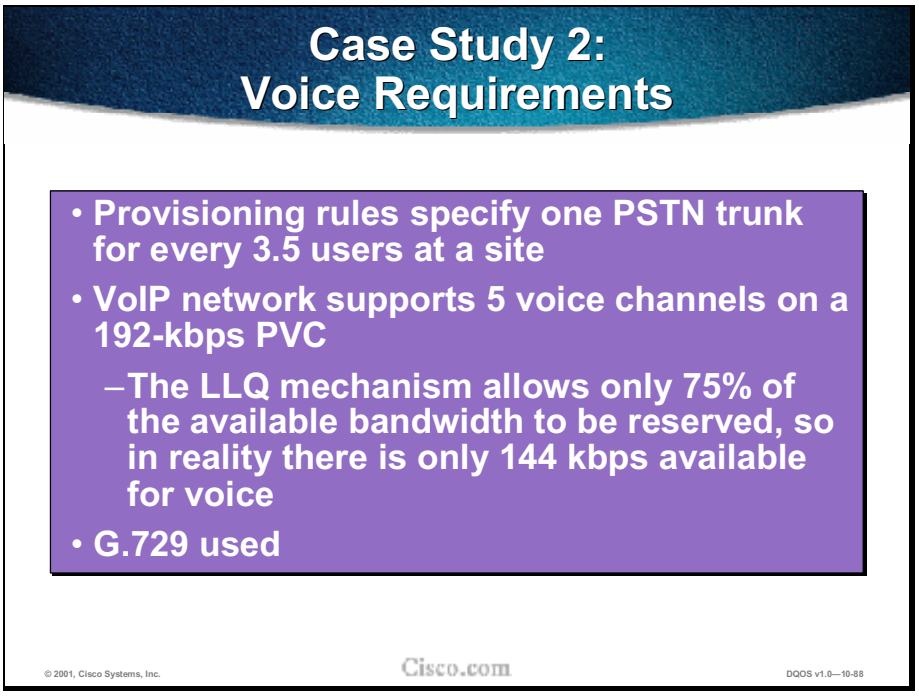


The MoSP consists of four core sites and approximately 210 remote sites. The core sites are located in Auckland, Hamilton, Wellington, and Christchurch. Each core site has two distribution routers and two core routers connected via two backbone switches.

The core routers connect to core routers at the other core sites via ATM. The distribution routers connect to remote sites via FR/ATM Interworking. Remote sites have a single Frame Relay attached router. Two PVCs connect back to the two distribution routers at the nearest core site. There are no servers or phone lines at the remote site, only IP Phones

All voice equipment is located at the four core sites. Wellington has two CCM clusters, one for Wellington and one for Palmerston. The other core sites have one CCM cluster each. A number of uOne voice-mail servers are associated with and collocated with the CCM clusters. Three of the four core sites host PSTN gateways. Calls from a remote site to and from the PSTN are routed back to the core site and out the gateway.

The figure illustrates one of the four core/distribution sites. These sites are physically located inside the carriers exchange, and hence there are no staff or IP Phones at these sites. The Frame Relay and ATM routers at the bottom of the illustration represent remote sites. Some of the remote sites are quite large, with up to 1,000 staff members, and these are ATM attached back to the distribution layer. Smaller sites are Frame Relay attached at the remote end and ATM attached at the distribution end.

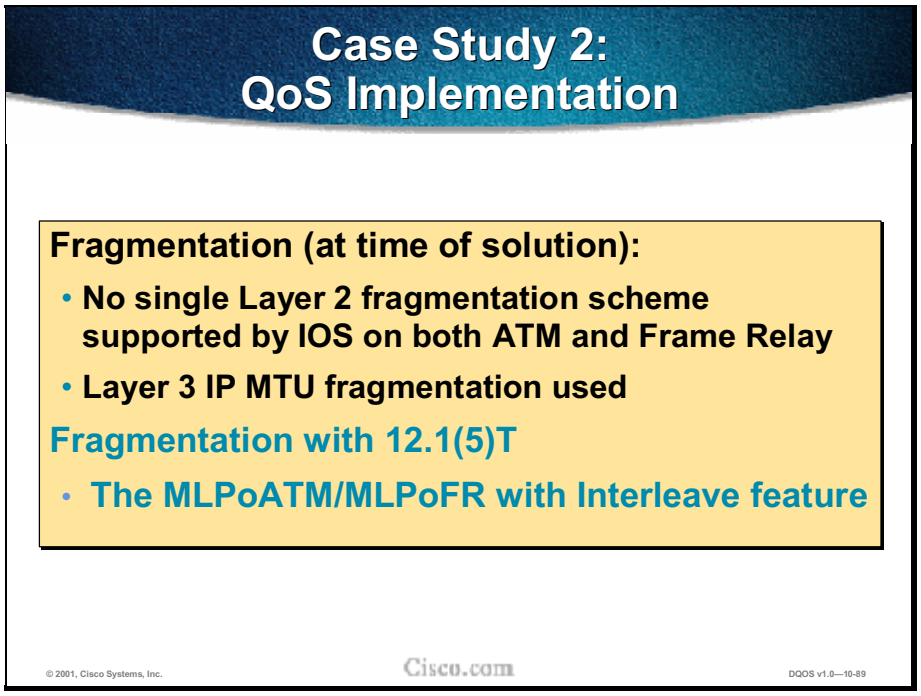


The MoSP provisioning rules specify one PSTN trunk for every 3.5 users at a site. It was also a MoSP design requirement that the VoIP network support five voice channels on a 192 kbps PVC. The LLQ mechanism allows only 75 percent of the available bandwidth to be reserved, so in reality there is only 144 kbps available for voice. The default VoIP G.729 payload is 20 bytes or 20 milliseconds (ms). With a TCP/IP overhead of 40 bytes, each VoIP packet ends up being 60 bytes. Thus two ATM cells are required per VoIP packet. The bandwidth requirements for one call on ATM and Frame Relay are therefore: Frame Relay: 50 pps x 60 bytes/pkt x 8 bits/byte = 24 kbps; ATM: 50 pps x 2 cells/pkt x 53 bytes/cell x 8 bits/byte = 42 kbps. So the bandwidth usage for five simultaneous calls is: Five calls using G.729 codec consumes 26 kbps x 5 = 130 kbps for the Frame Relay end. Five calls using G.729 codec consumes 42 kbps x 5 = 210 kbps for the ATM end. This created a problem since a 192-kbps ATM PVC would be oversubscribed carrying five calls.

To work around this, the sampling rate was changed to be fixed at 30 ms across the full network. This increased the RTP packet size to 70 bytes, which still fits into two ATM cells. At the same time the packet rate was reduced from 50 pps to 33 pps, and hence the Frame Relay and ATM bandwidth requirements were reduced to: Frame Relay: 33 pps x 70 bytes/pkt x 8 bits/byte = 20 kbps; ATM: 33 pps x 2 cells/pkt x 53 bytes/cell x 8 bits/byte = 27 kbps.

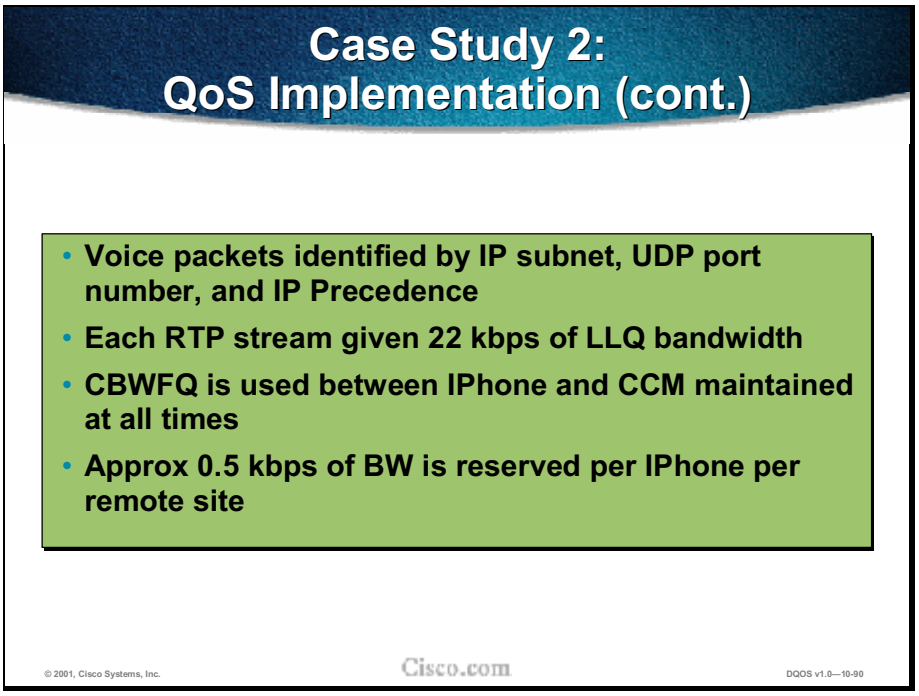
Now the bandwidth required to carry five simultaneous calls is:

Five calls using G.729 codec consumes 20 kbps x 5 = 100 kbps for the Frame Relay end. Five calls using G.729 codec consumes 27 kbps x 5 = 135 kbps for the ATM end. This will now fit on a 19-kbps PVC.



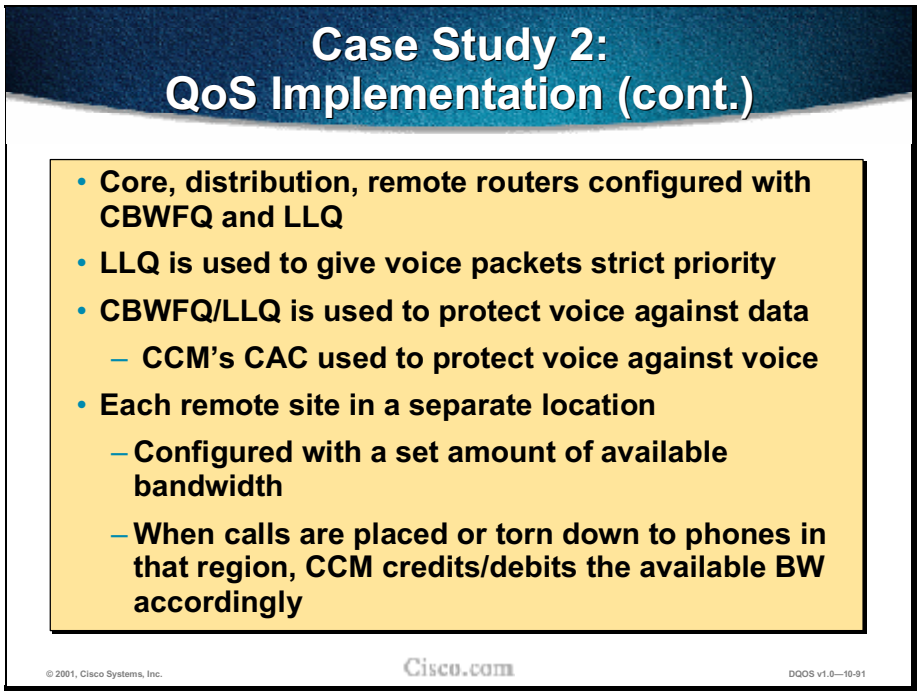
Fragmentation in the MoSP environment is an issue. No single Layer 2 fragmentation scheme was supported by IOS on both ATM and Frame Relay. The only solution that was available was Layer 3 IP MTU fragmentation. During early proof-of-concept testing, the MTU size was set to 300 bytes, but this caused the IP Phones to display “DSP Keepalive Timeout” messages. These symptoms were a result of the IP Phones not supporting packet reassembly. The compromise solution has been to fragment to 600 bytes only, which is more than the maximum skinny packet size of 480 bytes. This ensures that Skinny is never fragmented.

The right solution has arrived in 12.1(5)T, when the MLPoATM/MLPoFR with interleave feature is released. There are no known plans for the IP Phones to support reassembly.



Voice packets are identified by IP subnet, UDP port number, and IP Precedence. To achieve this, all voice subnets were assigned so that they were easily recognizable, in that all voice subnets have bit 4 set to '1' in the third octet (0.0.16.0). This very strict classification scheme was chosen to ensure that nonvoice flows were not subjected to LLQ by mistake, since this would break the voice protection. Each RTP stream is given 22 kbps of LLQ bandwidth. This is less then the 25 kbps consumed on ATM, but this works because the low latency queuing happens before ATM adding the SAR overhead.

CBWFQ is used to ensure that Skinny signalling between IP Phone and CCM is maintained at all times. Approximately 0.5 kbps of bandwidth is reserved per IP Phone at a given remote site. This is required only on the CBWFQ between distribution and remote router. There is no Skinny traffic travelling across the core WAN links. There would be H.323 intercluster traffic as well as H.323 traffic from gateway to CCM. But this is not currently being given special treatment, and to date this has not caused any issues.

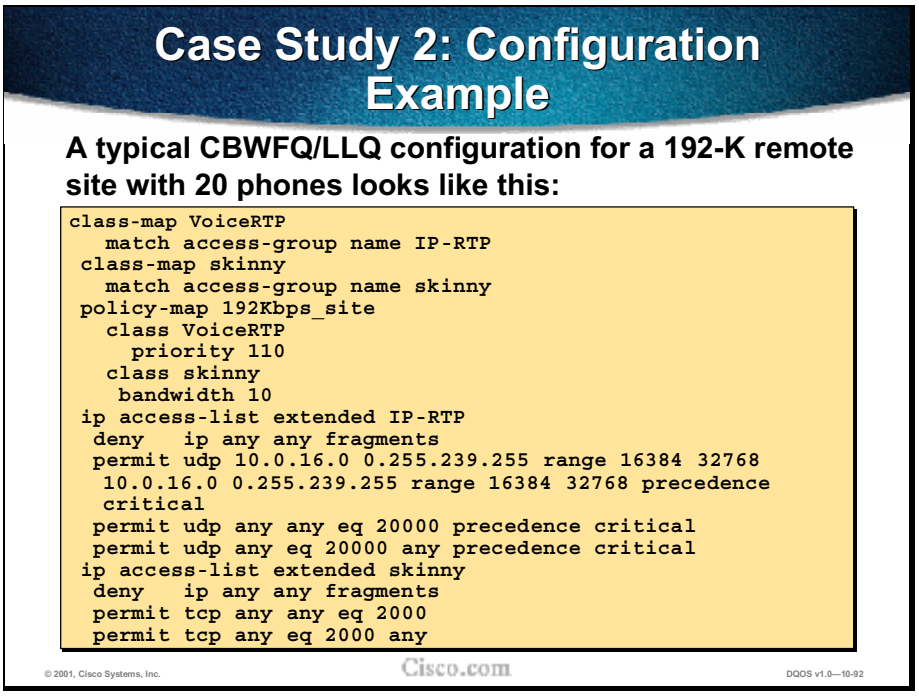


Two separate features are used in the MoSP network to guarantee the integrity of voice across the WAN. CBWFQ/LLQ is used to protect voice against data, while CCM's location-based connection admission control (CAC) is used to protect voice against voice.

The MoSP CCM design has each remote site in a separate location. That location is configured with a set amount of available bandwidth. When calls are placed or torn down to phones in that region, CCM credits/debits the available bandwidth accordingly. The use of locations-based CAC dictated a CCM design with one active CCM in each cluster. This is a requirement because all calls need to go through one CCM in order for the bandwidth accounting to work.

There is no intercluster CAC. The assumption here is that there is enough bandwidth available in the core to carry the load offered at all times. While this has proven true, another reason for some kind of CAC has surfaced. Imagine that user A in cluster 1 forwards his/her phone to a user B in cluster 2, and user B then forwards his/her phone back to A. When someone calls A or B, then a call-routing loop happens. The CCMs have no mechanism for detecting this condition, and the loop will continue 360 times. This is the maximum number of calls across an intercluster H.323 trunk. The impact on CCM CPU is significant. This problem remains unresolved at the time of writing.

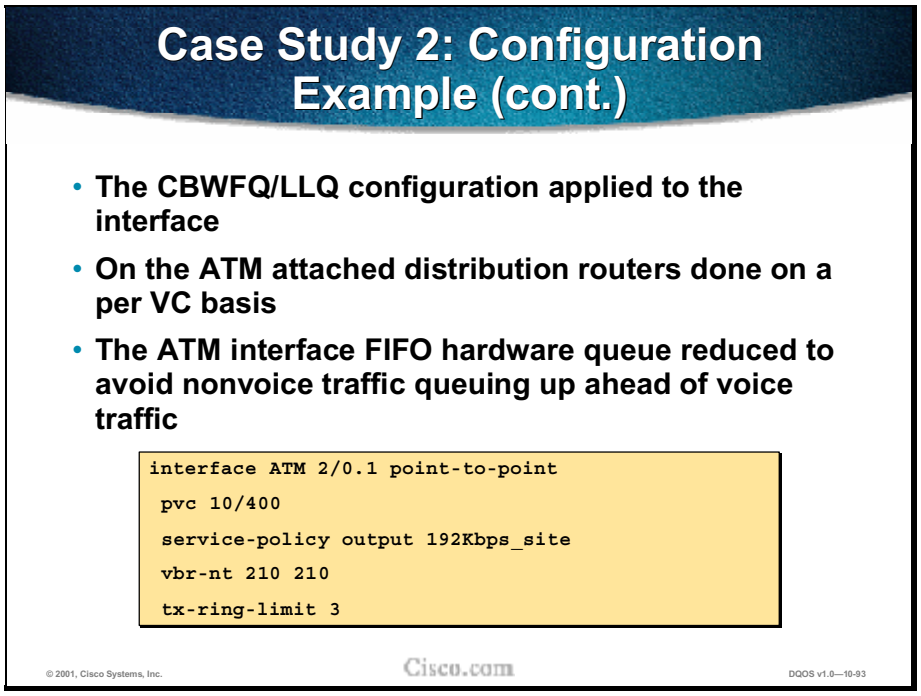
Core, distribution, and remote routers are all configured with CBWFQ and LLQ. LLQ is used to give voice packets strict priority.



A few notes about the configuration in the figure.

The first statement in the IP-RTP access-list denies a packet if it is an IP fragment. This is important because the second statement looks at Layer 4 attributes. A fragment no longer has an L4 header, and as a result the access-list will always permit the fragment and place it in the LLQ by mistake. In the MoSP network this is especially critical because IP MTU fragmentation is being done, and there are many fragments floating around. These could potentially consume all the LLQ bandwidth and compromise voice quality. But it is good practice to do the same thing in any network, since fragmentation is an integral part of IP and could be happening in the Internet before reaching the corporate network.

Also notice the third and fourth statements in the IP-RTP access-list. They are used to classify RTR/PIM traffic.



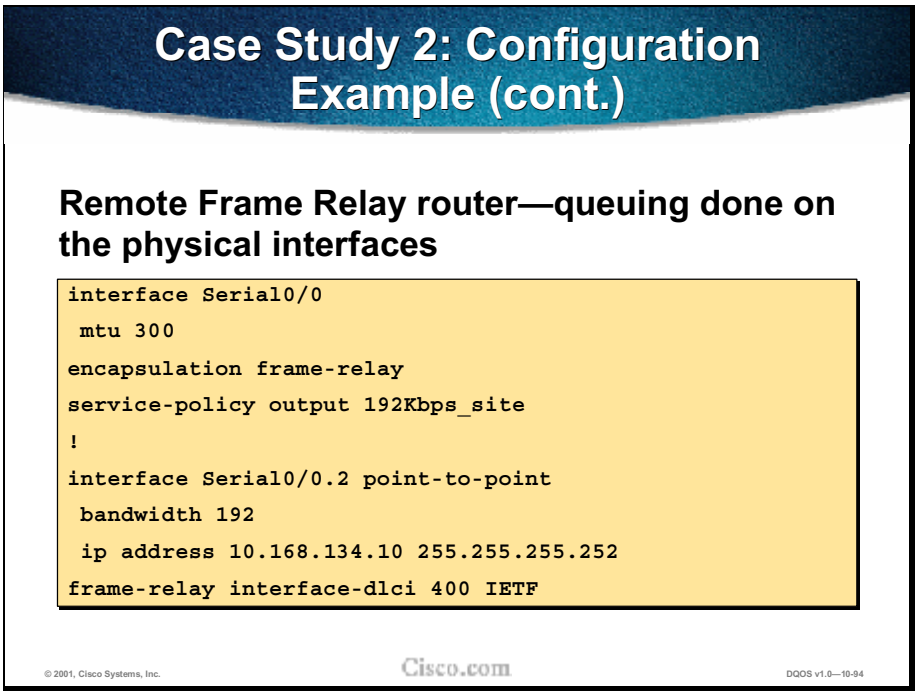
The CBWFQ/LLQ configuration has to be applied to the interface. On the ATM attached distribution routers this is done on a per-VC basis. The ATM interface also has to have the FIFO hardware queue reduced to avoid nonvoice traffic queuing up ahead of the voice traffic. The rule of thumb for this is:

tx-ring-limit = bandwidth / 64 kbps

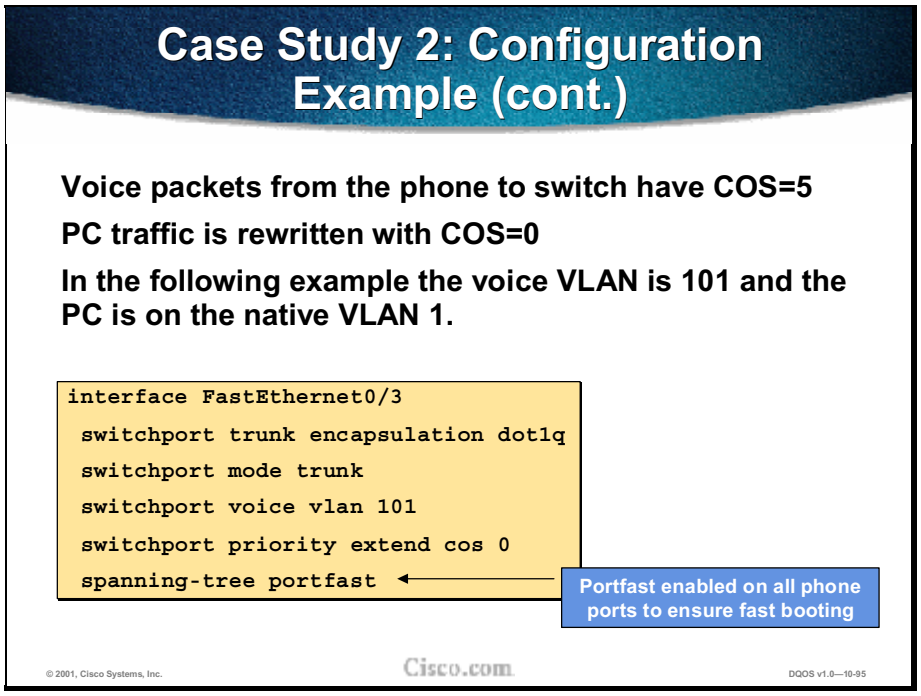
For simplicity the MoSP network has only three settings:

PVC speed	tx-ring-limit
Speed <= 2M	3
2M < speed <= 25M	400
25M < speed	1200

Per-VC queuing on Frame Relay is not supported in 12.1 mainstream. So on the remote Frame Relay router the queuing is done on the physical interfaces. This works because only one of the two PVCs is active at any given point in time. The tx-ring-limit on a serial interface is automatically adjusted to an appropriate level when fancy queuing is enabled.

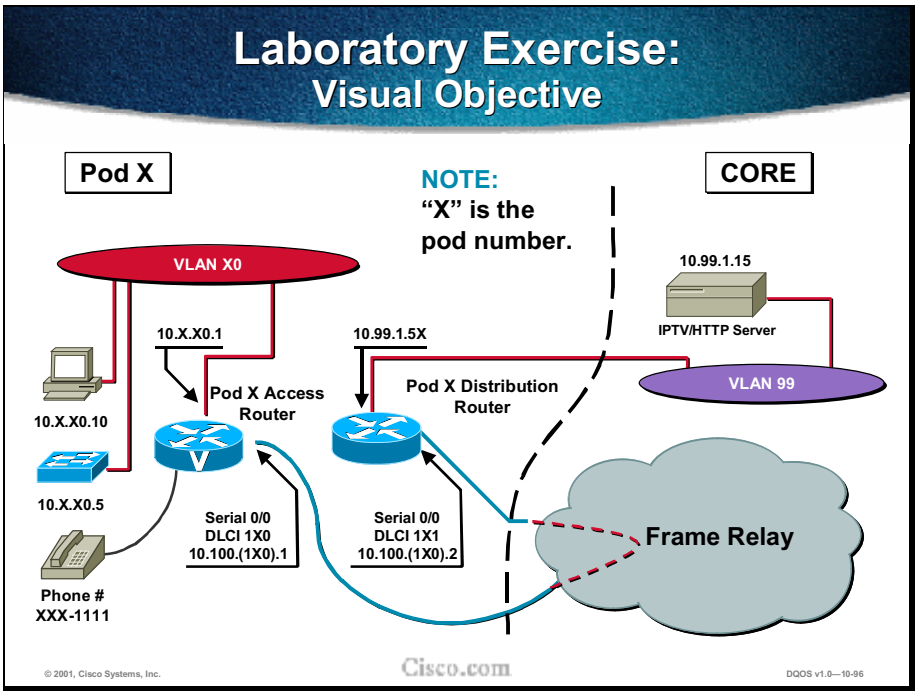


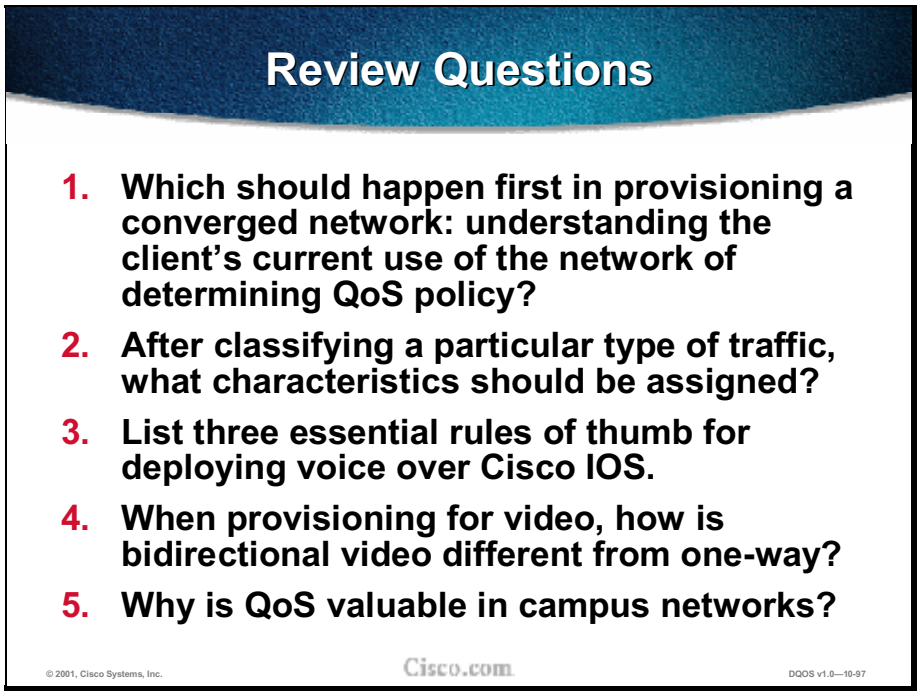
Per-VC queuing on Frame Relay is not supported in 12.1 mainstream. On the remote Frame Relay router the queuing is done on the physical interfaces. This works because only one of the two PVCs is active at any given point in time. The tx-ring-limit on a serial interface is automatically adjusted to an appropriate level when fancy queuing is enabled.



Most 2900XL switches run 12.0(5)XU and have QoS features enabled. IP phones connect to a 2900XL via 802.1q trunking, with a PC connected to the back of the phone. Phone and PC are on different subnets. Voice packets from the phone to switch have COS=5, while all PC traffic is rewritten with COS=0. Based on CoS, the switch then puts voice traffic in one queue and data in another. In the following example the voice VLAN is 101 and the PC is on the native VLAN 1. Portfast is enabled on all phone ports on the switch to ensure fast booting.

At that time IOS used in the network did not support ToS to CoS mapping. This means that both voice and data traffic coming from router to switch has COS=0. Hence the switch cannot queue voice packets in a separate queue outbound on a phone port. Voice packets could therefore potentially be dropped if the PC connected to the back of an IP phone is receiving a lot of data from a server on another switch port. As a workaround for this the router switch port is configured with a default COS=5. This means all traffic from the router will be given a high QoS, including data. This is acceptable because the router is attached to a slow WAN link and data from here can arrive only at a limited pace.





Answer these questions:

Q1) Which should happen first in provisioning a converged network: understanding the client's current use of the network or determining QoS policy?

Q2) After classifying a particular type of traffic, what characteristics should be assigned?

Q3) List three essential rules of thumb for deploying voice over Cisco IOS.

Q4) When provisioning for video, how is bidirectional video different from one-way?

Q5) Why is QoS valuable in campus networks?

Answers to the review questions appear in Appendix B.

Summary

Upon completing this module, you should be able to:

- **Design a converged multiservice network to provide proper QoS for voice, video, and data traffic**

11

Course Wrap-Up

Overview

The purpose of this class has been to teach the IOS quality of service (QoS) tools up to and including Cisco IOS Software Release 12.1(5)T. The class has also provided hands-on experience of using the QoS tools in labs through out the week. You should now be able to effectively employ these tools in managing voice, video, and mission-critical network traffic.

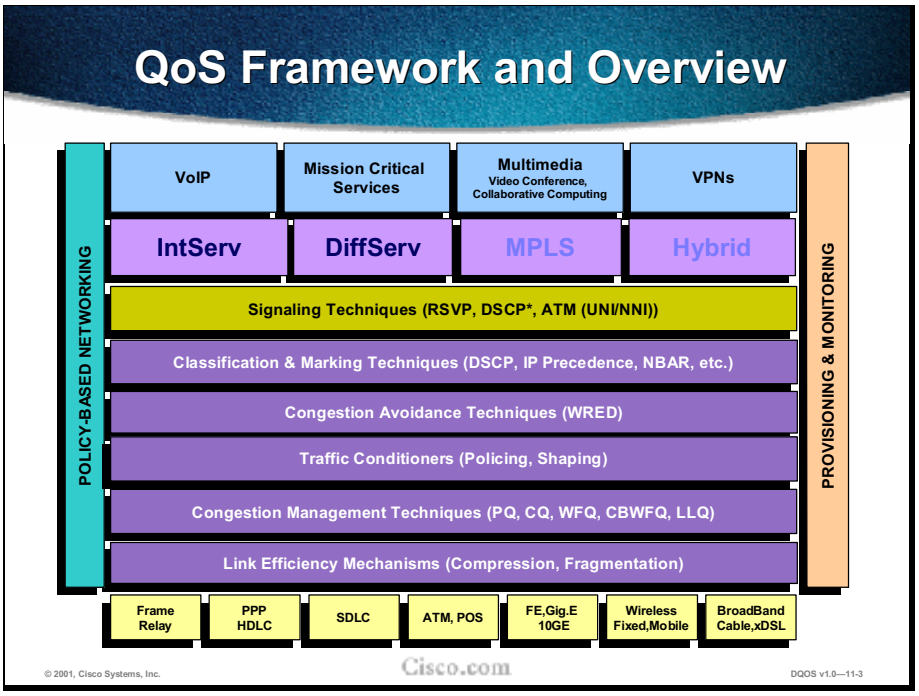
Objectives

The scope of this class includes the following:

- Classification and marking tools
- Congestion management tools
- Congestion avoidance tools
- Link efficiency tools
- Traffic conditioning tools
- Call admission control for voice and video
- QoS management tools

Outline

This chapter reviews what has been learned in this weeklong course.



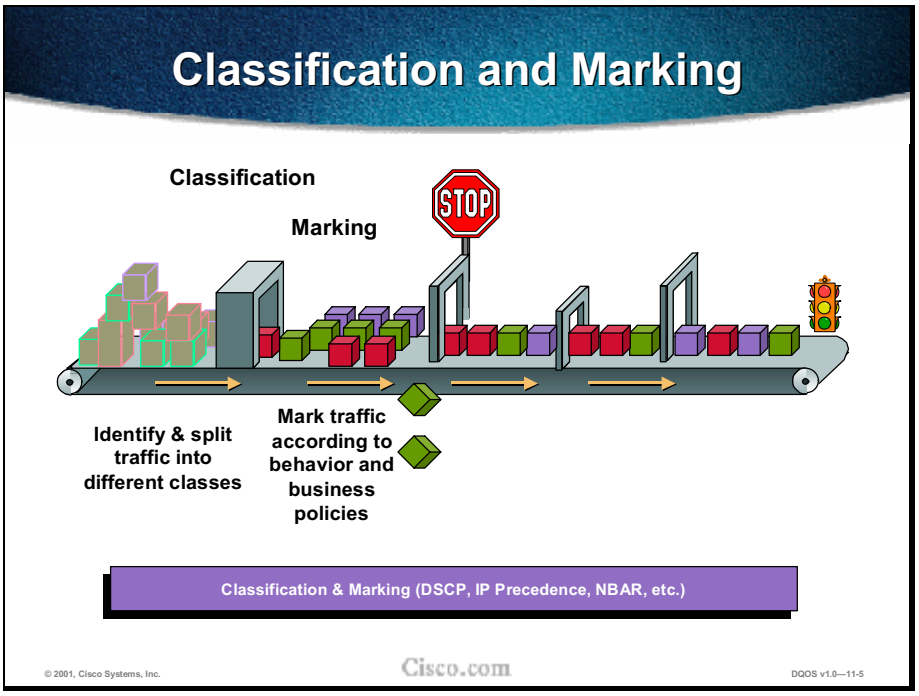
There are several benefits of implementing QoS in the enterprise network.

The QoS framework consists of:

- Signaling techniques
- Classification and marking tools
- Congestion avoidance tools
- Congestion management tools
- Traffic conditioners, the shaping and policing tools
- Link efficiency tools
- Provisioning mechanisms
- Signaling mechanisms
- Call admission control for voice and video

Two different methods for implementing QoS are IntServ and DiffServ.

Refer to Chapter 2, QoS Overview, for details.



Classification and marking is used for making the packets accessible to QoS handling within the network.

Classification is the selection of traffic to be marked, and marking is setting a value for the classified traffic.

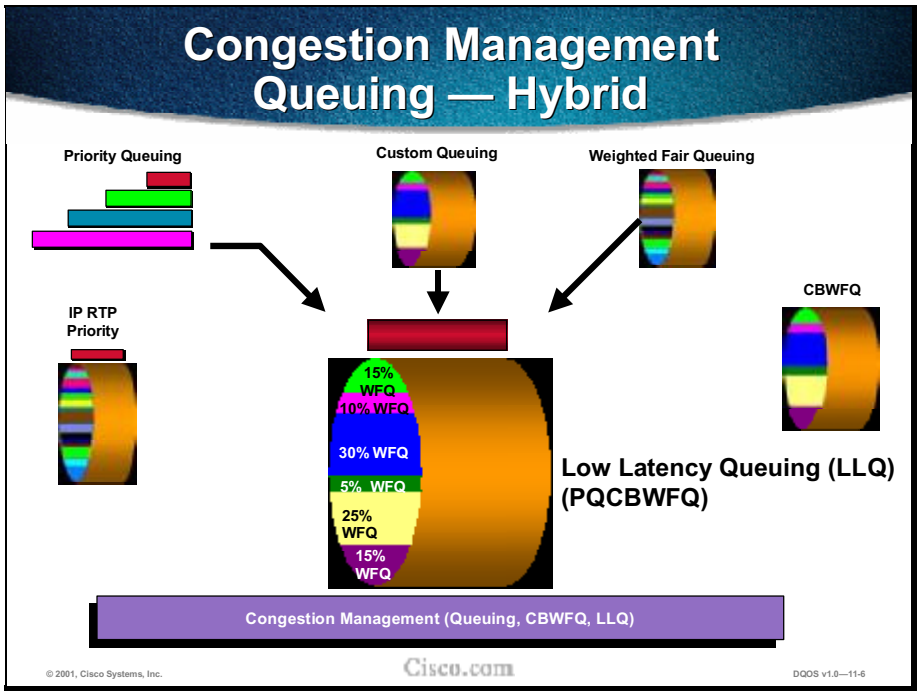
The CoS field is used for Layer 2 marking, while the ToS field is used for Layer 3 marking.

- IP Precedence and DiffServ are two key tools used in Layer 3 marking.

NBAR is used for Layer 4 to 7 marking.

Configuration of marking can be done via MQC, PBR, ACL/route maps, dial peers, or CAR.

Refer to Chapter 3, Classification and Marking, for details.

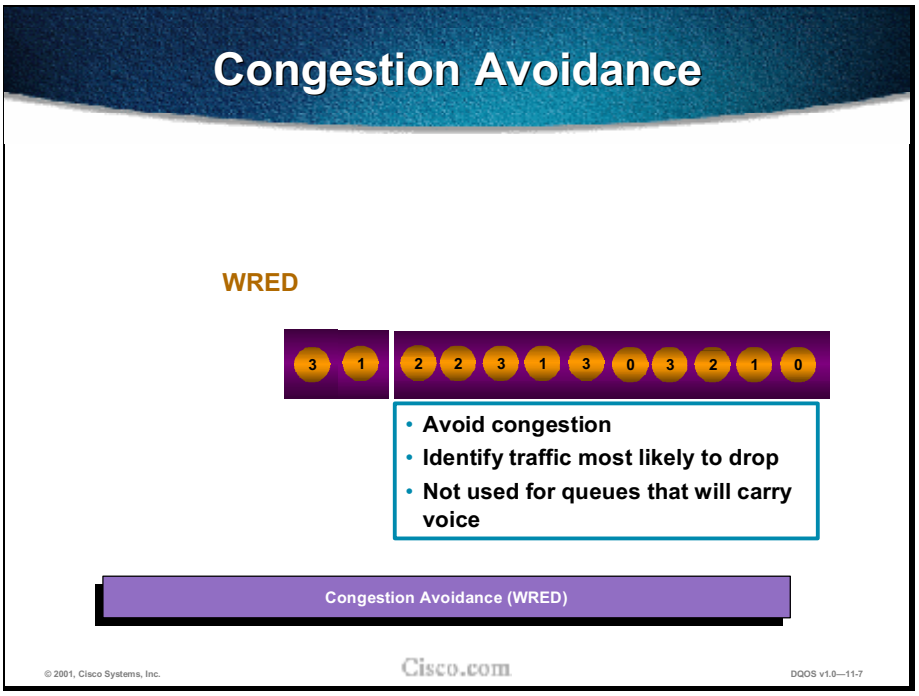


The following queuing techniques are available in IOS:

- PQ
- CQ
- WFQ/DWFQ
- CBWFQ
- LLQ (PQ/CBWFQ)

Proper queuing techniques must be chosen and applied for the right application.

Refer to Chapter 4, Congestion Management, for details.

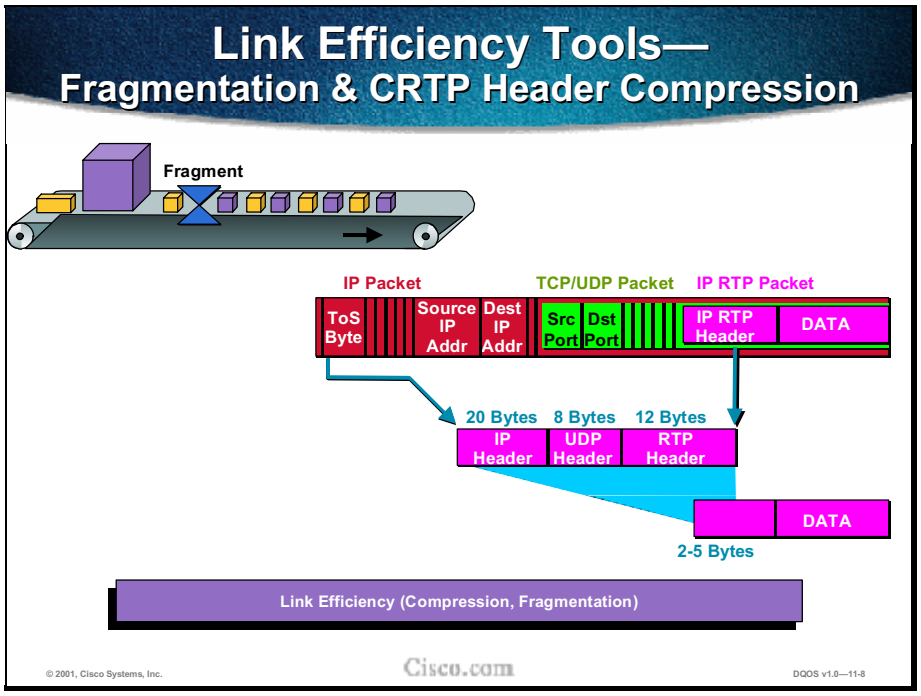


TCP's tail drop and global synchronization behavior negatively impact all traffic flows during congestion.

Several IOS congestion avoidance tools are available:

- RED
- WRED
- FRED

Refer to Chapter 5, Congestion Avoidance, for details.



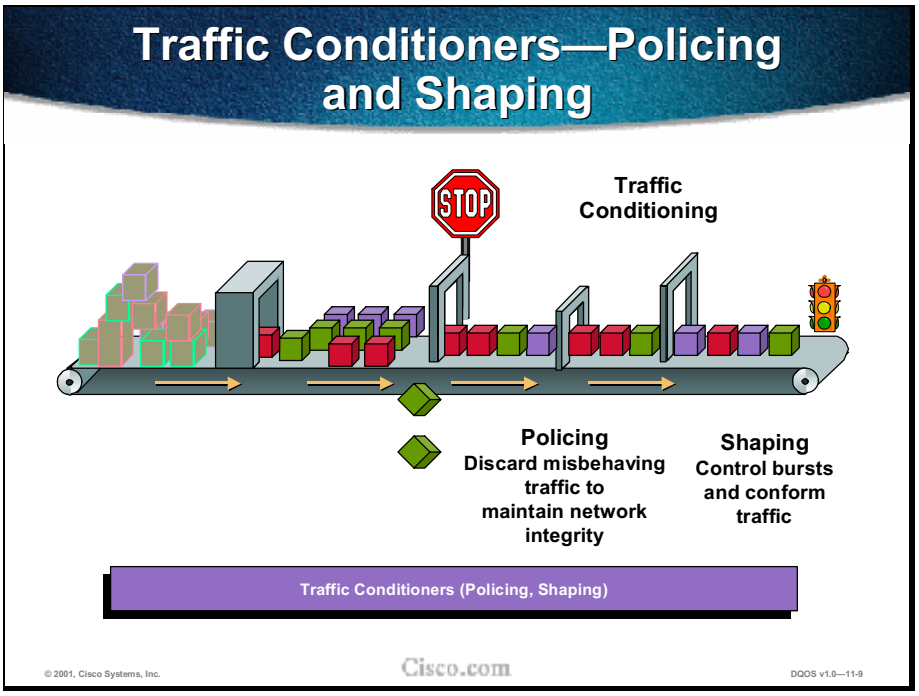
There are two categories of tools for link efficiency:

- Fragmentation and interleaving
  - LFI for MLP
  - FRF.12
  - FRF.11 Annex
  - Cisco Proprietary
- Header compression
  - CRTP

Application of each of the tools needs to be carefully implemented.

Configuration involves selection of key parameters and application of some global and interface commands.

Refer to Chapter 6, Link Efficiency Tools, for details.



Policing and shaping tools are used to regulate traffic to control congestion.

Policing simply rate-limits traffic and drops excess packets, while shaping deploys queues for buffering.

There are two options for policing:

1. Rate limiting using CAR
2. Class-based policing

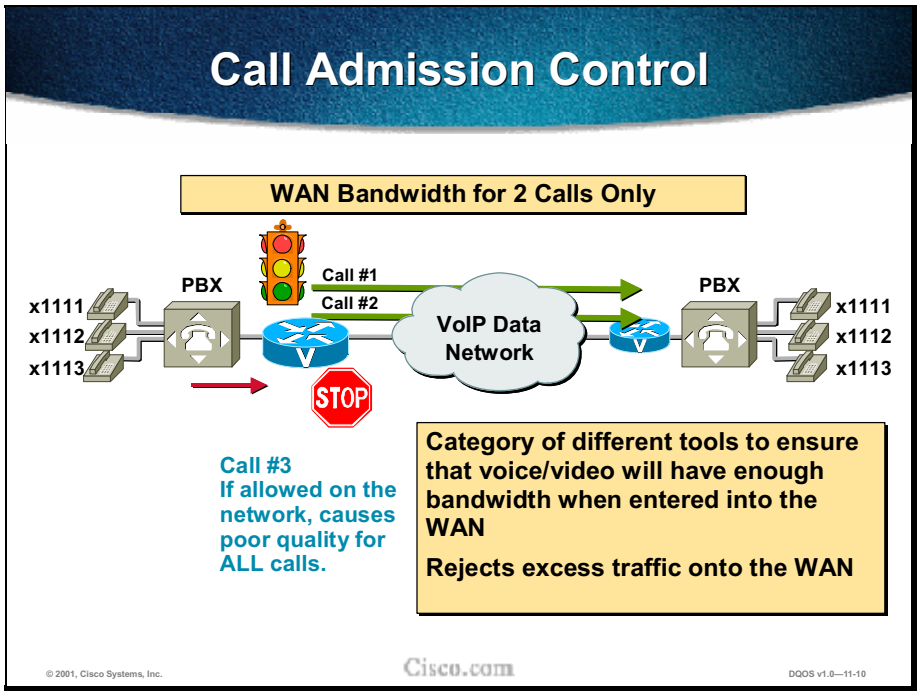
There are several options for traffic shaping:

- GTS
- DTS
- FRTS
- Class-based shaping
- Interface-based shaping

Selection of the appropriate tool must be done carefully.

Configuration involves the selection of various parameters and global and interface commands.

Refer to Chapter 7, Policing and Shaping, for details.



There are five local CAC methods:

1. Physical DS0 limitation
2. Max connections
3. Voice bandwidth for FR
4. Trunk conditioning
5. Local voice busyout (LVBO)

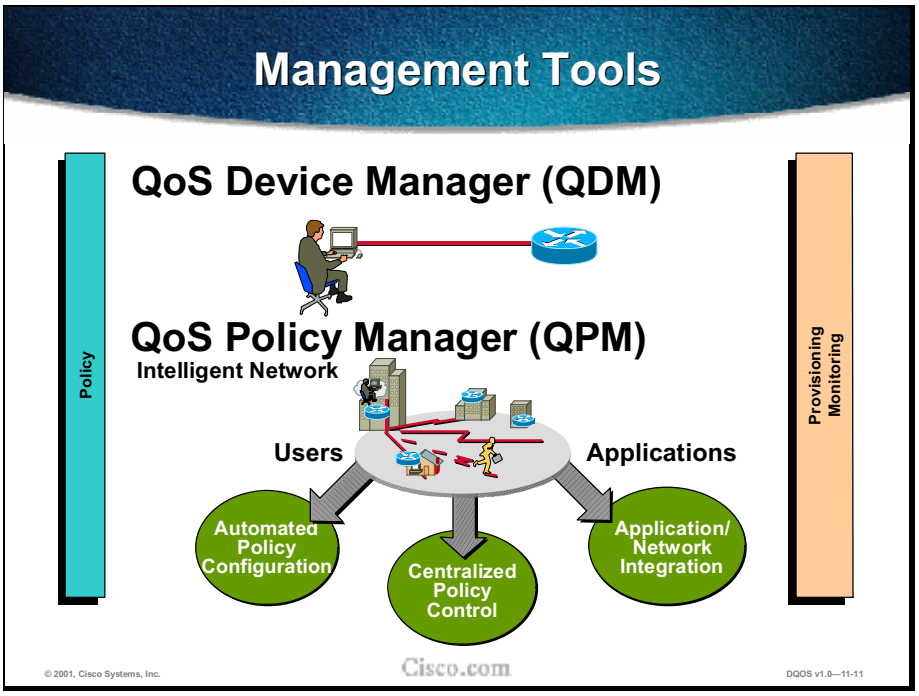
There are two measurement-based CAC methods:

1. AVBO (advanced voice busyout)
2. PSTN fallback

RSVP is the only CAC method that does bandwidth reservation.

The best method(s) of achieving call admission control to meet specific customer requirements must be carefully selected.

Refer to Chapter 8, Call Admission Control, for details.



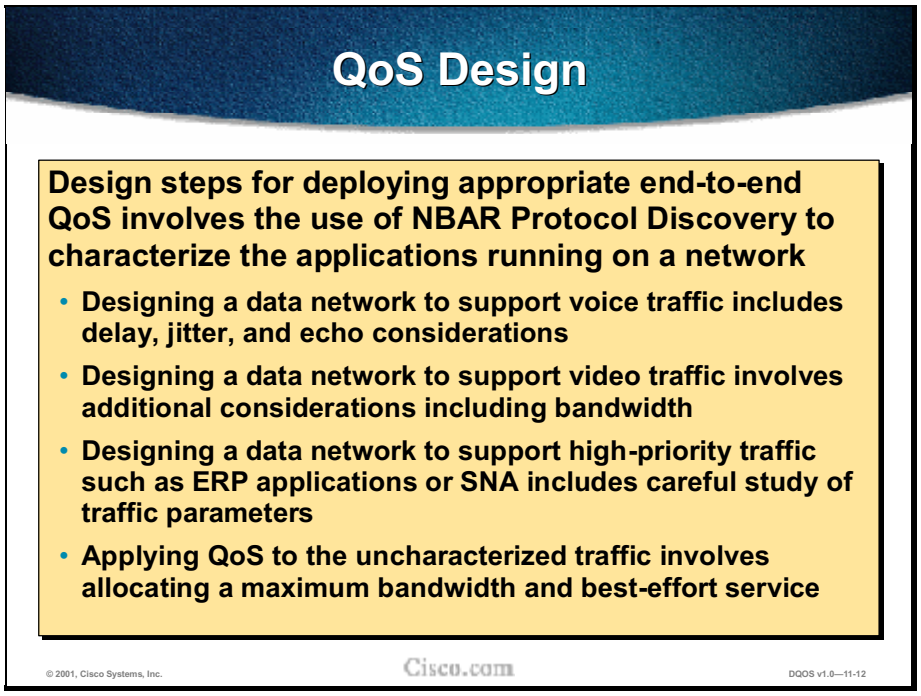
The QoS Device Manager (QDM) is used to monitor performance, establish baselines, and configure QoS policies.

The QoS Policy Manager (QPM) is used to configure advanced QoS policies, scale policy deployment, upload/verify/rollback policies, and deploy QoS policies by external time-based/event-based scripts.

The Cisco Service Assurance Agent (SA Agent) is used to measure key SLA metrics and monitor network performance between local and remote devices.

PM and SMS tools are used to monitor and troubleshoot network performance.

Refer to Chapter 9, Management Tools, for details.



Refer to Chapter 10, QoS Design, for details.

Summary

Cisco IOS 12.1(5)T offers sophisticated QoS tools that provide intelligent and predictable network administration on converged networks.

A

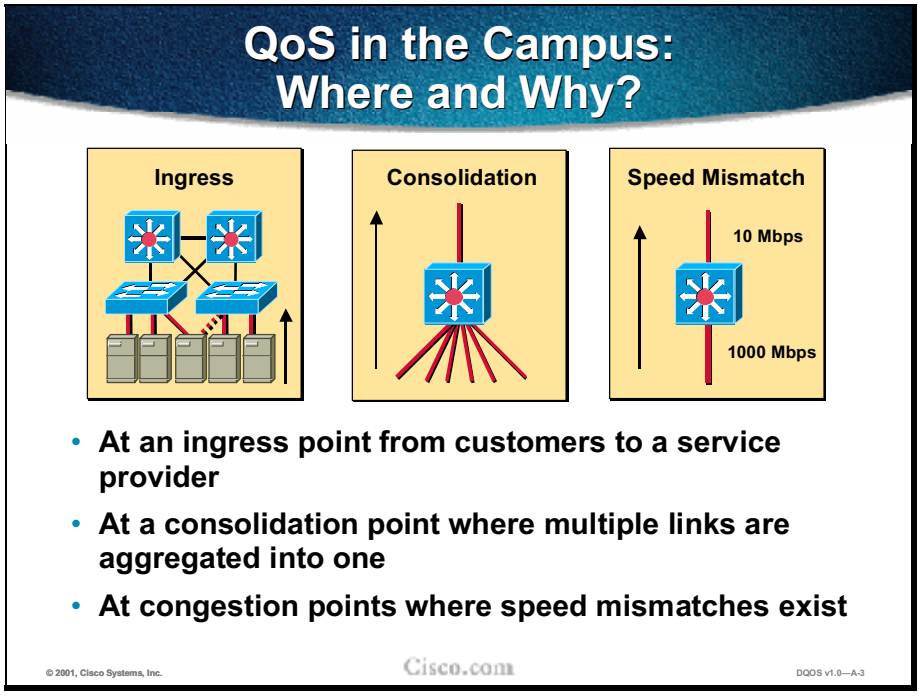
QoS on the Catalyst Switches

Overview

This chapter explains the quality of service (QoS) features available on the Catalyst line of switches. The various features are explained in detail, using the Catalyst 6XXX platform as the example. Then features available on all the other platforms are detailed.

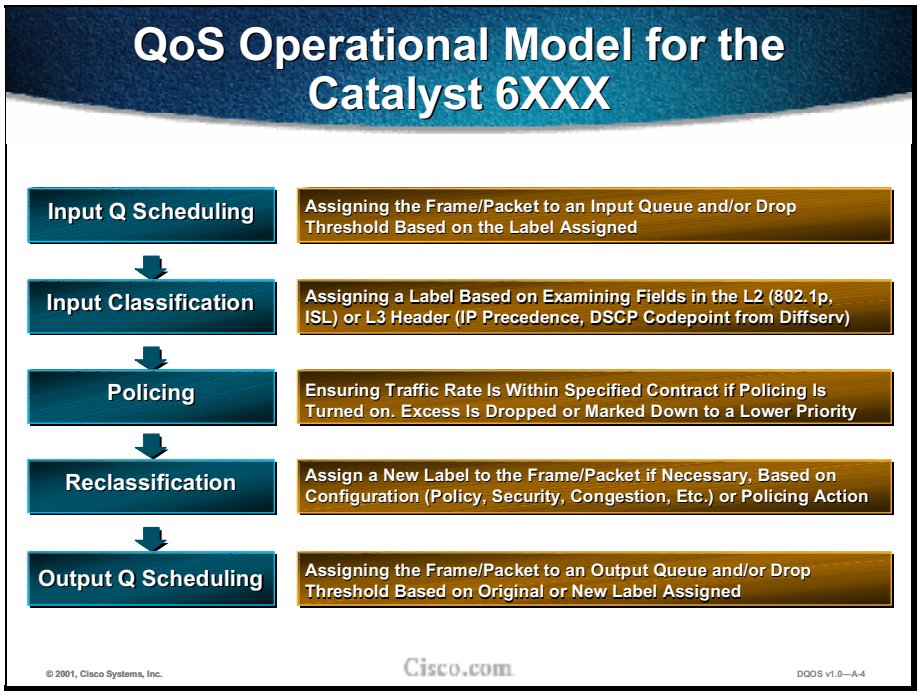
Objectives

- Explain the QoS features available on the Cisco Catalyst 6XXX
- Compare and contrast the QoS features available specifically on the Catalyst 2900XL/3500XL, 4XXX, 5XXX, and 6XXX series switches



QoS is a method that attempts to ensure that network requirements of different applications are met by prioritizing traffic according to its relative importance.

Though most often thought of in the WAN, where congestion is most likely to occur, QoS features can also provide major benefits in the campus.



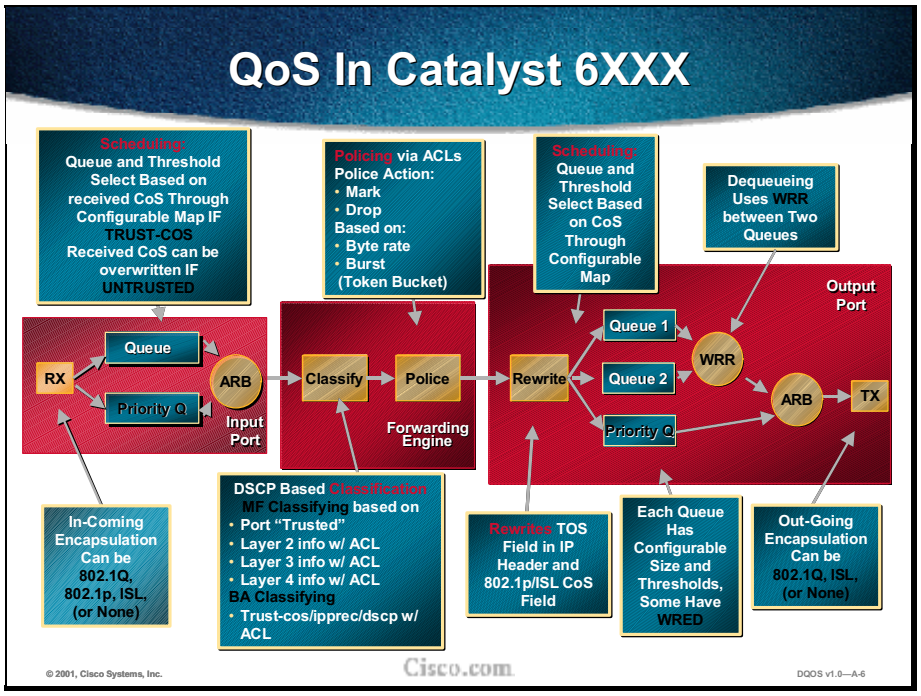
The QoS operational model for the Catalyst 6XXX starts with **input scheduling**. When a switch receives a frame on this Ethernet port, it first assigns the frame or packet to an input queue or drop threshold based on the packet's class of service (CoS).

Input classification, which resides on the policy feature card (PFC), is assigning a label based on the fields in the Layer 2 or Layer 3 header.

If **policing** is enabled, it ensures that the traffic rate is within a specified limit. Traffic that exceeds that limit is dropped or marked with a lower priority, based on its original priority and the switch configuration.

Reclassification is assigning a new label to the frame packet, if necessary. This can be statically configured, or it can be done dynamically by the policing action.

Output queue scheduling is assigning the frame to an output queue or drop threshold based on its assigned label.

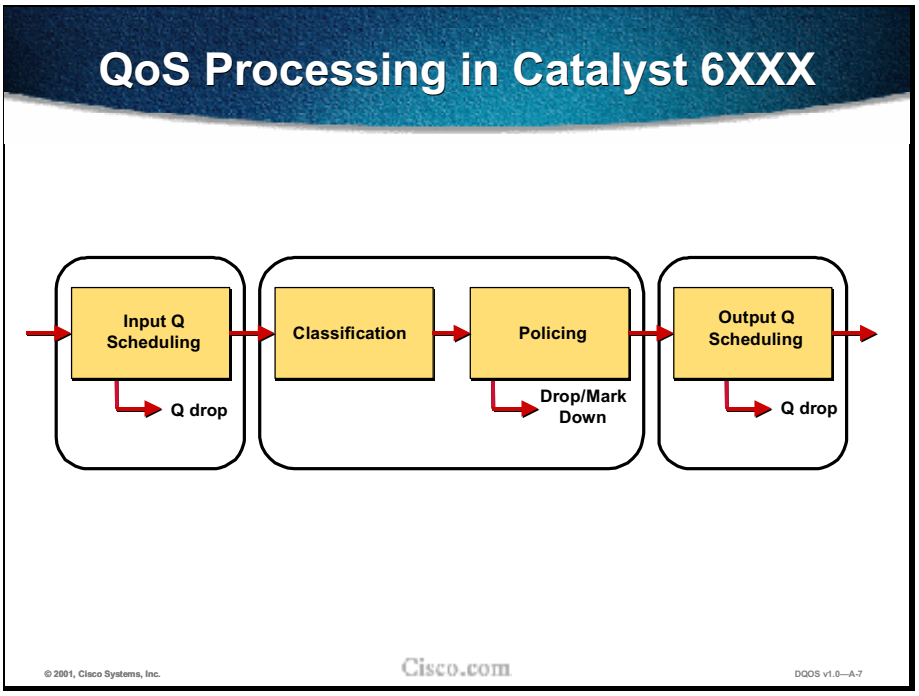


This slide provides an overview of frame flow through a Catalyst 6XXX switch. Each area is covered in greater detail on subsequent slides.

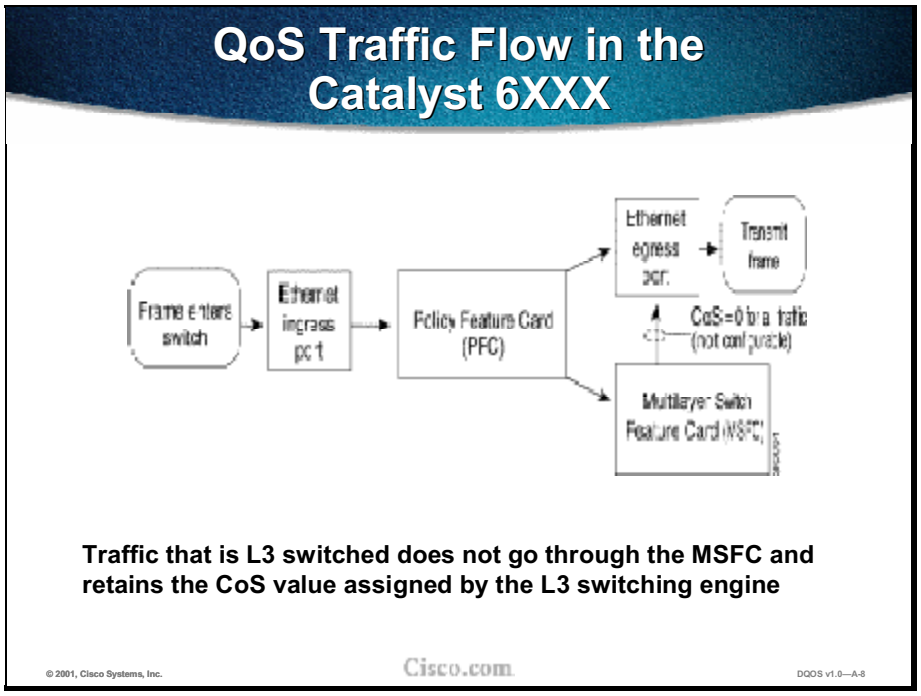
When a frame arrives with 802.1q, ISL or standard Ethernet framing on the input port, the switch first determines which input queue the frame should be in. On 10/100 modules, Trust CoS must be enabled for any special input queuing to occur. The inbound queue and thresholds are based on the frame's ISL or 802.1q priority if "trusted," or the port priority if "untrusted" or the frame is not marked. The input port then sends the frame to the forwarding engine.

The forwarding engine performs additional classification as needed based on trust, any Layer 2, 3, or 4 information with ACLs, and/or IP Precedence or DSCP values. If policing is associated with an ACL, that action takes place next. Policing can mark a frame with a lower priority, or drop the frame entirely, based on the token bucket algorithm. The frame is then passed to the outbound port.

The output port may rewrite the Layer 2 and/or Layer 3 priority, then it passes it to the appropriate outbound queue, determined by the port's configuration. Depending on the card, there can be two queues that are serviced in a weighted round robin (WRR) fashion and another priority queue whose traffic is dispatched immediately. Within each queue there are configurable weighted random early detection (WRED) thresholds for congestion avoidance.



This diagram illustrates what is responsible for where the input queuing happens, then the classification and policing. Layer 2 does possess some QoS capabilities, but the main portion of the classification and policing is done on the policy feature card (PFC).

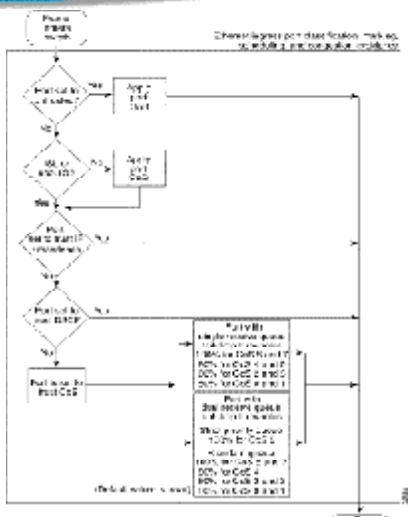


The frame enters the switch at the ingress port. It is sent to the PFC and is then either sent to the Ethernet egress port or to the multilayer switch feature card (MSFC), if it needs to be routed. If the packet is forwarded to the MSFC, the MSFC reencapsulates it with a new Ethernet frame and applies a CoS of 0. That is not configurable.

Ingress Marking & Queuing

TO SEE QUEUE STRUCTURE:
sh port capabilities <mod/port>

```
quebec (enable) sh port capa 5/1
Model                    WS-X6248-RJ-45
Port                     5/1
Type                     10/100BaseTX
Speed                    auto,10,100
Duplex                   half,full
Trunk encap type         802.1Q,ISL
Trunk mode               on,off,desirable,auto,nonegotiate
Channel                  yes
Broadcast suppression    percentage(0-100)
Flow control             receive-(off,on),send-(off)
Security                 yes
Membership               static,dynamic
Fast start               yes
QOS scheduling           rx-(1q4t),tx-(2q2t)
CoS rewrite              yes
ToS rewrite              DSCP
UDLD                     yes
Inline power             no
AuxiliaryVlan
1..1000,1025..4094,untagged,dot1p,none
SPAN                     source,destination
COPS port group          5/1-48
```

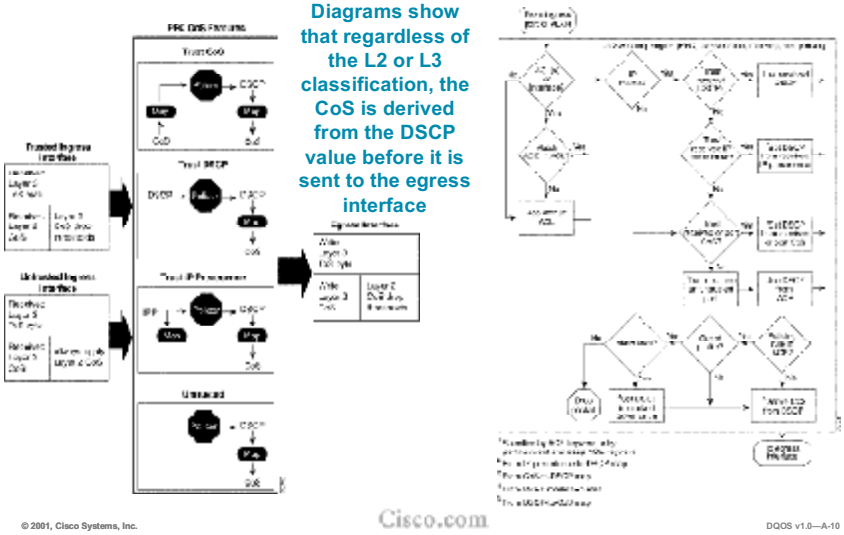


This diagram illustrates what happens to a frame as it goes through its classification or queue scheduling classification, policing, and output queue.

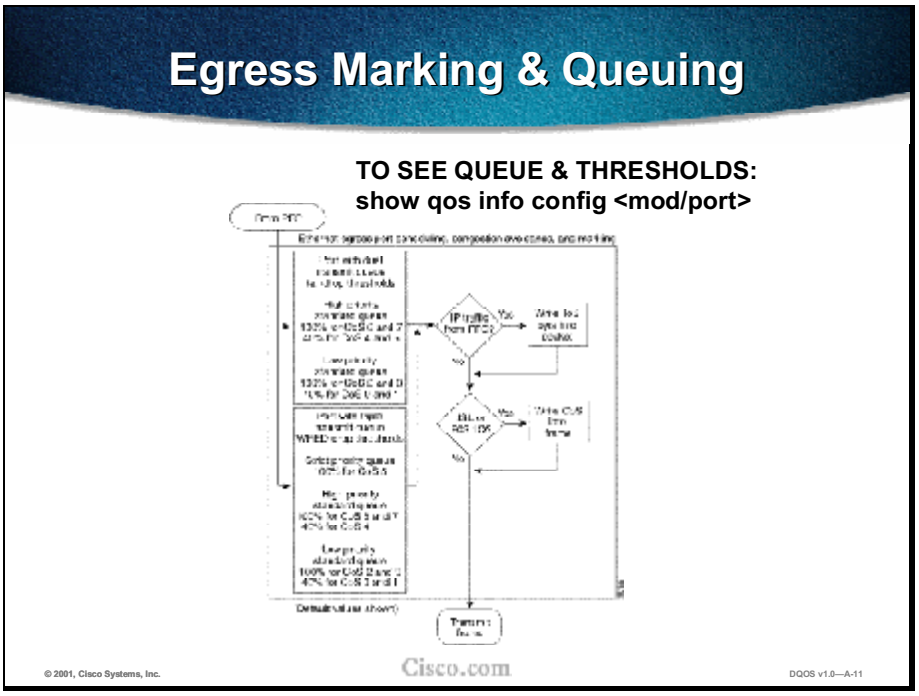
In this case, the RX queue is set to 1q4t (one queue, four thresholds) and the TX queue is set to 2q2t (two queues, two thresholds). As the frame enters the switch, one of two things happens. If the port is untrusted, the switch applies a CoS (0 by default). If it is trusted, the switch still applies the configured CoS if the frame is not 802.1q or ISL.

QoS in the L3 Engine (PFC)

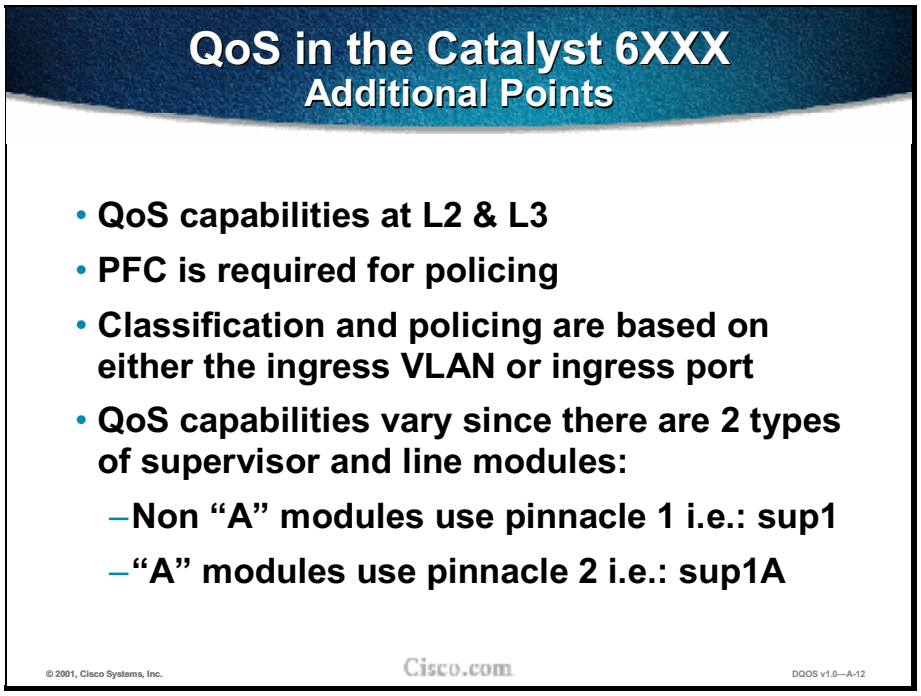
Diagrams show that regardless of the L2 or L3 classification, the CoS is derived from the DSCP value before it is sent to the egress interface



These diagrams show that regardless of the L2 or L3 classification, the CoS is derived from the DSCP value before it is sent to the egress interface.

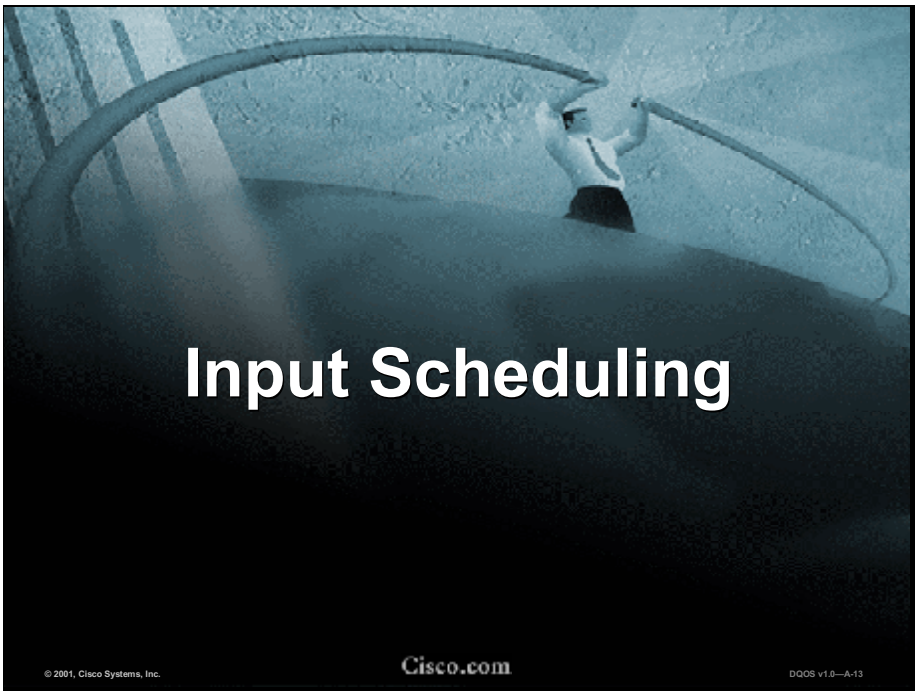


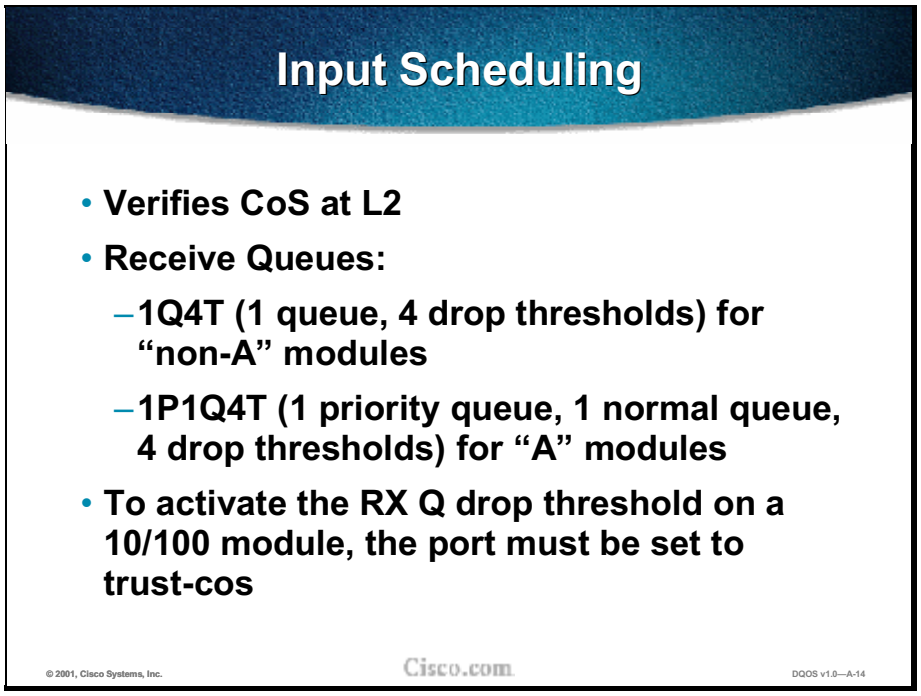
A **show qos info config** command displays the QoS parameters for a given port.



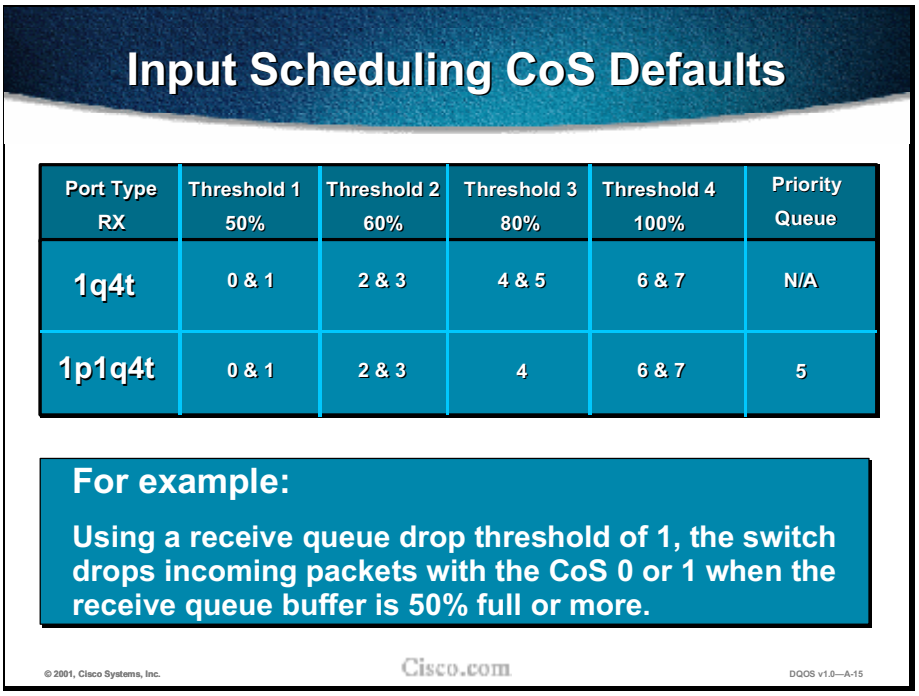
Features are as follows:

- There are QoS capabilities at L2 and L3.
- PFC is required for policing.
- Classification and policing are based on either the ingress VLAN or ingress port.
- QoS capabilities vary since there are two types of supervisor and line modules:
  - “Non-A” modules use pinnacle 1 i.e.: sup1
  - “A” modules use pinnacle 2 i.e.: sup1A

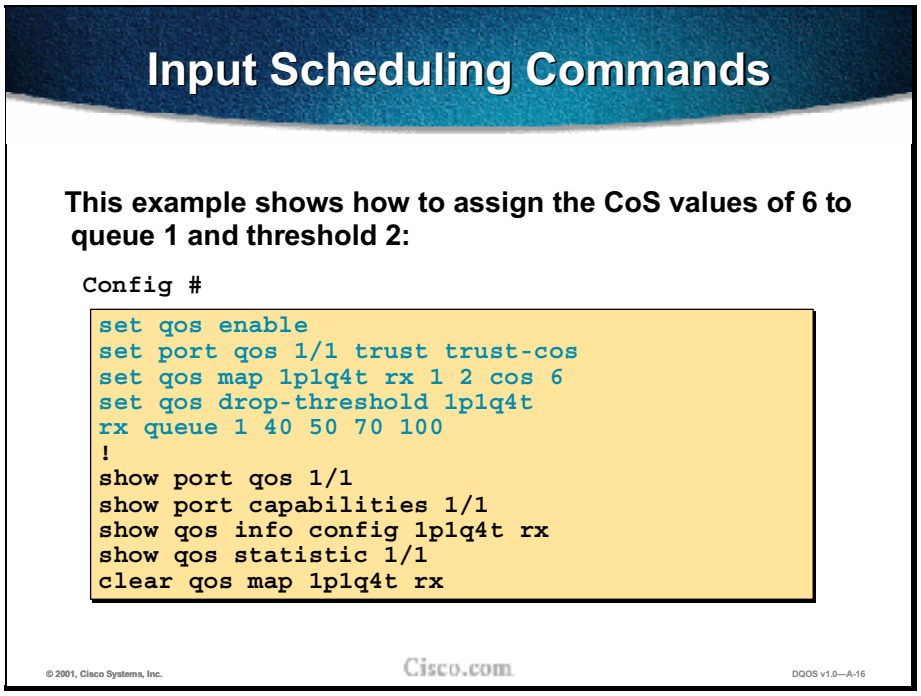




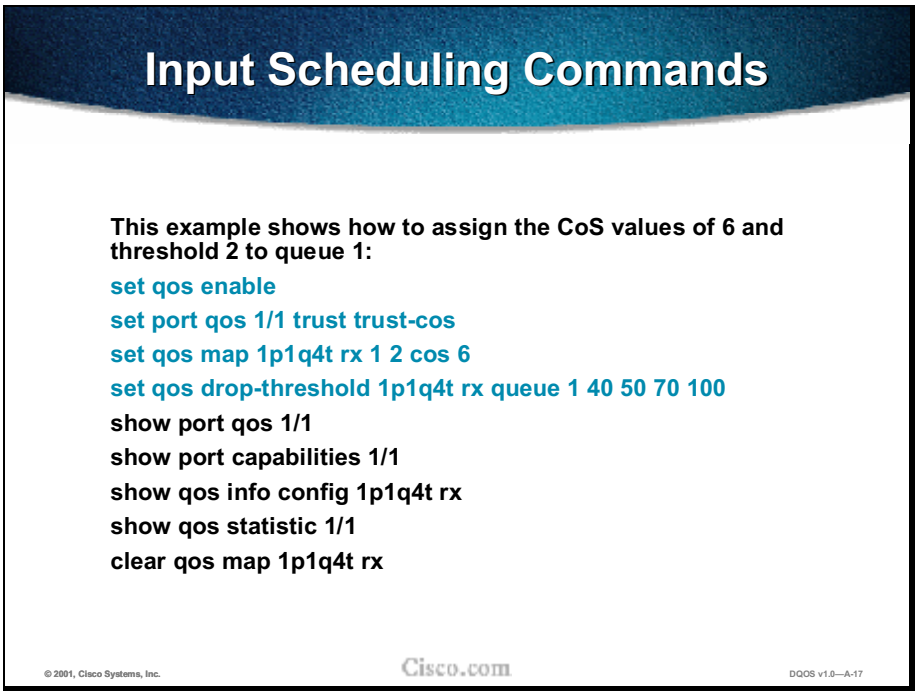
As with all the switches, the 6XXX can copy an 802.1p or ISL CoS field or a port priority CoS into the internal header, as with other switches, this internal header may get copied into an 802.1p or ISL value on the egress port. You can set a port to be trusted or untrusted (trusted is supported only on Gig E); if it is trusted, it is forwarded directly to the buffer and then to the switching fabric. If the port is set to trust-cos, which activates the Rx tail-drop thresholds, the packet is placed into the appropriate queue threshold. If it is untrusted or not marked with an ISL or 802.1Q tag, the traffic is forwarded to the switching fabric with the default CoS value as configured at the port. The default is 0.



Here we see the default values for ingress scheduling.

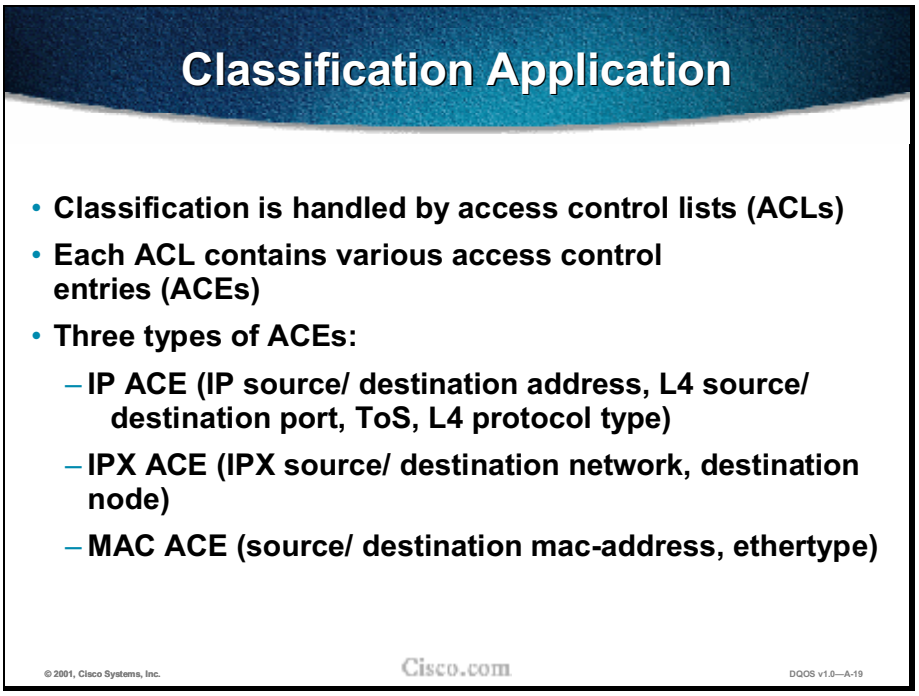


This configuration example shows how to modify the default ingress values.



In this example, queue 1 is set to 40, 2 is set to 50, and so forth. The **show** commands reveal stats for a given port, that is, is it in trust CoS or untrusted? **clear qos map** erases any previous configurations associated with that particular queue and goes back to the defaults.


Classification
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—A-18

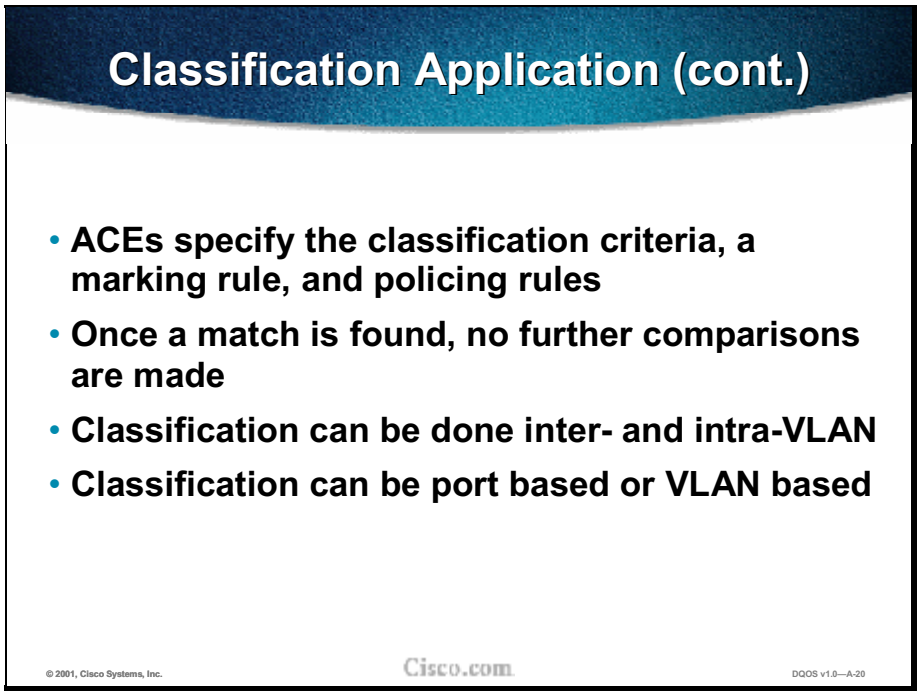


ACEs specify the classification criteria, a marking rule, and policing rules.

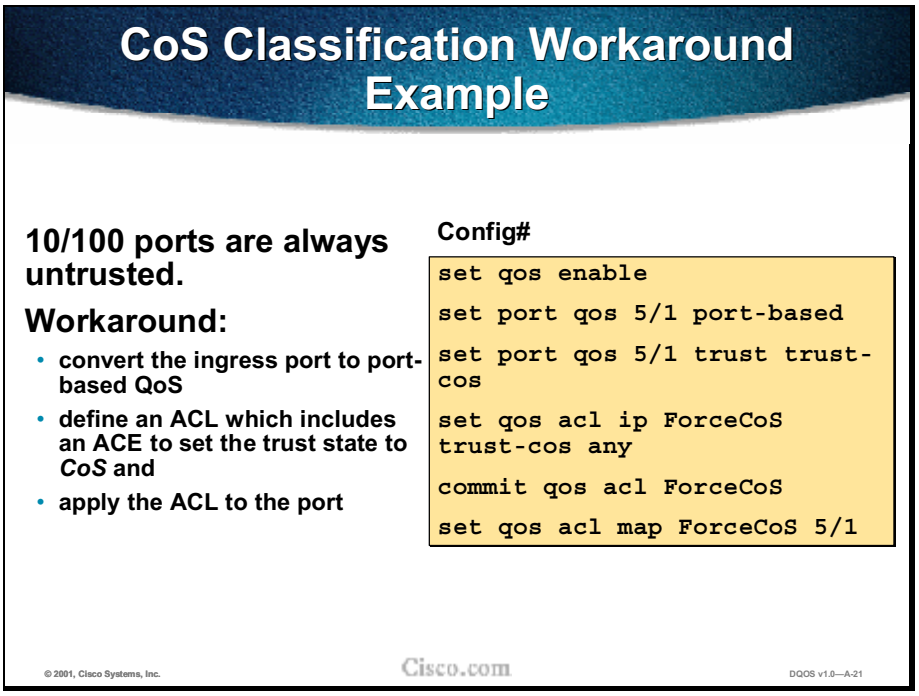
Once a match is found, no further comparisons are made.

Classification can be port based or VLAN based.

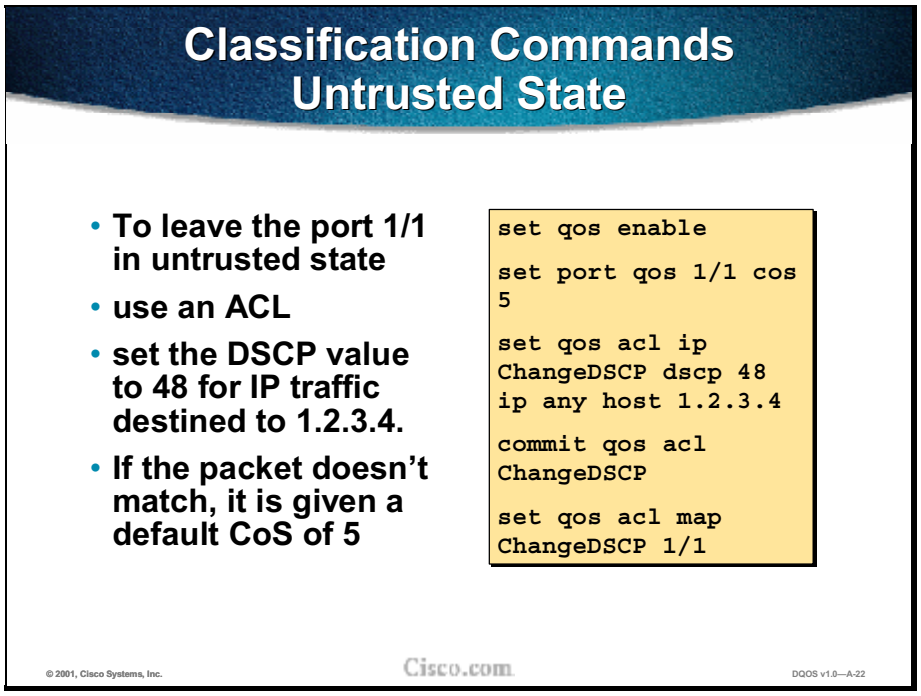
Classification can be done inter- and intra-VLAN.



ACEs specify the classification criteria and marking and policing rules. Once a match is found, it doesn't do any further comparisons. Classifications can be done inter- or intra-VLAN, or classification can be done on a port or VLAN basis.



By default, 10/100 ports always trust. A workaround is to convert the ingress port to port-based QoS, define an ACL, which includes an ACE to set the trust state to *CoS* and apply the ACL to the port.



This shows another classification example, showing how to leave a port in untrusted state and use an ACL to set the DSCP value to 48 for IP traffic destined to 1.2.3.4. If the packet doesn't match, it is given a default CoS of 5.

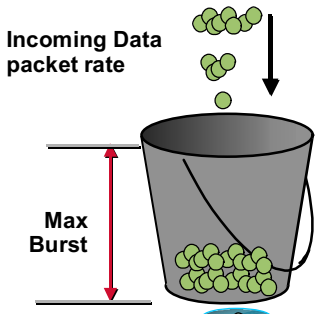
Policing on the Catalyst 6XXX

- Policing uses a token-bucket scheme
- If the packet is out-of-profile, choose drop or mark down the DSCP
- Two policers are applied to each packet:
  - Individual microflow policer: Supports up to 63 policing rules.
  - Aggregate-flow policer: Supports up to 1023.

Policing is the ability to limit bandwidth consumed by a flow of traffic on a given link or port. By using policing capabilities, service providers can regulate bandwidth consumption within a network. Without policing, customers are free to consume the maximum amount of bandwidth that is normally a fixed medium rate of either 10 Mbps, 100 Mbps, or 1 Gbps. This allows a greater degree of control in how much bandwidth is assigned to a particular port. The policer that is applied to each packet is either a microflow or an aggregate flow. Microflows are for individual, specific flows. Aggregate flows combine a range of microflows.

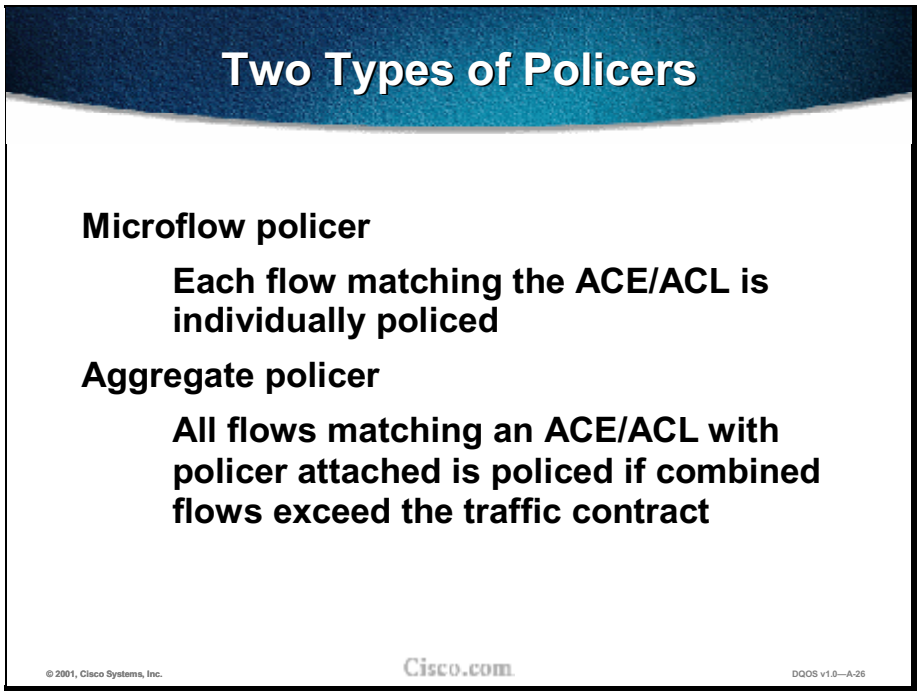
Token Bucket for Policing

- Start with a bucket without tokens
- Tokens can be added at a bursty rate
- Tokens are leaked at a specified constant rate
- If the bucket overflows, packet gets policed



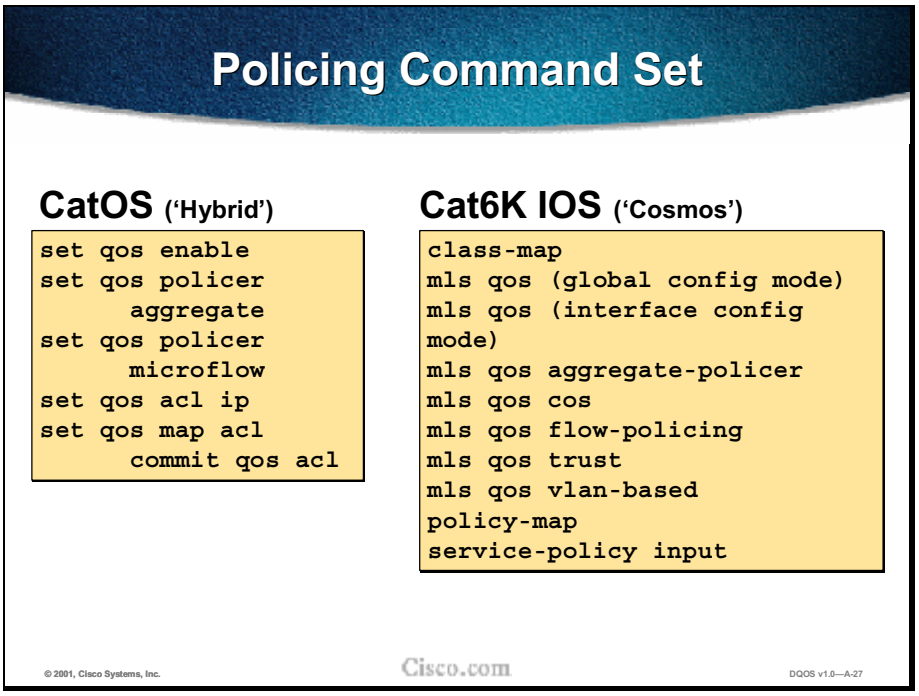
Tokens are leaking out of the bucket at an avg rate specified by the user. This happens every 'leaking interval' (0.25 msec in the Catalyst 6000.)

Policing uses a token bucket algorithm. If the packet is out of profile, one can choose to drop or mark down the differentiated services code point (DSCP) value.

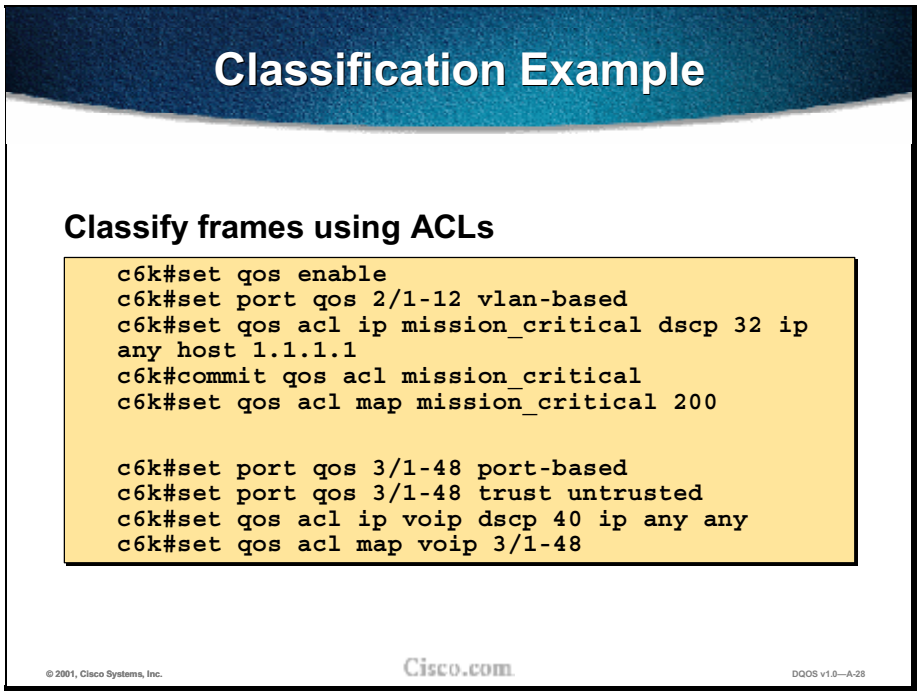


The Catalyst 6XXX provides two types of policing:

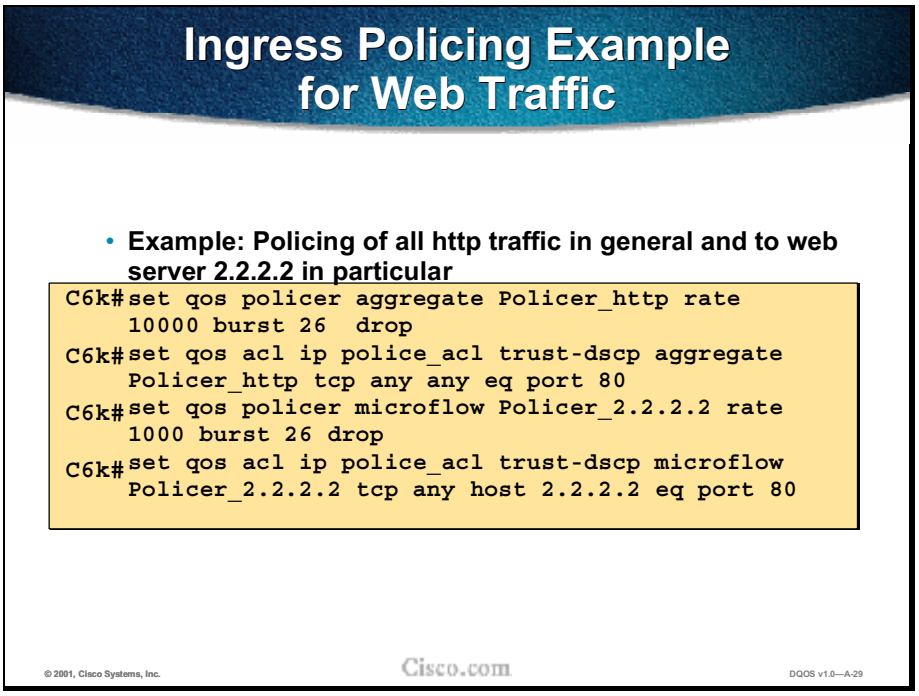
- When adding a microflow policer to an ACE/ACL, each flow matching the ACE/ACL is individually policed.
- When adding an aggregate policer to an ACE/ACL, all flows matching an ACE/ACL with a policer attached is policed if the combined flows exceed the traffic contract.



These are the commands used for policing.



Here we are classifying frames using ACLs. All traffic from VLAN 200 to 1.1.1.1 gets a CoS of 4, which has an internal DSCP value of 32. IP phones are on Module 3 and get a DSCP of 40, which is a default CoS of 5. Therefore, IP phones will be mapped to the high-priority queue. Looking at the first example, we set the ports 2/1-12 to VLAN-based QoS. Then we set the access list "mission_critical" DSCP value of 32 from any IP host-to-host 1.1.1. Then that rule is applied to VLAN 200. This is the classification step.



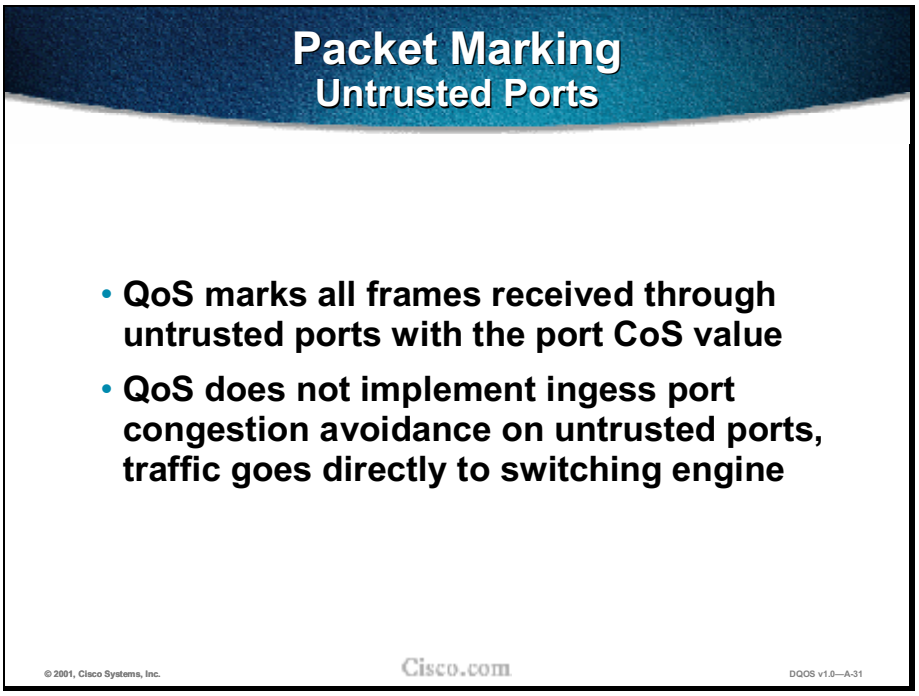
This is a typical application. All HTTP traffic is being aggregate policed, and web traffic 2.2.2.2 is being microflow policed.

The first line defines an aggregate policer named **Policer_http** and allows 10 Mbps (value in the command is in kbps) in 26-kbps bursts. The second line is an ACL that defines the traffic to be assigned to the aggregate flow named Policer_http.

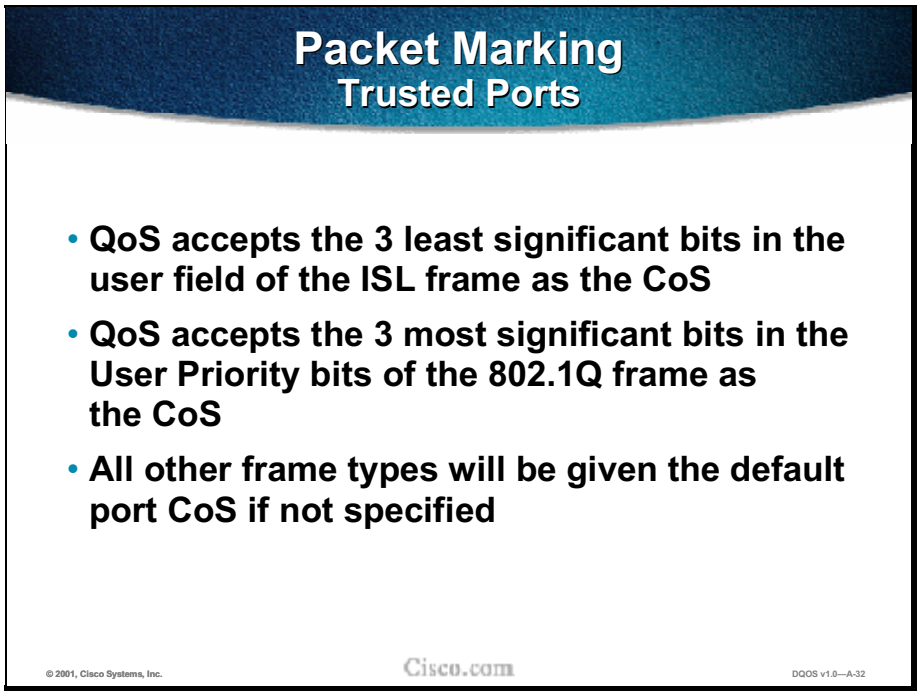
The third line defines a microflow policer named **Policer_2.2.2.2** and allows 1 Mbps in 26-kbps bursts. The fourth line is an ACL that defines the traffic to be assigned to that microflow.

The **trust-dscp** parameter in both ACLs tells the switch not to modify the CoS on inbound traffic.

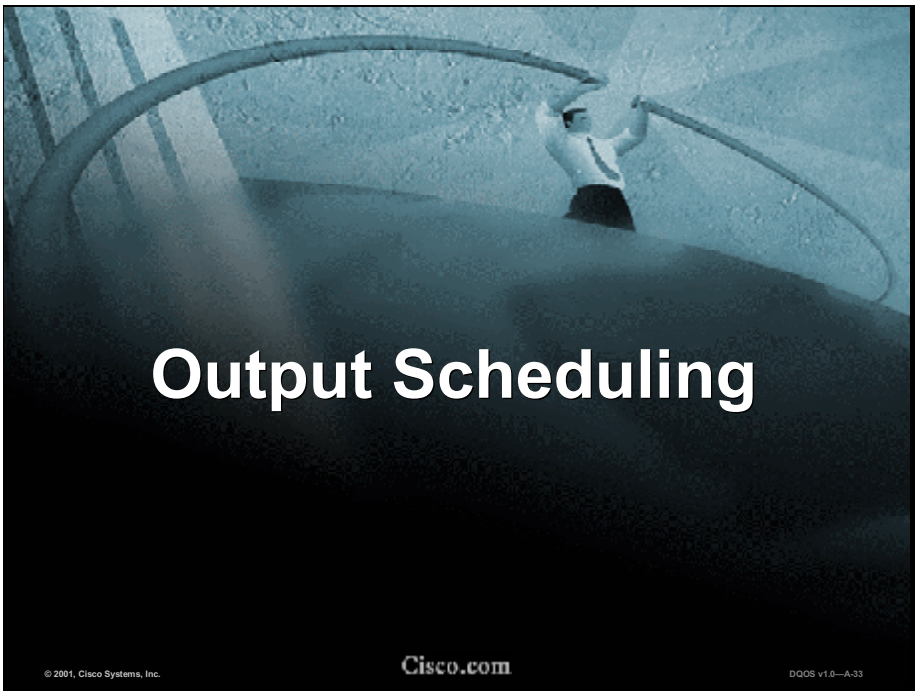

Reclassification
Marking
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—A-30

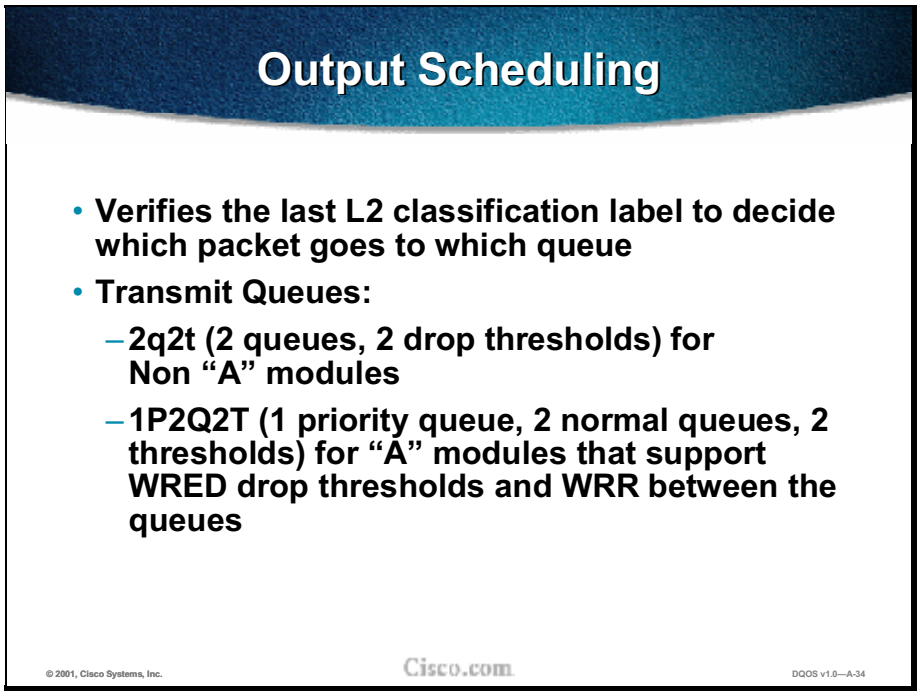


On untrusted ports, the switch marks all frames received with the port class-of-service value. QoS does not implement ingress port congestion avoidance on untrusted ports; traffic goes directly to the switching engine.



On trusted ports the switch accepts the three least significant bits of the ISL user field or the three most significant bits of the 802.1Q priority as the CoS. All other frame types are given the port priority 0 unless otherwise specified.





Output scheduling verifies the Layer 2 classification label to decide which packet goes into which queue. Transmit queues are as follows: two queues, two thresholds for "non-A" modules and on the 10/100 cards, and one priority queue, two WRR queues, and two thresholds for "A" modules. (See slide below.)

Output Scheduling CoS Defaults

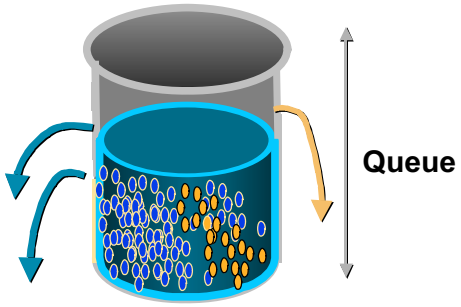
Port Type RX	Drop Threshold	Low Priority Queue 1	High Priority Queue2	Strict Priority Queue 3
2q2t	Low Drop Threshold 2 100%	2 & 3	6 & 7	N/A
	High Drop Threshold 1 80%	0 & 1	4 & 5	N/A
1p2q2t	Low Drop Threshold 2 100%	2 & 3	6 & 7	N/A
	High Drop Threshold 1 80%	0 & 1	4	5

This table shows the default values for the egress queues.

Weighted RED

WRED addresses:

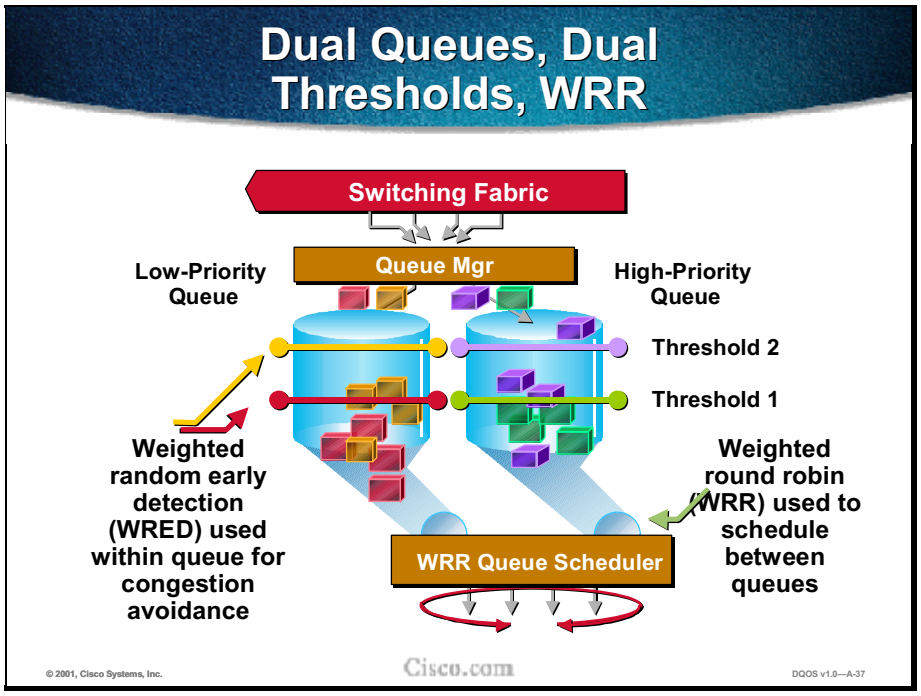
- **In the event that packets need to be dropped, what class of packets should be dropped**



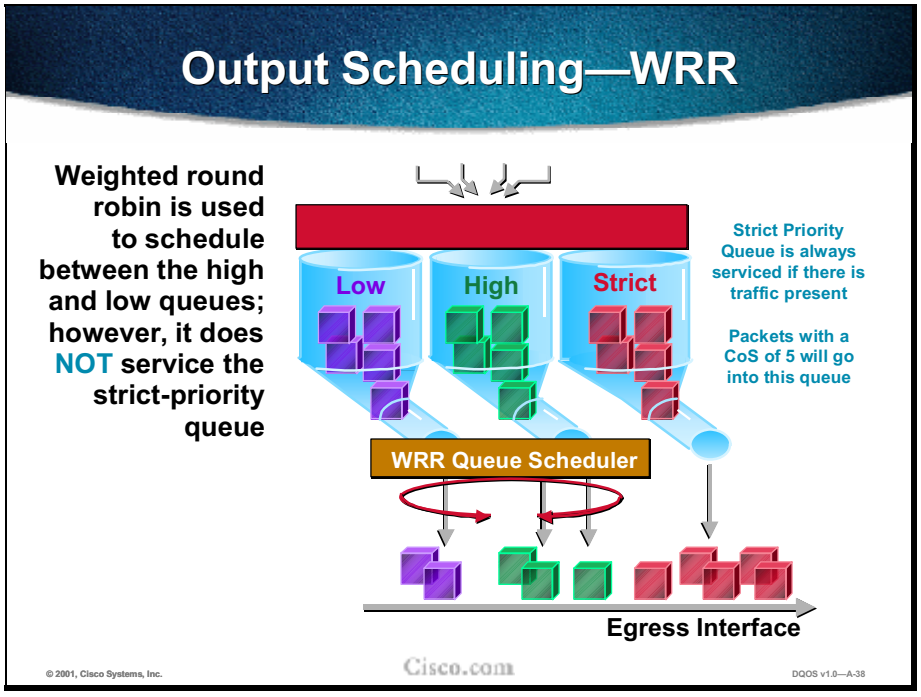
Packets Classified as Blue Start Dropping at a 50% Queue Depth. Drop Rate Is Increased as Queue Depth Is Increased

Packets Classified as Gold Are Dropped at 90% Queue Depth

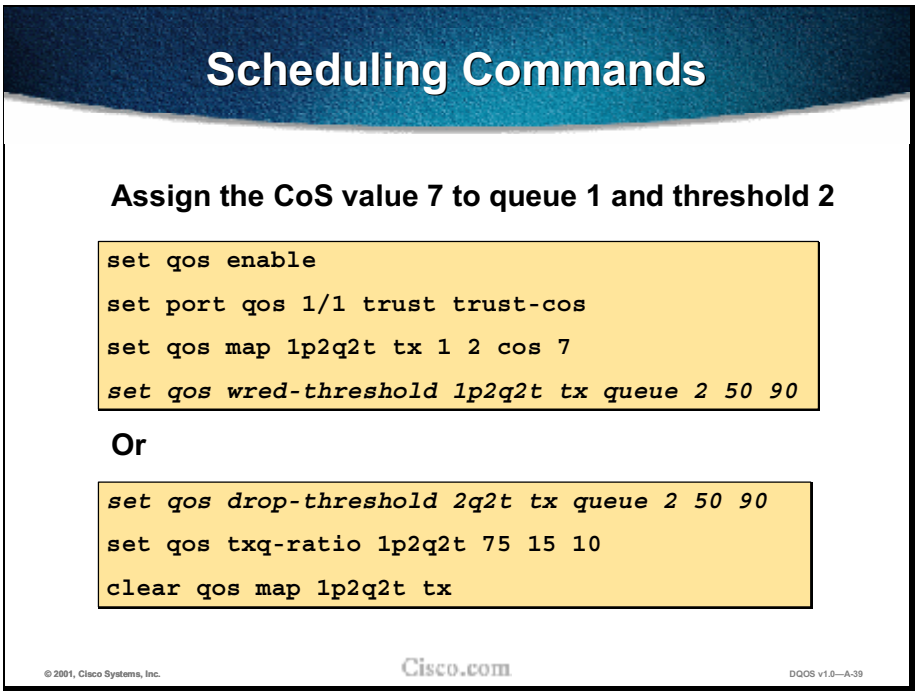
This figure reviews the concept of weighted random early detection (WRED).



This figure illustrates output scheduling and WRED thresholds for the “non-A“ cards.

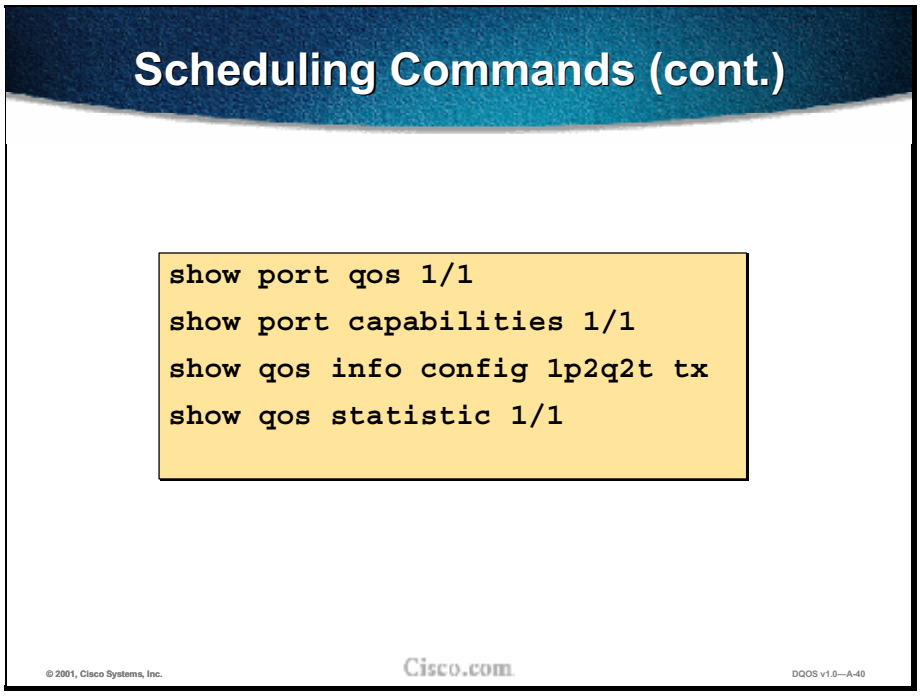


This figure illustrates output scheduling for the “A“ cards.

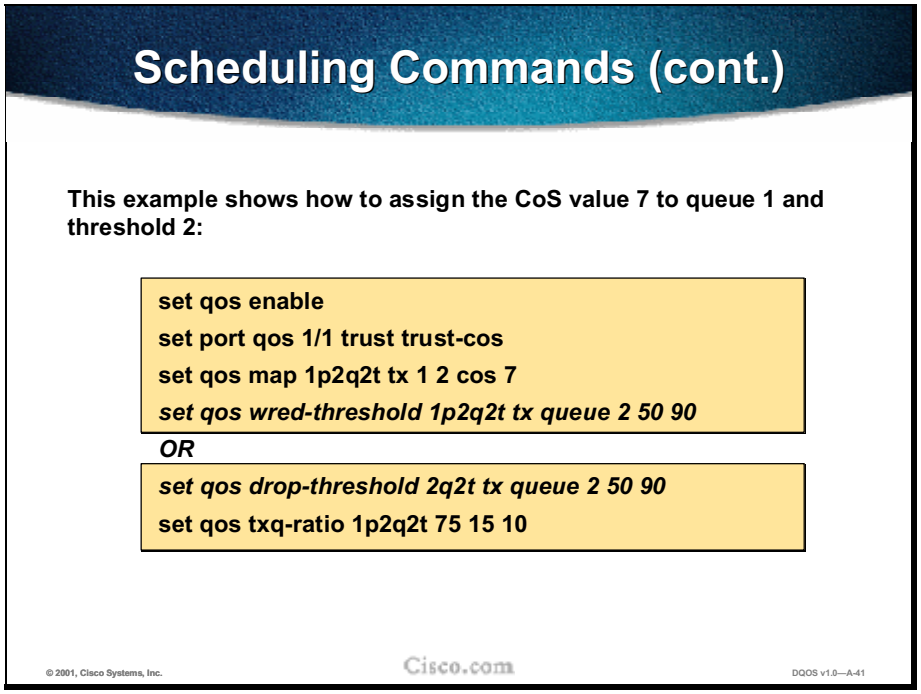


Output scheduling consists of configuring:

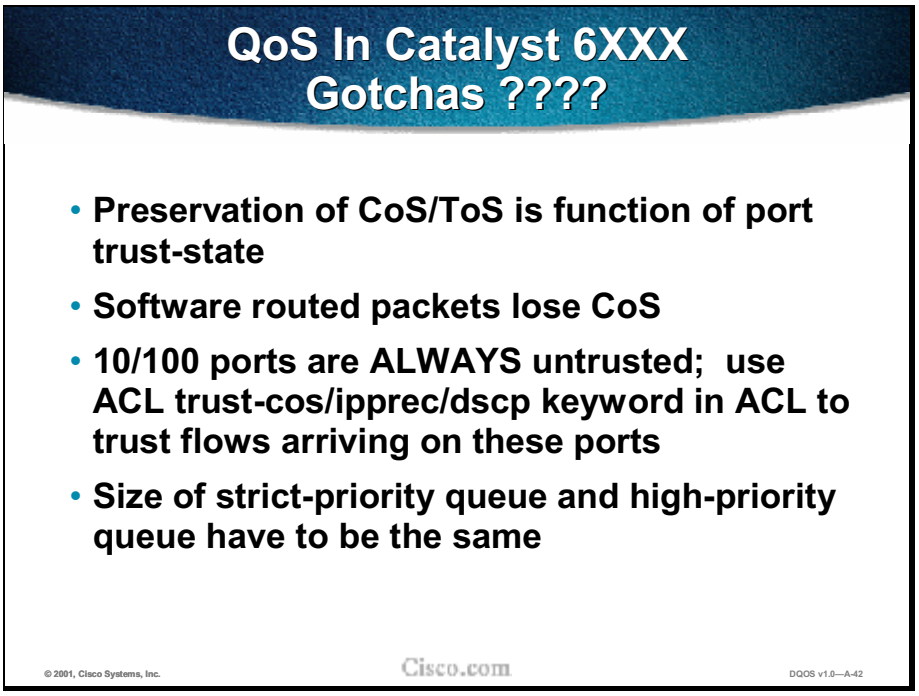
- Queues or thresholds to which packets are assigned on the basis of the CoS
- Buffer memory for each individual queue
- Weight for the WRR between the queues



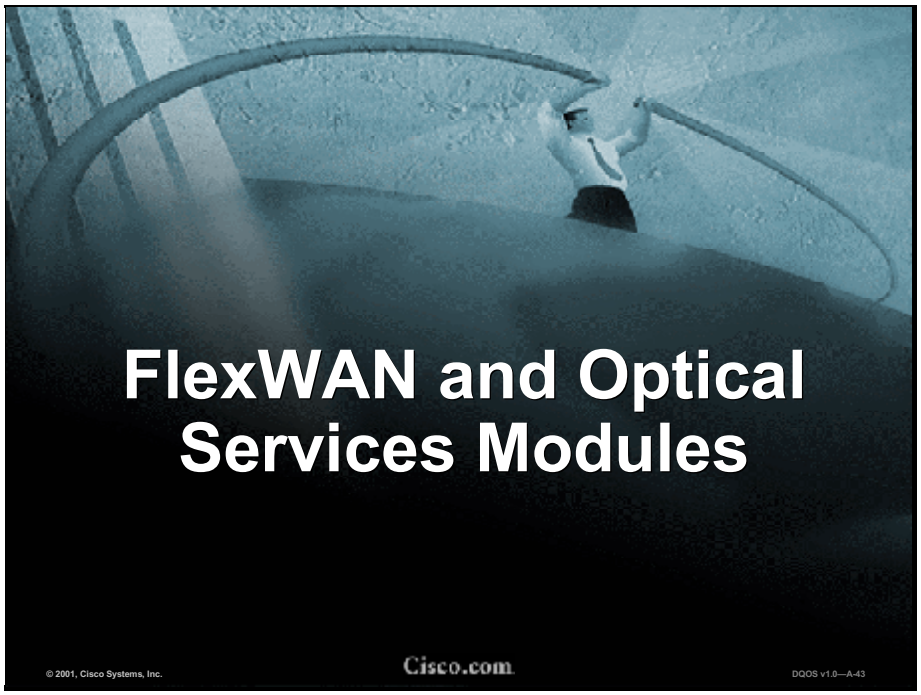
These commands are used to confirm the configuration.

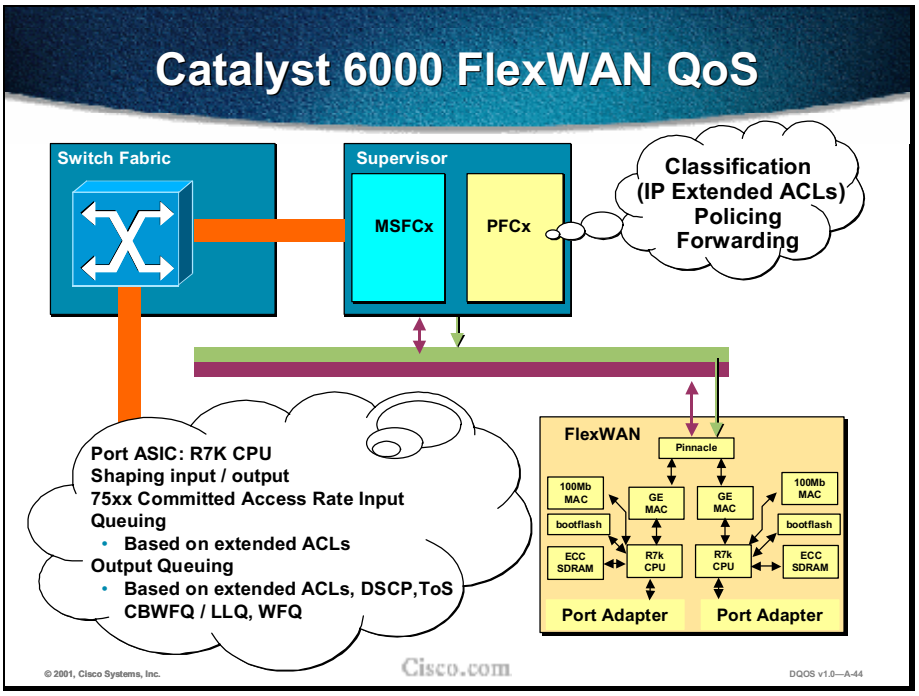


This example shows how to assign the CoS value 7 to queue 1 and threshold 2.

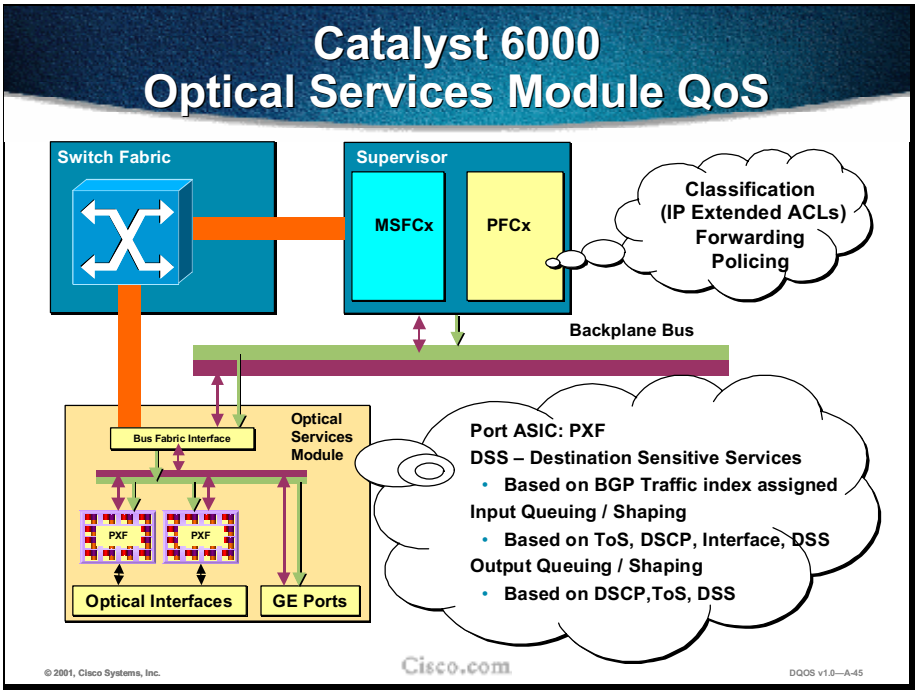


Potential challenges include the following:Preservation of CoS and ToS is a function of the port trust state. Packets routed through an MSFC lose the class of service (a product of reencapsulation). The 10/100 ports are always untrusted by default and use ACL trust CoS to enable trust. The sizes of the strict-priority queue and the high-priority queue have to be the same.

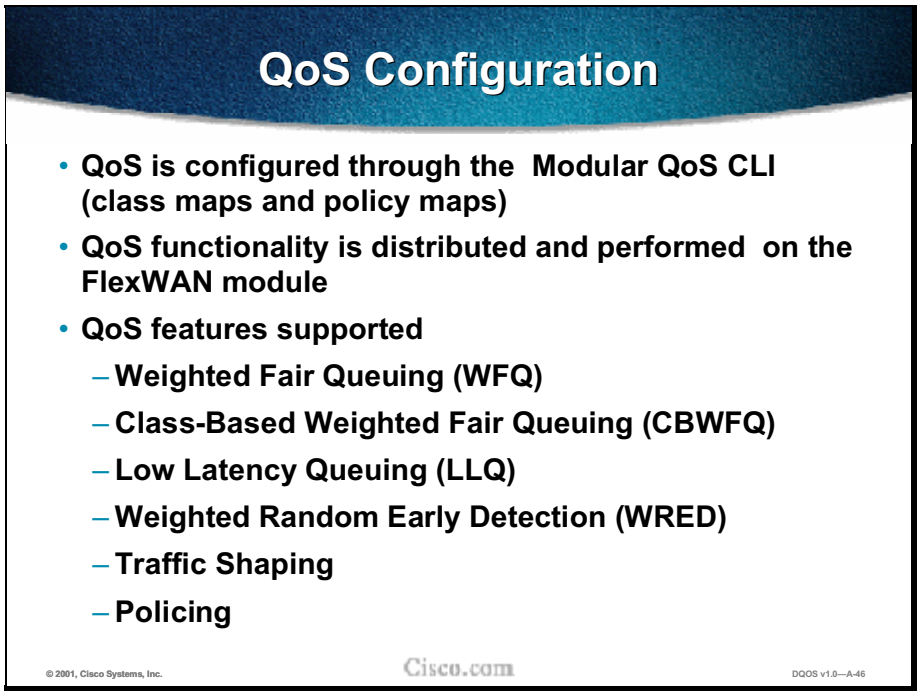

FlexWAN and Optical Services Modules
© 2001, Cisco Systems, Inc.
Cisco.com
DQOS v1.0—A-43



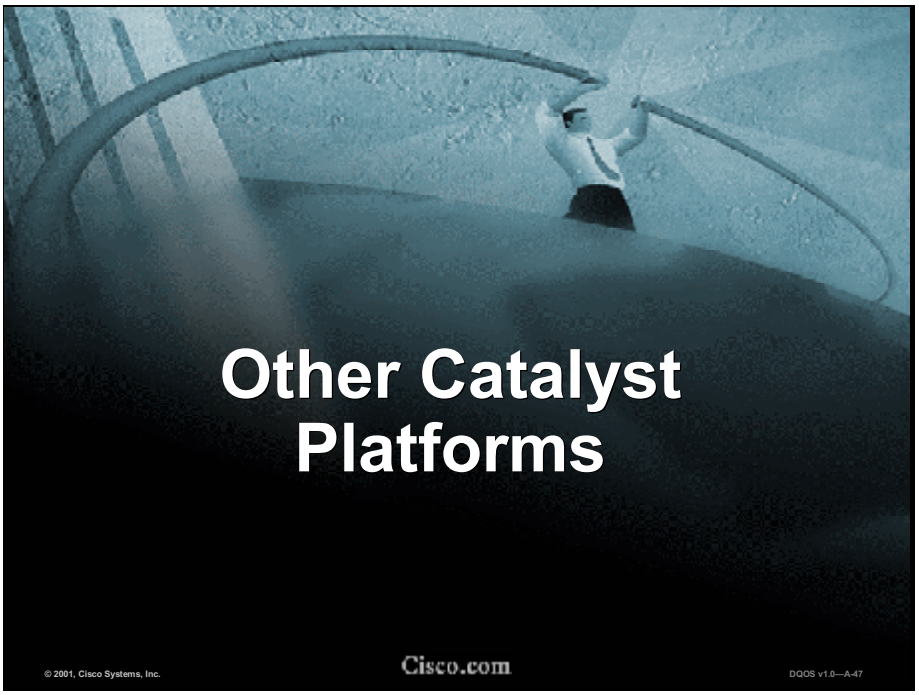
This diagram illustrates the QoS architecture for the FlexWAN module.

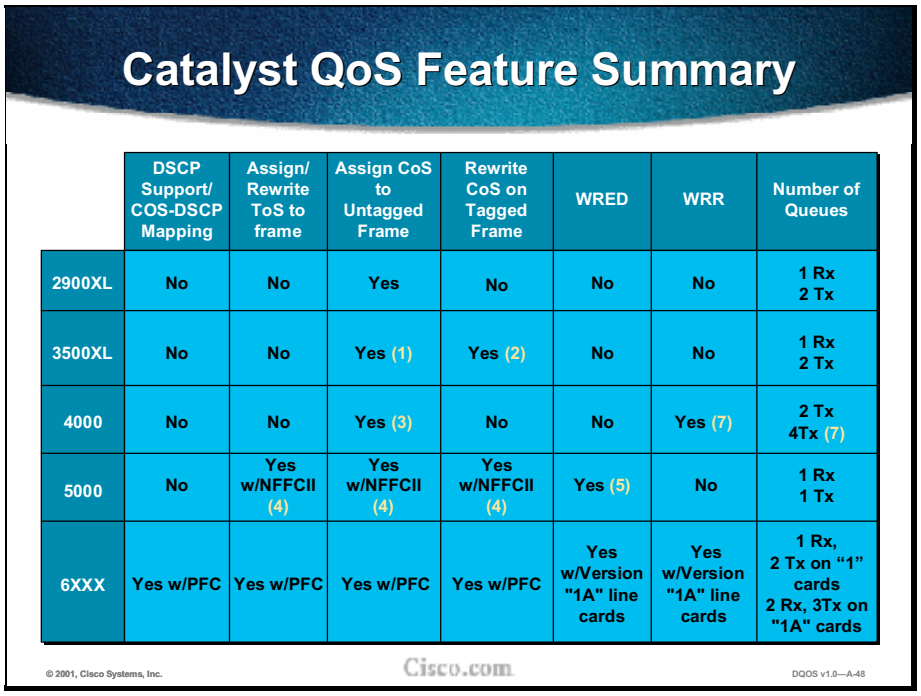


This diagram illustrates the QoS architecture for the Optical Services module.



Both the FlexWAN and Optical Services modules run “router” IOS. QoS features will be dependant on the version of IOS installed on the module.





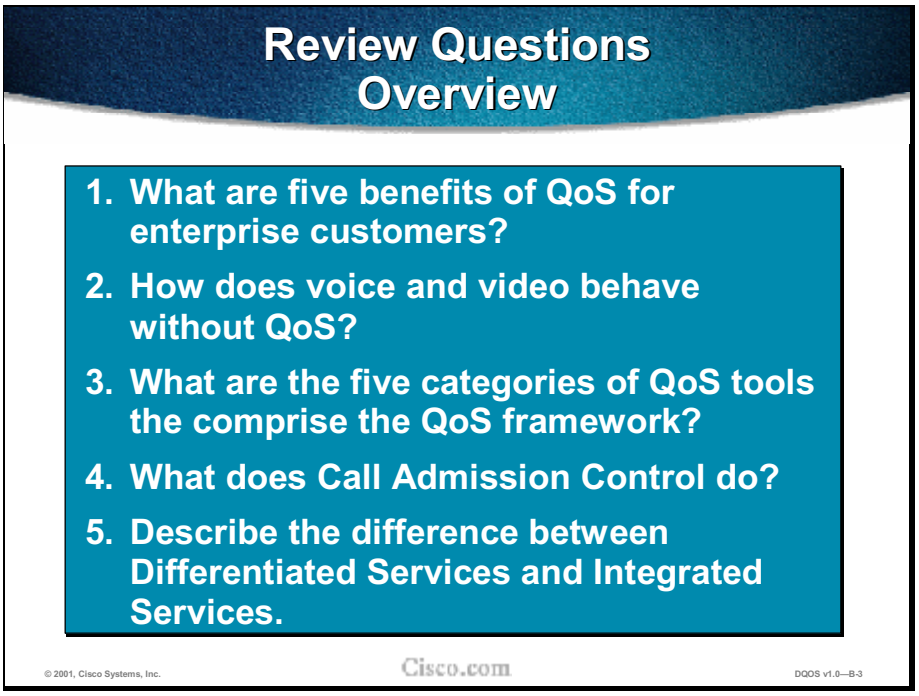
While the 6XXX series switches were used throughout this section as examples, other switch platforms also have QoS features to offer.

Notes

1. CoS rewrite is based on per-port CoS setting.
2. It is only supported on the 3524-PWR and 3548, not supported on other Catalyst 3500XL models.
3. Catalyst 4000 sets CoS based on a *global* setting, not a *port* setting.
4. It requires an Ethernet line card with inline rewrite capability.
5. WRED supported on NFFC-II is used in conjunction with WS-X5234-RJ45, WS-X6239-RJ21, WS-X5236-FX-MT, or WS-X5237-FX-MT (that is, SAINT five-line cards).
6. SAINT4 and earlier line cards use tail drop.
7. WRR and four Transmit queues are only on GE ports of Layer 3 switching line card (WS-X4232-L3). All 10/100 ports on other line cards have two Transmit queues only.

B

Review Questions and Answers



1. What are five benefits of QoS for enterprise customers?

   — Answer: Reliability, control of bandwidth, jitter, delay, ability to classify services, ability to tailor and shape network traffic.

2. How does voice and video behave without QoS?

   — Answer: Jitter, delay, bandwidth congestion, and dropped packets.

3. What are the five categories of QoS tools that comprise the QoS framework?

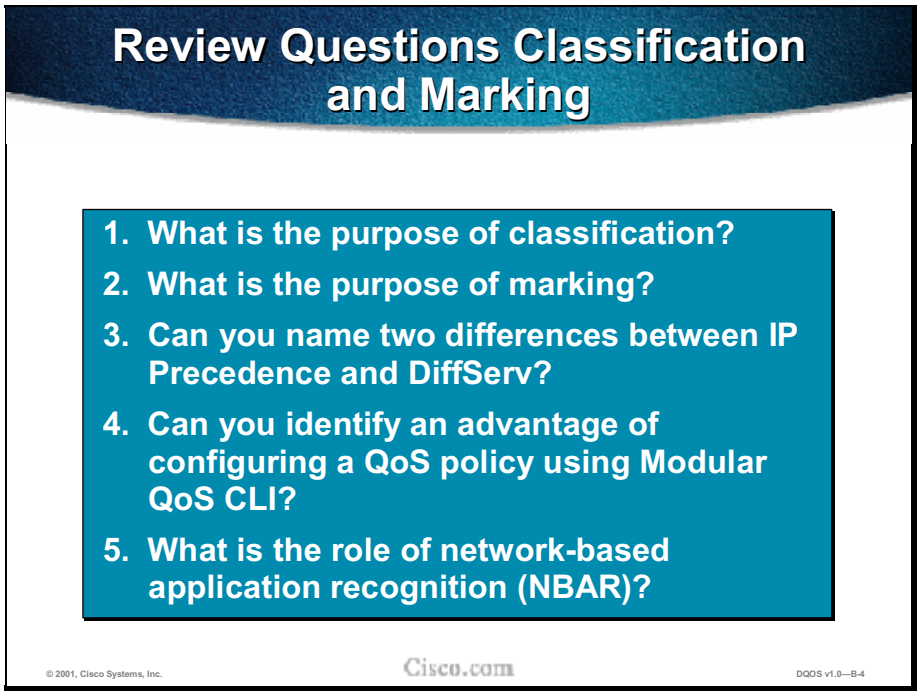
   — Answer: Classification, congestion management, congestion avoidance, policing and shaping, link efficiency; set of tools to manage different types of data traffic according to the needs of traffic and the business.

4. What does call admission control do?

   — Answer: For connection traffic, a method of controlling egress onto the WAN if bandwidth is not available.

5. Describe the difference between Differentiated Services and Guaranteed Services.

   — Answer: Differentiated Services are a set of tools to manage the flow of traffic and thus differentiate how different traffic is to be forwarded. Guaranteed Services is a means to reserve bandwidth to ensure transmission of priority classified network traffic.



1. What is the purpose of classification?

   — Answer: Classification is the selection of traffic to be marked. Traffic is classified as having different priorities.

2. What is the purpose of marking?

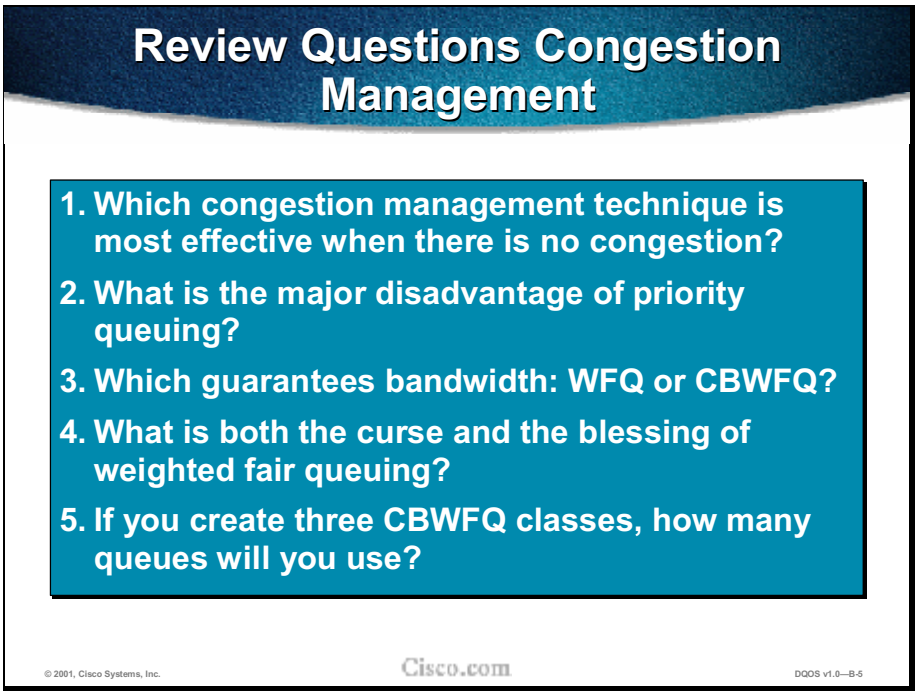
   — Answer: Each classification is marked for identification so QoS methods can be applied. Can you name two differences between IP Precedence and DiffServ?

   — Some possible answers: IP Precedence identifies up to six classes of service, DiffServ up to 64. IP Precedence values have no standardized settings; DiffServ values can be predefined per-hop behaviors (PHBs). With IP Precedence the eight-bit field in the IPv4 header is called the type-of-service field; with DiffServ it is called the DS (Differentiated Services) field.

4. Can you identify an advantage of configuring a QoS policy using Modular QoS CLI?

   — Some possible answers: MQC provides a convenient and efficient user interface for configuring QoS; MQC reduces configuration steps and time; MQC provides a uniform CLI structure for all QoS features. What is the role of network-based application recognition (NBAR)?

   — Answer: NBAR is a classification and protocol-discovery feature.



1. Which congestion management technique is most effective when there is no congestion?

   — Answer: None. Congestion management features operate to control congestion once it occurs.

2. What is the major disadvantage of priority queuing?

   — Answer: Because lower-priority traffic is often denied bandwidth in favor of higher-priority traffic, use of PQ could, in the worst case, result in lower-priority traffic never being sent (protocol starvation).

3. Which guarantees bandwidth: WFQ or CBWFQ?

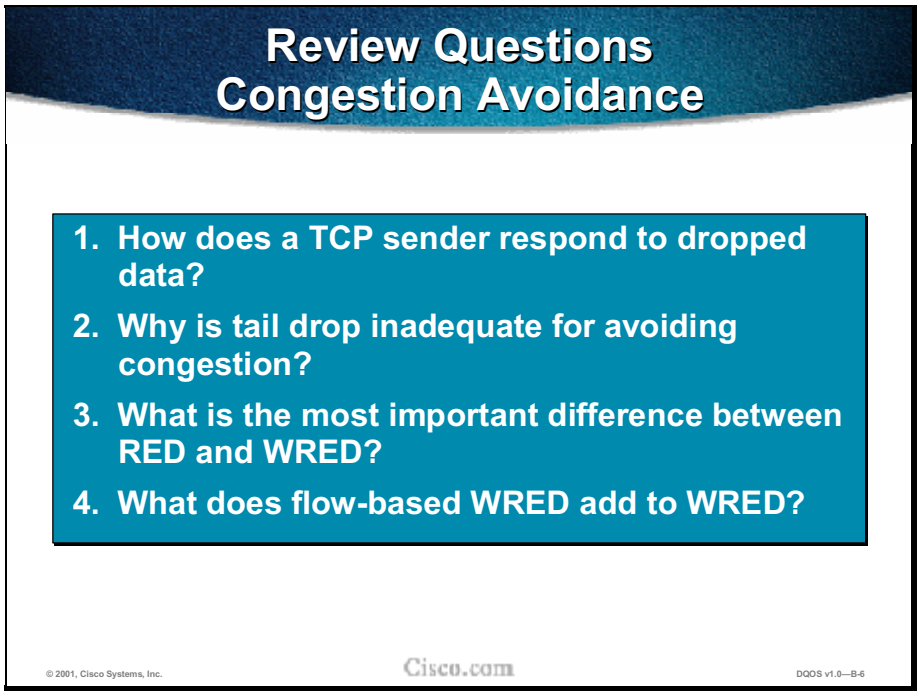
   — Answer: CBWFQ guarantees bandwidth as configured by the bandwidth command.

4. What is both the curse and the blessing of weighted fair queuing?

   — Answer: When there are multiple flows, weighted fair queuing can be too fair, and even the higher-priority traffic begins to experience unacceptable latency.

5. If you create three CBWFQ classes, how many queues will you use?

   — Answer: Three. A queue is reserved for each class, and traffic belonging to a class is directed to the queue for that class.



1. How does a TCP sender respond to dropped data?

   — Answer: When a TCP sender detects a dropped data segment, it retransmits the segment. Then it slows its transmission rate so that it is half of what it was before the drop was detected. This is known as the TCP slow-start mechanism.

2. Why is tail drop inadequate for avoiding congestion?

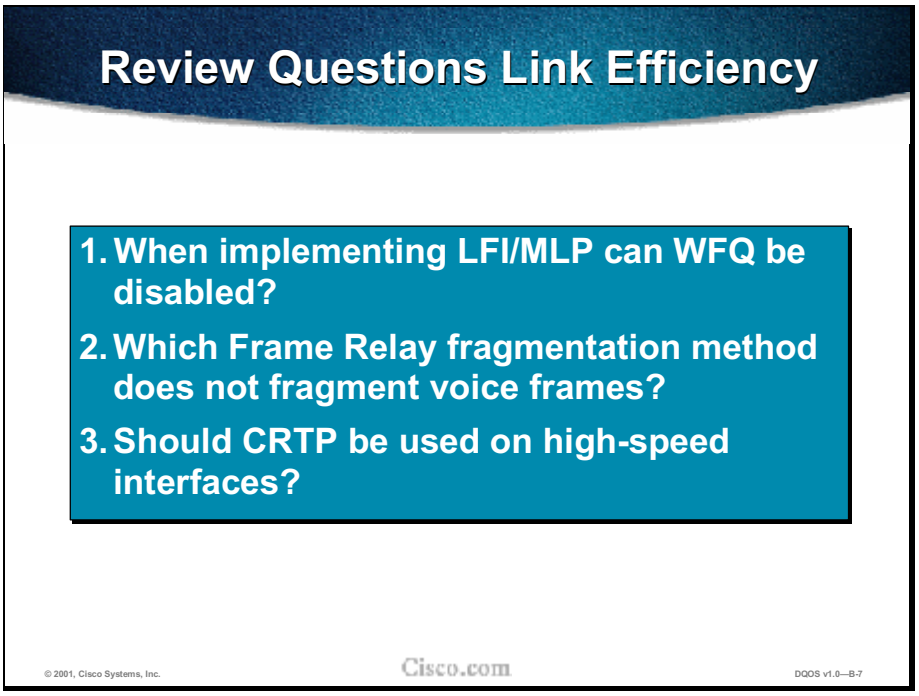
   — Answer: Tail drop treats all traffic equally and does not differentiate between classes of service. If the receiving router drops all traffic that exceeds the queue limit, as is done with tail drop, many TCP sessions then simultaneously go into slow start. Consequently, traffic temporarily slows down to the extreme and then all flows slow-start again. This activity creates a condition of global synchronization, which results in significant link underutilization.

3. What is the most important difference between RED and WRED?

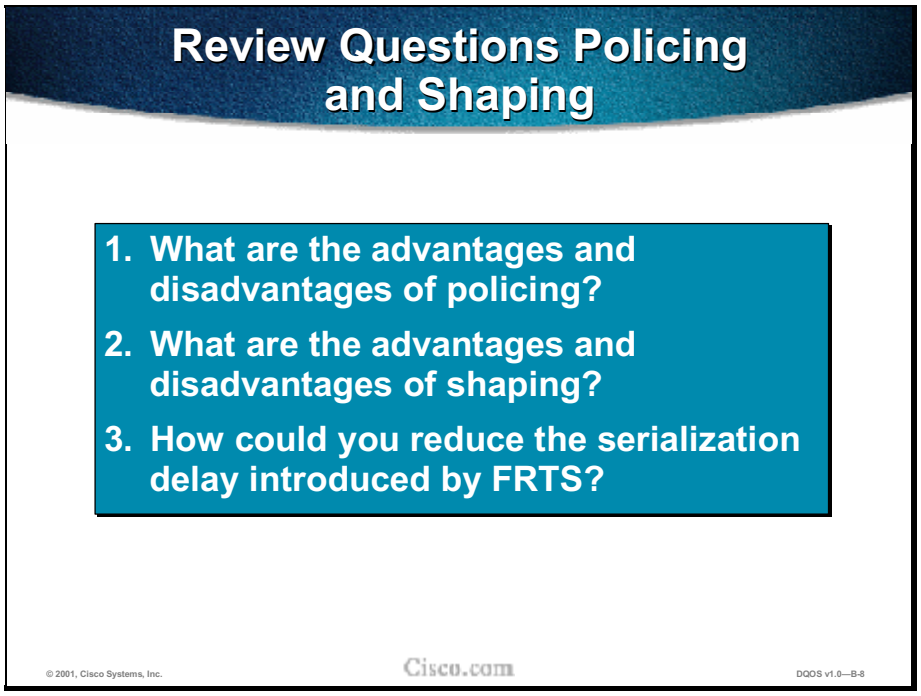
   — Answer: RED drops packets randomly. WRED combines IP Precedence and RED and does packet drops based on IP Precedence.

4. What does flow-based WRED add to WRED?

   — Answer: WRED tends toward bias against fragile flows. Flow-based WRED affords greater fairness to all flows on an interface. Flow-based WRED ensures that each flow does not consume more than its permitted share of the output buffer resources. Flow-based WRED determines which flows monopolize resources and more heavily penalizes those flows.



1. When implementing LFI/MLP, can WFQ be disabled?
   - Answer: No, WFQ must be enabled.
2. Which Frame Relay fragmentation method does not fragment voice frames?
   - Answer: FRF.11 Annex C.
3. Should CRTP be used on high-speed interfaces?
   - Answer: No, CRTP is CPU intensive, and the trade-off is not worth it.



1. What are the advantages and disadvantages of policing?

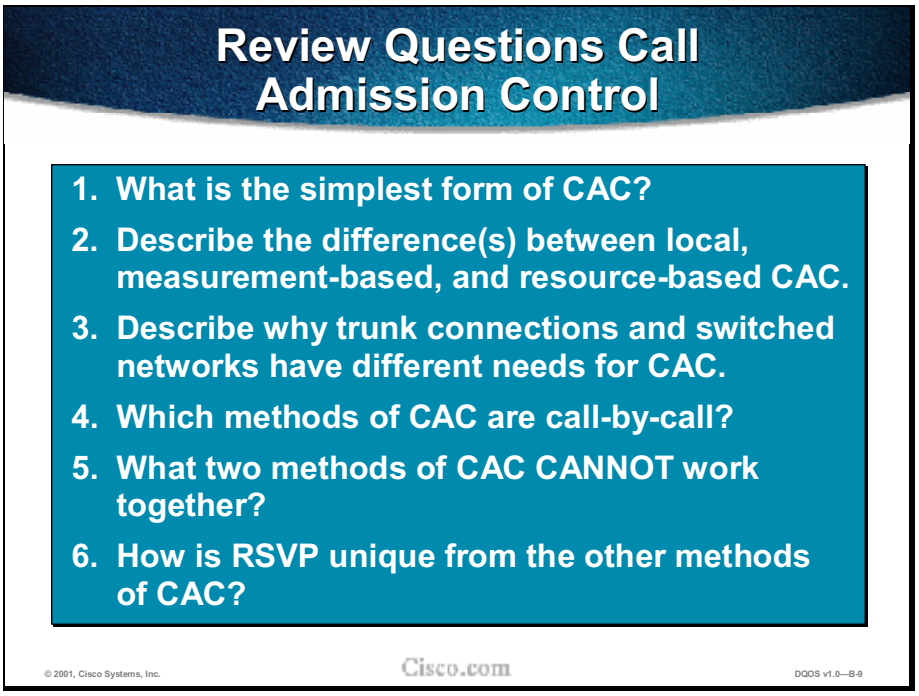
   — Answer: The advantage is having a tool for rate-limiting and speed mismatches. The disadvantage is that traffic shaping introduces jitter into the transmission process.

2. What are the advantages and disadvantages of shaping?

   — Answer: The advantage is a having tool for rate-limiting and speed mismatches. The drawback of policing is dropped packets and increased retransmissions.

3. How could you reduce the serialization delay introduced by FRTS?

   — Answer: Keep the Bc-to-CIR ratio as close to 1 as possible (Reference slide 71).



1. What is the simplest form of CAC?

   — Answer: Physical DS0 limitations.

2. Describe the difference(s) between local, measurement-based, and resource-based CAC.

   — Answer: Local CAC mechanisms function on the OGW. Measurement-based CAC looks ahead into the packet network to gauge the state of the network, in order to determine whether to allow a new call. Resource-based CAC mechanisms either calculate resources that are needed and/or available, or reserves resources for the call.

3. Describe why trunk connections and switched networks have different needs for CAC.

   — Answer: Connection trunk networks have a permanent trunk in place (that is, always ready) and always terminates to a fixed and predetermined destination (that is, point-to-point link). Once the connection is established, the signal is transparent to the gateway.

4. Which methods of CAC are call-by-call?

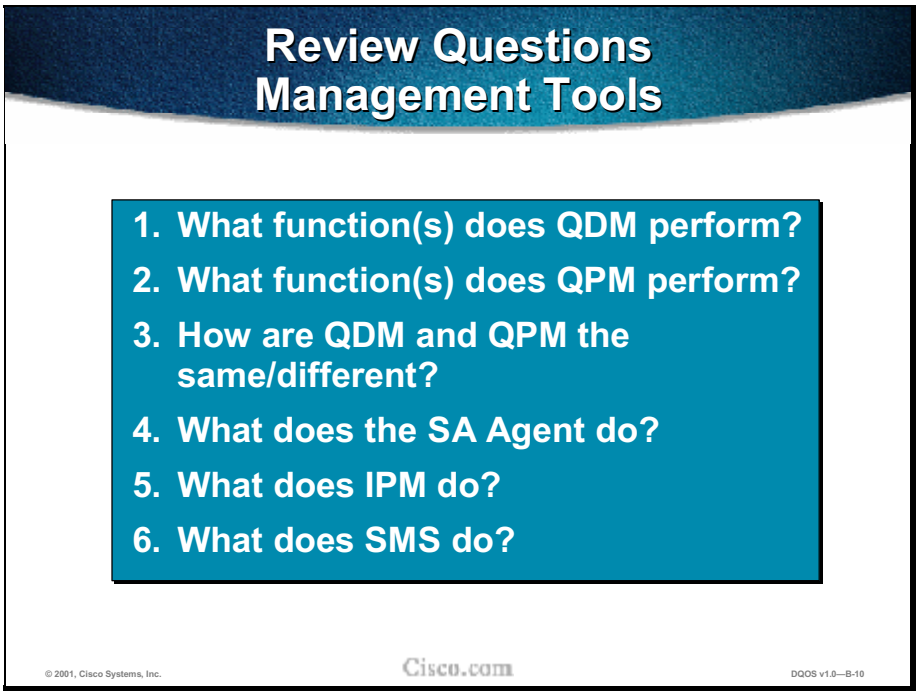
   — Answer: Voice bandwidth for FR, incoming DS0 limitations, GK bandwidth, PSTN fallback, and RSVP.

5. What two methods of CAC *cannot* work together?

   — Answer: RSVP and PSTN fallback.

6. How is RSVP unique from the other methods of CAC?

   — Answer: Reserved bandwidth for the duration of the call.



1. What functions does QDM perform?

   — Answer: Monitors and configures devices, establishes baselines.

2. What functions does QMP perform?

   — Answer: Deploys enterprise-wide QoS policy from a centralized location.

3. How are QDM and QPM the same/different?

   — Answer: They are complementary. QDM configures devices; QPM manages QoS policy from a centralized location.

4. What does the SA Agent do?

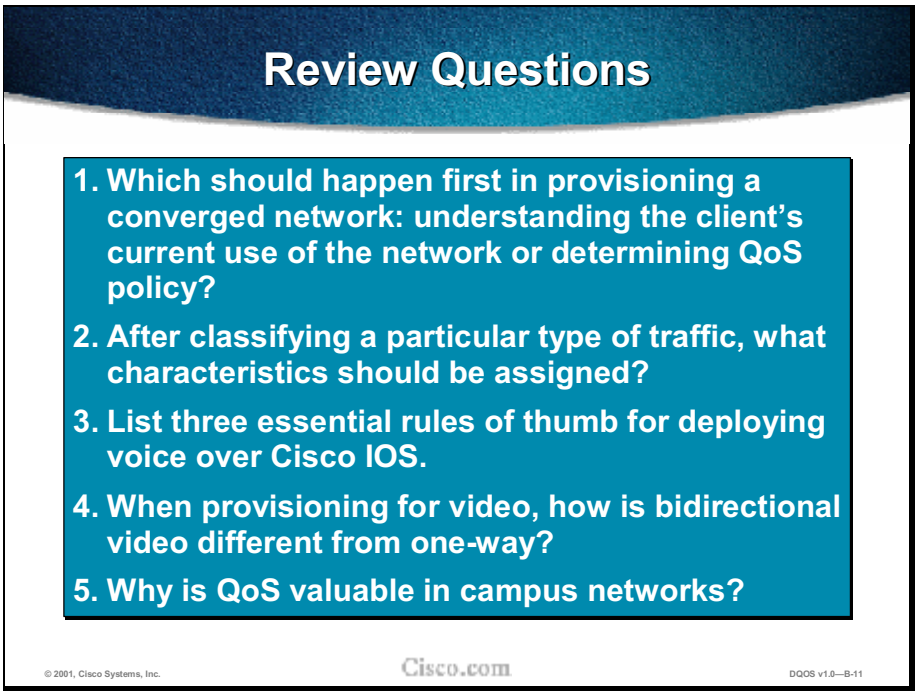
   — Answer: Monitors network performance, network resources, and applications by measuring response times and availability.

5. What does IPM do?

   — Answer: Notifies the network engineer when network response time degrades or a monitored link becomes unavailable and helps pinpoint the device or link causing the problem. Provides graphics for network performance.

6. What does SMS do?

   — Answer: Defines, monitors, and reports on SLAs and a network's ability to support them..



1. Which should happen first in provisioning a converged network: understanding the client's current use of the network or determining QoS policy?

   — Answer: This is a trick question, designed to generate a memorable discussion. The correct answer depends on the client and the situation. In some cases the business may be very clear about the QoS policy it wants; in others the business will be interested in solving problems on their network.

2. After classifying a particular type of traffic, what characteristics should be assigned?

   — Answer: Level of priority, maximum and minimum bandwidth.

3. List three essential rules of thumb for deploying voice over Cisco IOS.

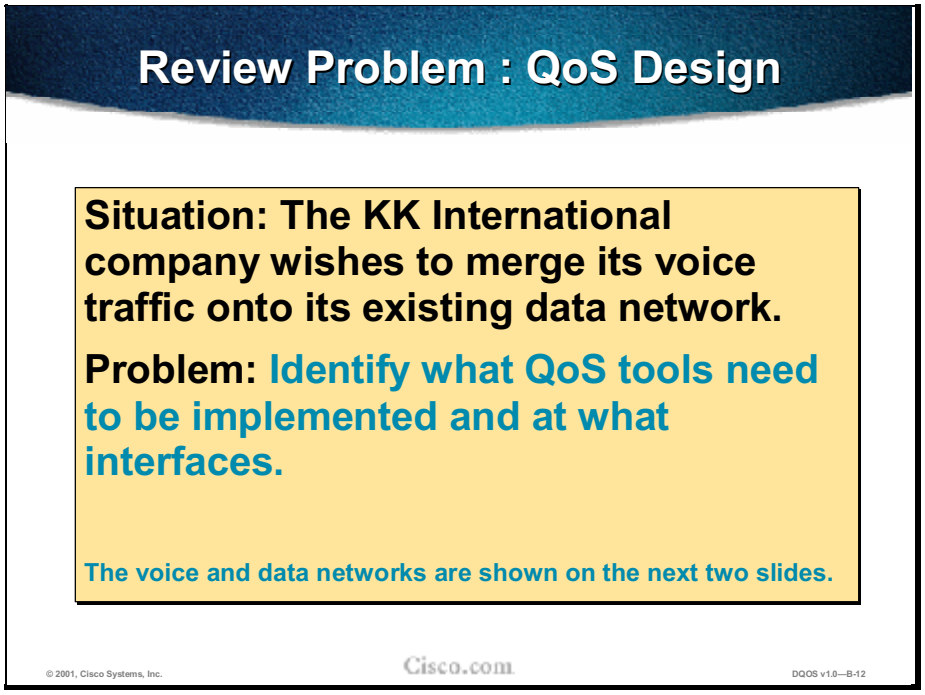
   — Answer: There are many rules of thumb. Here are correct answers. Set IP Prec=5; measure/calculate network packet delay to no more than 150 to 200 ms; do not use WRED for voice.

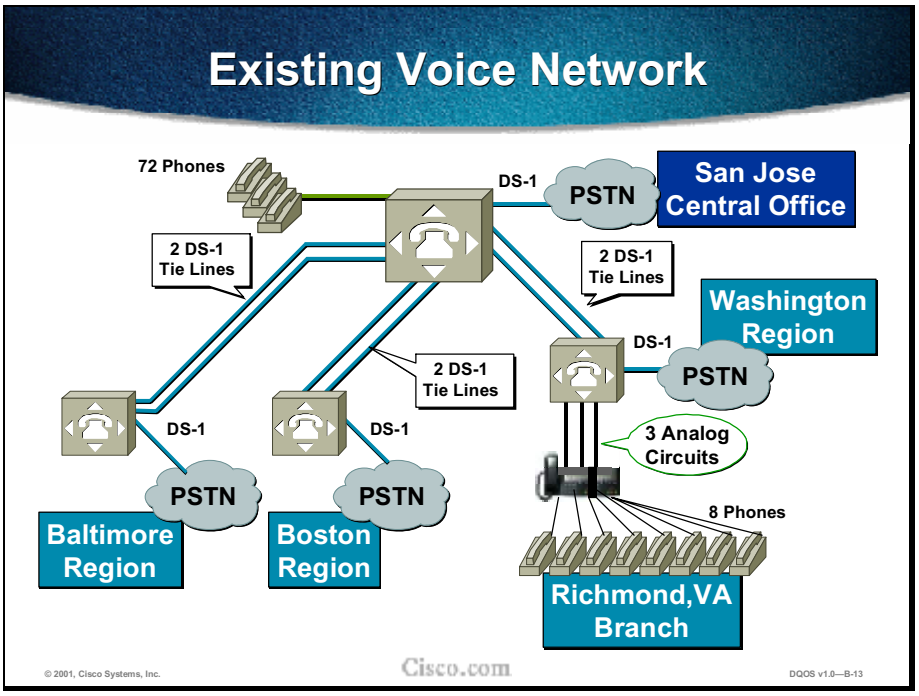
4. When provisioning for video, how is bidirectional video different from one-way?

   — Bidirectional video on slow bandwidth links should queue video traffic with priority queuing capabilities and allocate a bandwidth of 384 kbps. Traffic in excess of 384 kbps would be dropped if the interface becomes congested. Also use an admission control mechanism to ensure that this value is not exceeded. On the other hand one-way video traffic, such as IP/TV, should use a CBWFQ scheme because the delay tolerances are much higher.

5. Why is QoS valuable in campus networks?

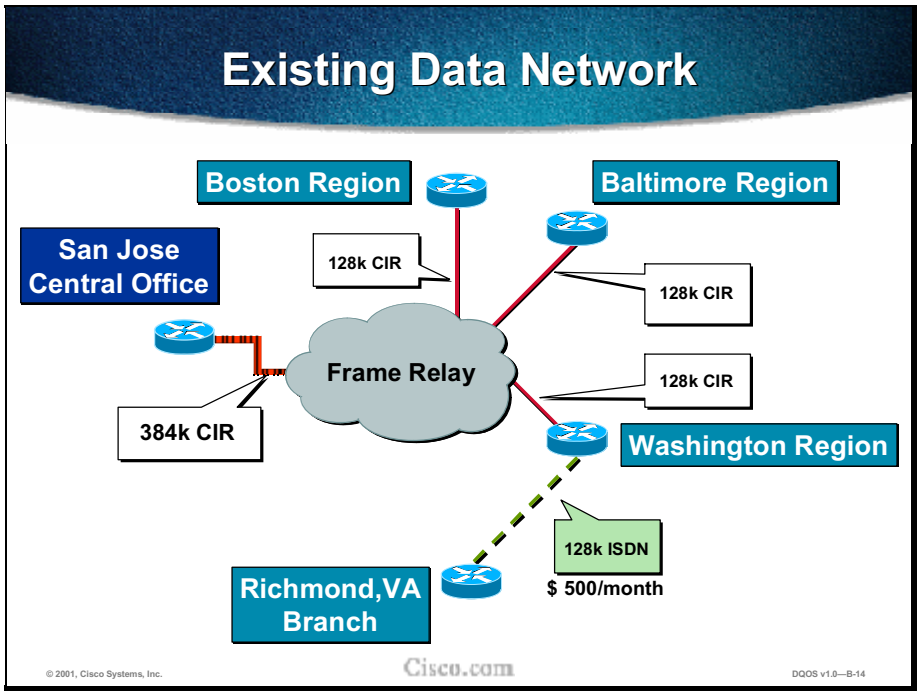
   — Answer: When applications such as voice and video, which are sensitive to loss and delay, began to traverse the data network, network designers gradually came to understand that buffers and not bandwidth are the issue in the campus. Buffers can fill instantaneously. When this occurs, packets can be dropped when attempting to enter the interface buffer. For applications like voice, which are extremely drop intolerant, this results in voice-quality degradation.





The existing voice network for this enterprise is as follows:

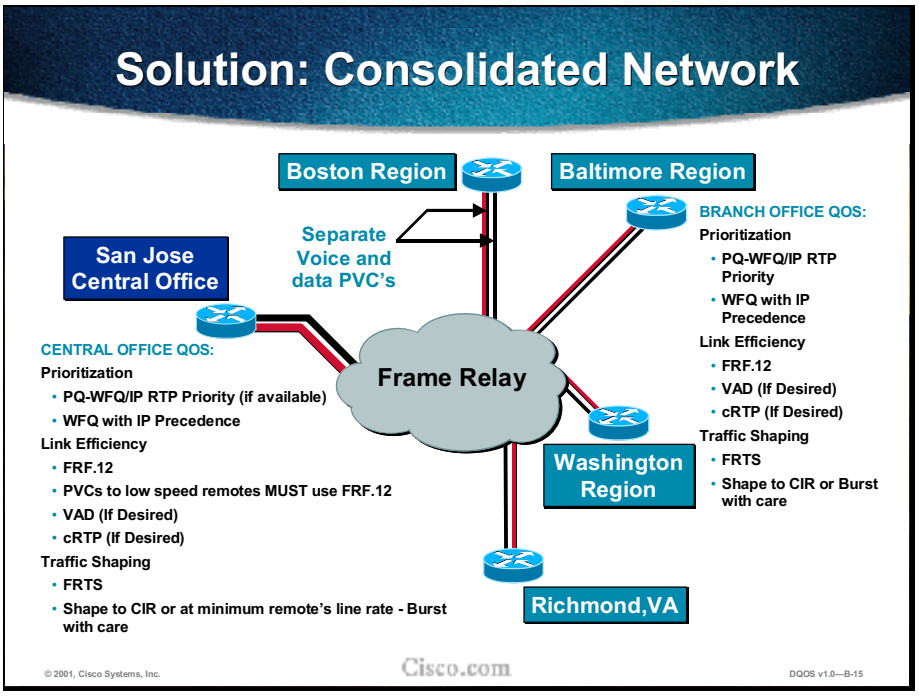
- The central office and the three regional offices have PBXs.
- Each of the four PBXs is connected to the PSTN via a DS-1.
- Each regional office is connected to the central office by two DS-1 circuits provisioned as 48 DS-0s.
- The Washington region is connected to the Richmond branch keyset by three DS-0 analog circuits.
- The Richmond branch also has a connection to the PSTN by a fractional DS-1 configured as four DS-0s.
- The staffing of the locations and the total number of telephones is:
  - Central office—65/72
  - Washington—25/30
  - Baltimore and Boston—20/25 each
  - Richmond—8/8



The customer has an existing Frame Relay network used for data only. There are the following links:

- From the central office in San Jose to the Frame Relay WAN—384-kbps CIR
- From each of the three regional offices—Boston, Baltimore, and Washington to the Frame Relay WAN—128-kbps CIR
- Between the Washington regional office and its Richmond, Virginia, branch—128-kbps ISDN link

Assume that the ISDN link costs $500 per month, flat rate.



The solution shown here is one of several possible ones. The footprint of the consolidated network is kept the same as the old data network, with one exception—the ISDN connection from Richmond to Washington is replaced by a FR circuit. This solution uses separate PVCs for voice and data and uses FRF.12 on the data circuits.

CIR for voice PVCs is calculated according to the compression algorithm used and by using the "total CIR=75% of voice bandwidth" rule.

C

URL Reference Guide

Deploying Cisco Quality of Service for Enterprise Networks

URL Reference

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

Chapter 3 Classification and Marking

For an overview of classification see *Classification Overview* at the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcprt1/qcdclass.htm

For additional information on 802.1p/Q marking see the following URLs:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t3/dtbridge.htm#xtocid114535
http://www.cisco.com/cpress/cc/td/cpress/design/topdown/td0512.htm

For information on Inter-Switch Link (ISL)
http://www.cisco.com/univercd/cc/td/doc/product/software/ios113ed/113ed_cr/switch_c/xcisl.htm
http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_5_5/cnfg_gd/qos.htm

For a detailed discussion of DiffServe classification and marking, see *Implementing DiffServ for End-to-End Quality of Service* at the following URL:
http://cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/dtdfsv.htm

For the IETF DiffServ working group charter
http://www.ietf.org/html.charters/diffserv-charter.html

For a detailed discussion of DiffServe classification and marking, see *Implementing DiffServ for End-to-End Quality of Service* at the following URL:
http://cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/dtdfsv.htm

For a detailed discussion of MQC, see *Modular Quality of Service Command-Line Interface* at the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120xe/120xe5/mqc/mcli.htm

See *Modular Quality of Service Command Line Interface* for a complete descriptions and syntax of all MQC commands. It is located at the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120xe/120xe5/mcli.htm
See also:
http://cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_r/qrdcmd3.htm
http://cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_r/qrdcmd1.htm

For more information on Marking, see the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/cbpmark2.htm#xtocid253640

For a detailed discussion of Class-Based Marking, see *Class-Based Marking* at the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/cbpmark2.htm
For further discussion of reclassification of Layer 2 traffic to Layer 3, see for example *Campus Infrastructure Considerations* at the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/voice/ip_tele/network/dgcampus.htm

Note in the example that you must enable Cisco Express Forwarding (CEF) before you configure NBAR. For more information on CEF, refer to *Cisco Express Forwarding Overview* at the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/switch_c/xcprt2/xcdcef.htm.

For a description of all NBAR features and commands, see *Network Based Application Recognition* at the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/dtnbar.htm
For a succinct description of PBR, see *Quality of Service Fact Sheet* at the following URL:
http://www.cisco.com/warp/public/cc/so/neso/vvda/avvid/eeqos_ds.htm

For additional information on PBR, see also *Quality of Service Overview* at the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcdintro.htm

For more detailed information on PBR, see *Configuring Policy-Based Routing* at the following URL:http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcprt1/qcdpbr.htm

For an overview of configuring ACLs for all protocols see *Access Control Lists: Overview and Guidelines*, at the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_c/scprt3/scdacls.htm

For details on configuring ACLs for the IP protocol, see the "Configuring IP Services" chapter in the *Cisco IOS IP and IP Routing Configuration Guide*, at the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/ip_c/ipcprt1/1cdip.htm
For a detailed description of the role of dial peers, see *Configuring Voice over IP* at the following URL:
http://cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/multi_c/mcprt1/mcdvoip.htm#xtocid232308

For a detailed explanation of configuring dial peers for ip precedence, see the section "Configuring IP Precedence for Dial Peers" in *Configuring Voice over IP* at the following URL:
http://cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/multi_c/mcprt1/mcdvoip.htm#xtocid2323015

For a detailed discussion of both the policing and the classification and marking aspects of CAR, see *Configuring Committed Access Rate* at the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/qos_c/qcpart1/qccar.htm

Chapter 4 Congestion Management

Specific descriptive and configuration details are available in the references provided throughout this chapter. See especially *Congestion Management Overview* and *Configuring Weighted Fair Queuing* at the following URLs:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcprt2/qcdconmg.htm
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcprt2/qcdwfq.htm

For a detailed discussion of congestion management, see *Congestion Management Overview* at the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcprt2/qcdconmg.htm#xtocid182440

To configure PQ and CQ see *Configuring Priority Queuing* at the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcprt2/qcdpq.htm

For a broader discussion of WFQ, see the section "Weighted Fair Queuing" in *Congestion Management Overview* at the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcprt2/qcdconmg.htm#xtocid182444

After the weight for a packet is assigned, the packet is enqueued in the appropriate class queue. CBWFQ uses the weights assigned to the queued packets to ensure that the class queue is serviced fairly. For more details refer to:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120t/120t5/cbwfq.htm
For complete instructions on all available commands, see the section "Class-Based Weighted Fair Queuing Configuration Task List" in *Configuring Weighted Fair Queuing* at the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcprt2/qcdwfq.htm#xtocid243909
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcprt2/qcdconmg.htm
For additional details, see the section "IP RTP Priority Configuration Task List" in *Configuring Weighted Fair Queuing* at the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcprt2/qcdwfq.htm#xtocid2439022

For additional details on configuring Frame Relay and Frame Relay map classes, see *Configuring Frame Relay* at the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/wan_c/wcdfrely.htm

Chapter 5 Congestion Avoidance

For a detailed discussion of TCP congestion behavior see *Geoff Huston, Telstra,* "The Future for TCP," Internet Protocol Journal, Vol. 3, No. 2, June, 2000 at the following URL:
http://www.cisco.com/warp/public/759/ipj_3-3/ipj_3-3_futureTCP.html
For an overview of Congestion Avoidance, see *Congestion Avoidance Overview* at the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcprt3/qcdconav.htm

See also, *DiffServ Compliant Weighted Random Early Detection* at the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/dtdswred.htm
For complete details on configuring WRED and Flow-Based WRED at the interface level, see Configuring Weighted Random Early Detection at the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcprt3/qcdwred.htm

For additional details on DiffServ-compliant WRED, and Per-VC configuration, see the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/dtdswred.htm#xtocid1550421

No URLs cited in Chapter 06 Link Efficiency, 07 Policing and Shaping or 08 Call Admission Control.

Chapter 9 Management Tools

Network managers can configure SA Agent to collect additional information with the optional characteristics option. The full list of optional characteristics can be found at:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/fun_r/frprt3/frd3001c.htm
For a complete list of SA Agent commands refer to:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/fun_r/frprt3/frd3001c.htm

A full description of each of these commands can be found at:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120t/120t5/saaoper.htm#xtocid1543115

QoS Design

This case study can be found at: http://www.cisco.com/warp/public/cc/pd/rt/2600/profiles/mfien_bc.htm

DQoS Lab Topology

Topology

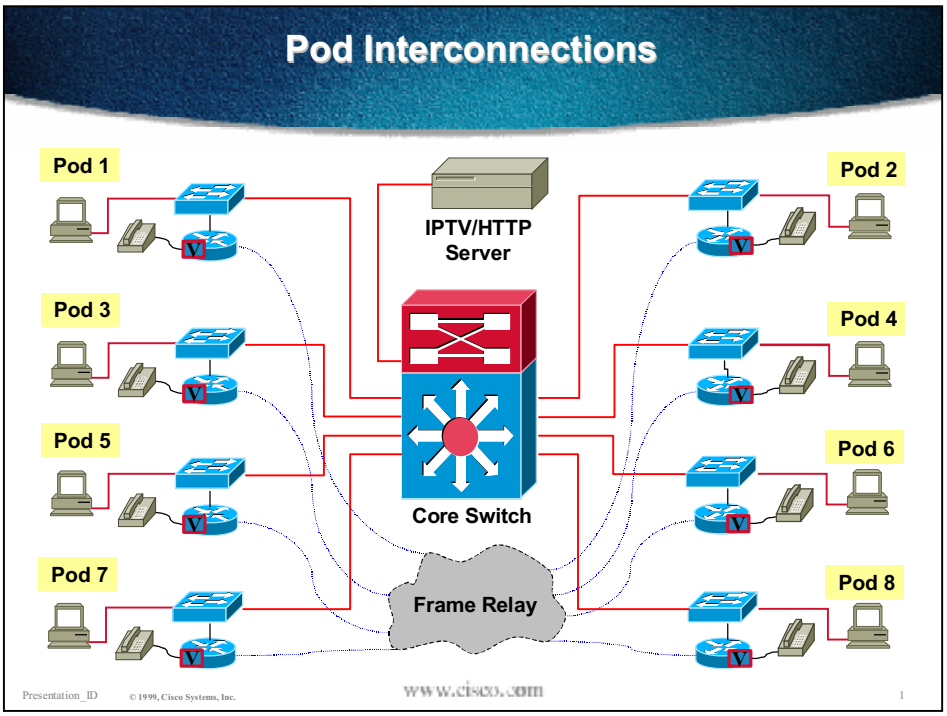


Figure 1 Lab Topology with Pod Interconnections

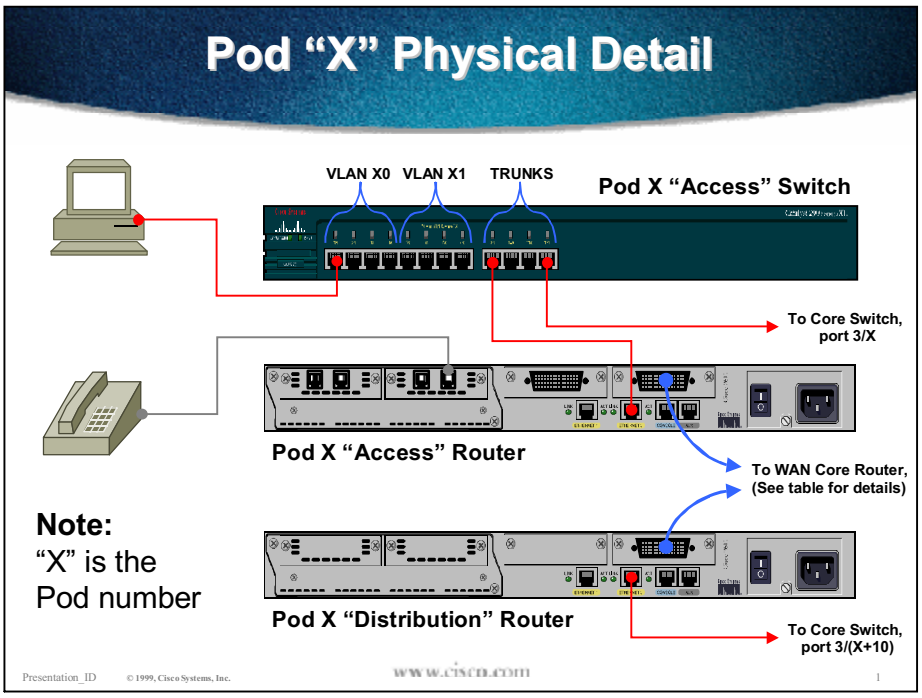


Figure 2 Physical Detail at Pod Level

Note	Regardless of what platform is used for the core switch, its primary function is to provide connectivity between all of the distribution router Ethernet ports and the IPTV/HTTP server. Most Catalyst switches, 2900XL and above, can perform this function with their default configurations.

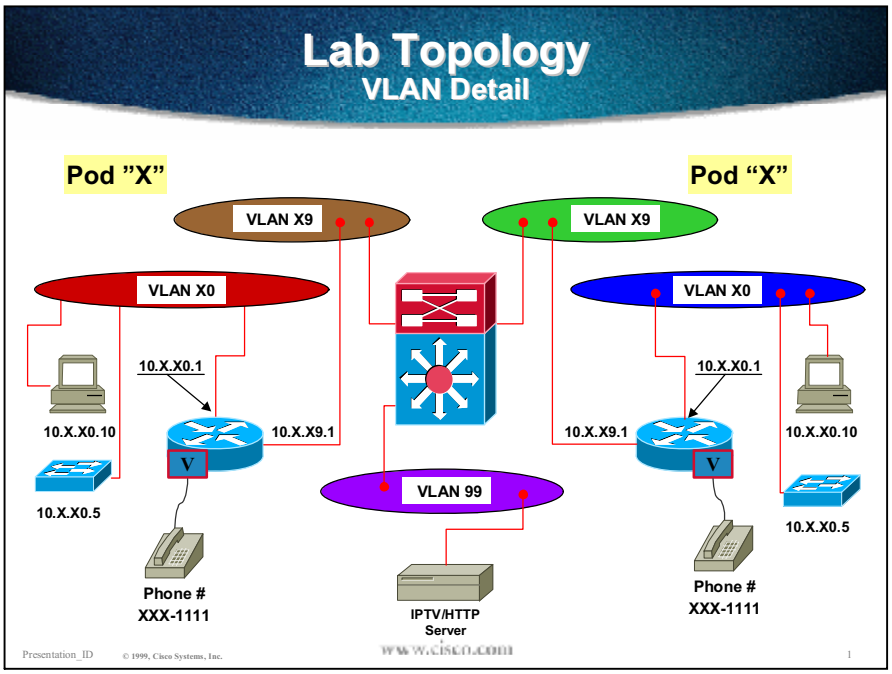


Figure 3 VLAN Detail

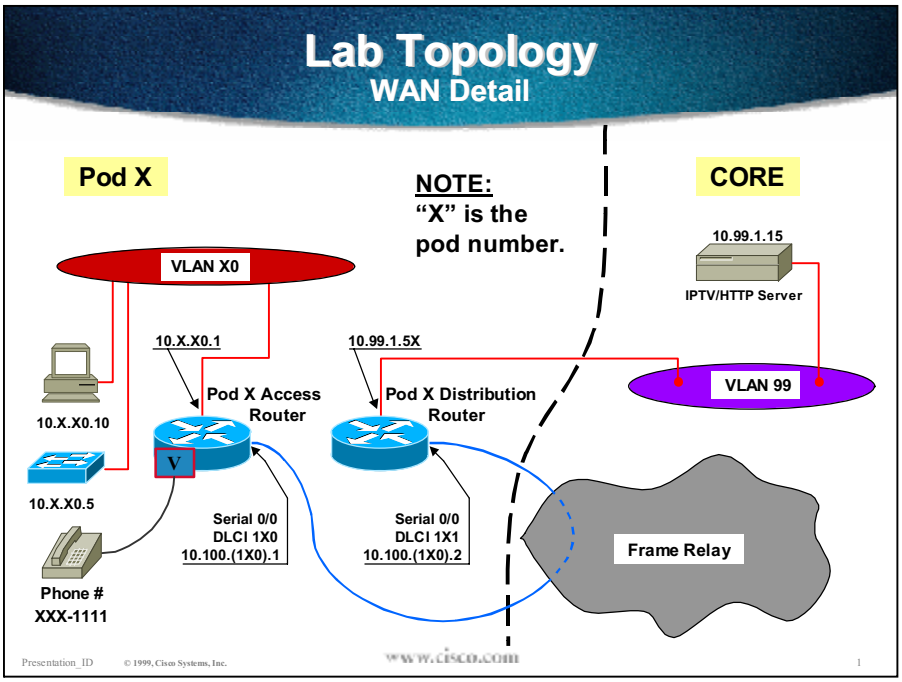


Figure 4 WAN Detail

Addressing/Numbering

The following tables list the addressing and numbering used in the lab.

Note The addressing assumes an IP Phone for each pod, and a Catalyst 6000 with an MSFC (or other route processor) as the core LAN switch. These are optional "value adds" that may be used at the learning partner's discretion.

LAN IP Addressing

	VLAN x0				VLAN x1		VLAN x9	
	Router	Switch	PC	Traffic source	Access Router	IP Phone	Access Router	MSFC
Pod 1	10.1.10.1	10.1.10.5	10.1.10.10	10.1.10.123	10.1.11.1	10.1.11.10	10.1.19.1	10.1.19.2
Pod 2	10.2.20.1	10.2.20.5	10.2.20.10	10.2.20.123	10.2.21.1	10.2.21.10	10.2.29.1	10.2.29.2
Pod 3	10.3.30.1	10.3.30.5	10.3.30.10	10.3.30.123	10.3.31.1	10.3.31.10	10.3.39.1	10.3.39.2
Pod 4	10.4.40.1	10.4.40.5	10.4.40.10	10.4.40.123	10.4.41.1	10.4.41.10	10.4.49.1	10.4.49.2
Pod 5	10.5.50.1	10.5.50.5	10.5.50.10	10.5.50.123	10.5.51.1	10.5.51.10	10.5.59.1	10.5.59.2
Pod 6	10.6.60.1	10.6.60.5	10.6.60.10	10.6.60.123	10.6.61.1	10.6.61.10	10.6.69.1	10.6.69.2
Pod 7	10.7.70.1	10.7.70.5	10.7.70.10	10.7.70.123	10.7.71.1	10.7.71.10	10.7.79.1	10.7.79.2
Pod 8	10.8.80.1	10.8.80.5	10.8.80.10	10.8.80.123	10.8.81.1	10.8.81.10	10.8.89.1	10.8.89.2

Other Addressing/Numbering

	WAN Link				Miscellaneous			
	Access Router		Distribution Router		Access Router Loop 0 Address	Distribution Router Ethernet Address	Analog Phone#	IP Phone#
	DLCI	IP	DLCI	IP				
Pod 1	110	10.100.110.1	111	10.100.110.2	10.1.200.1	10.99.1.51	111-1111	111-2222
Pod 2	120	10.100.120.1	121	10.100.120.2	10.2.200.1	10.99.1.52	222-1111	222-2222
Pod 3	130	10.100.130.1	131	10.100.130.2	10.3.200.1	10.99.1.53	333-1111	333-2222
Pod 4	140	10.100.140.1	141	10.100.140.2	10.4.200.1	10.99.1.54	444-1111	444-2222
Pod 5	150	10.100.150.1	151	10.100.150.2	10.5.200.1	10.99.1.55	555-1111	555-2222
Pod 6	160	10.100.160.1	161	10.100.160.2	10.6.200.1	10.99.1.56	666-1111	666-2222
Pod 7	170	10.100.170.1	171	10.100.170.2	10.7.200.1	10.99.1.57	777-1111	777-2222
Pod 8	180	10.100.180.1	181	10.100.180.2	10.8.200.1	10.99.1.58	888-1111	888-2222

Lab Guide for Chapter 3: Classification and Marking

This document contains a laboratory exercise that uses MQC to classify and mark traffic.

Laboratory Exercise 3: Using the MQC to Classify and Mark traffic.

Complete the following laboratory exercise to practice what you have learned in this chapter.

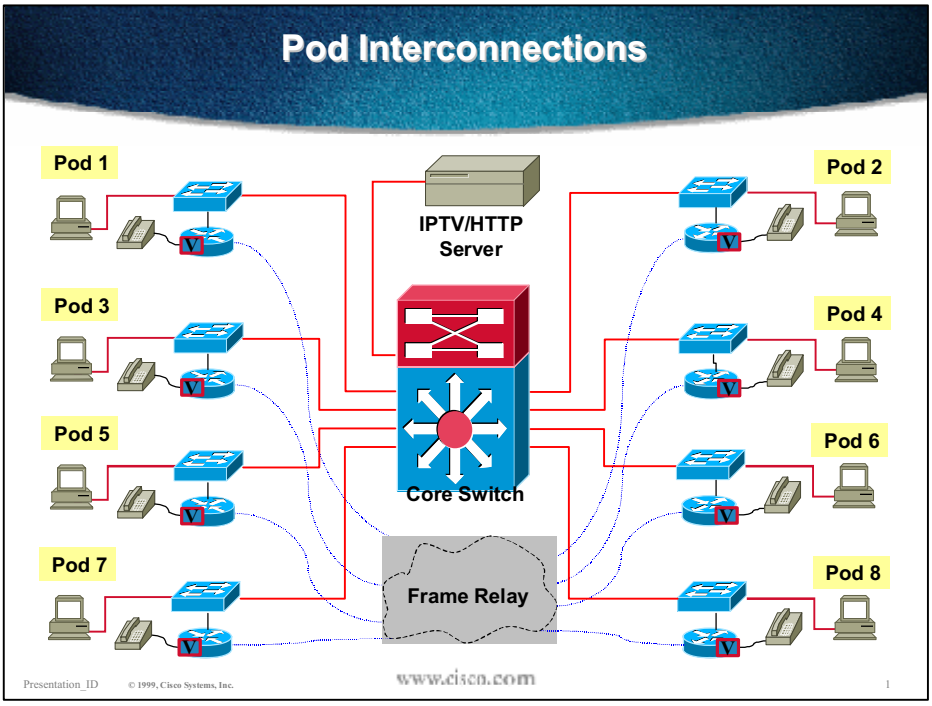
Objectives

In this lab you will complete the following tasks:

- Configure NBAR Protocol Discovery to gather statistics on your router.
- Define Class Maps to classify traffic based on access lists.
- Define Class Maps to classify traffic based on NBAR.
- Configure a policy map to mark DSCP values (Layer 3) based on class maps.
- Configure Catalyst switches to assign a COS to traffic arriving on a given port.
- Configure a policy map-to-map COS values (Layer 2) to DSCP values (Layer 3.)

Visual Objective

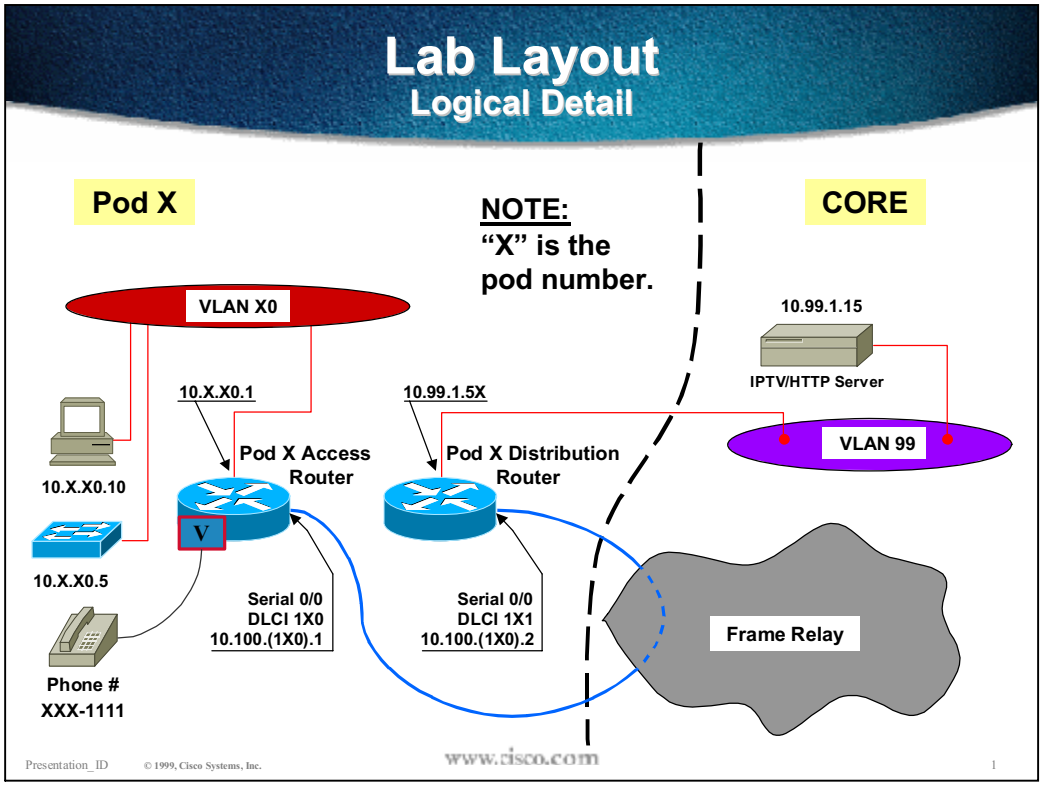
The following figures display the configuration you will be working with in this laboratory exercise.



Addressing/Numbering

The following tables list the addressing and numbering used in this and future labs.

LAN IP Addressing



	VLAN x0				VLAN x1		VLAN x9	
	Router	Switch	PC	Traffic Source	Access Router	IP Phone	Access Router	MSFC
Pod 1	10.1.10.1	10.1.10.5	10.1.10.10	10.1.10.123	10.1.11.1	10.1.11.10	10.1.19.1	10.1.19.2
Pod 2	10.2.20.1	10.2.20.5	10.2.20.10	10.2.20.123	10.2.21.1	10.2.21.10	10.2.29.1	10.2.29.2
Pod 3	10.3.30.1	10.3.30.5	10.3.30.10	10.3.30.123	10.3.31.1	10.3.31.10	10.3.39.1	10.3.39.2
Pod 4	10.4.40.1	10.4.40.5	10.4.40.10	10.4.40.123	10.4.41.1	10.4.41.10	10.4.49.1	10.4.49.2
Pod 5	10.5.50.1	10.5.50.5	10.5.50.10	10.5.50.123	10.5.51.1	10.5.51.10	10.5.59.1	10.5.59.2
Pod 6	10.6.60.1	10.6.60.5	10.6.60.10	10.6.60.123	10.6.61.1	10.6.61.10	10.6.69.1	10.6.69.2
Pod 7	10.7.70.1	10.7.70.5	10.7.70.10	10.7.70.123	10.7.71.1	10.7.71.10	10.7.79.1	10.7.79.2
Pod 8	10.8.80.1	10.8.80.5	10.8.80.10	10.8.80.123	10.8.81.1	10.8.81.10	10.8.89.1	10.8.89.2

Other Addressing/Numbering

	WAN Link				Miscellaneous			
	Access Router		Distribution Router		Access Router Loop 0 Address	Distribution Router Ethernet Address	Analog Phone#	IP Phone#
	DLCI	IP	DLCI	IP				
Pod 1	110	10.100.110.1	111	10.100.110.2	10.1.200.1	10.99.1.51	111-1111	111-2222
Pod 2	120	10.100.120.1	121	10.100.120.2	10.2.200.1	10.99.1.52	222-1111	222-2222
Pod 3	130	10.100.130.1	131	10.100.130.2	10.3.200.1	10.99.1.53	333-1111	333-2222
Pod 4	140	10.100.140.1	141	10.100.140.2	10.4.200.1	10.99.1.54	444-1111	444-2222
Pod 5	150	10.100.150.1	151	10.100.150.2	10.5.200.1	10.99.1.55	555-1111	555-2222
Pod 6	160	10.100.160.1	161	10.100.160.2	10.6.200.1	10.99.1.56	666-1111	666-2222
Pod 7	170	10.100.170.1	171	10.100.170.2	10.7.200.1	10.99.1.57	777-1111	777-2222
Pod 8	180	10.100.180.1	181	10.100.180.2	10.8.200.1	10.99.1.58	888-1111	888-2222

Command List

Refer to this list of commands if you need command assistance during the laboratory exercise.

ROUTER CONFIGURATION COMMANDS		
Prompt	**Command**	**Description**
`Router(config)#`	**`ip cef`**	Enables Cisco Express Forwarding (required for NBAR and service policy operation).
`Router(config-if)#`	**`ip nbar protocol-discovery`**	Enables the gathering of NBAR statistics for a given interface.
`Router(config)#`	**`class-map match-any`** *`name`*	Defines a class map with a specified *name* to match any contents.

ROUTER CONFIGURATION COMMANDS		
Prompt	**Command**	**Description**
`Router(config-cmap)#`	**match cos** *level*	Specifies COS matching criteria for the class map.
`Router(config-cmap)#`	**match access-group** *{number | name name}*	Specifies an access list to be used for matching criteria.
`Router(config)#`	**ip access-list extended** *name*	Defines an extended, named IP access list.
`Router(config-ext-nacl)#`	**permit ip** *source destination*	Permits any packet with a given source and destination address.
`Router(config)#`	**policy-map** policy-name	Specifies the name of the service policy to configure.
`Router(config-pmap)#`	**class** class-name	Specifies the name of a predefined class, which was defined with the **class-map** command, included in the service policy.
`Router(config-pmap-c)#`	**set ip [precedence | dscp]** *level*	Assigns an IP Precedence or DSCP level to packets that match the class-map criteria.
`Router(config-if)#`	**service-policy input** *name*	Applies a policy map to inbound traffic on an interface.
ROUTER MONITORING COMMANDS		
Prompt	**Command**	**Description**
`Router#`	**show class-map**	Displays all traffic class information.
`Router#`	**show class-map** *class-name*	Displays the traffic class information for the user-specified traffic class *name*.
`Router#`	**show policy-map**	Displays all configured service policies.
`Router#`	**show policy-map** *policy-map-name*	Displays the user-specified service policy.
`Router#`	**show policy-map interface** *type number*	Displays configurations and statistics of all input and output policies that are attached to an interface.
`Router#`	**show policy-map interface ?**	Displays options.
`Router#`	**show ip nbar protocol-discovery**	Displays statistics gathered by NBAR.
`Router#`	**clear count** *interface*	Clears the statistical counters for an interface, INCLUDING those for a service policy. NOTE: This can only be done on a "major" interface, and will clear counters associated with its sub-interfaces.
`Router#`	**clear access-list counters**	Resets the counter associated with extended access lists.

2900XL SWITCH COMMANDS		
Prompt	**Command**	**Description**
`Switch(config-if)#`	**switchport priority default** *level*	Sets the COS level for untagged frames coming into an interface.
`Switch#`	**show interface** *interface* **switchport**	Displays the COS assigned to untagged frames arriving on the interface (last line of the output).

Setup

The equipment in the lab has been configured. See the "Lab Layout: Logical Detail" diagram for specifics.

Scenario

This exercise explores the tools that are required for implementing QoS on a network.

The first step in implementing QoS is to determine the existing traffic types on a network. For this, we will use NBAR.

The second step is to sort the traffic using class maps.

Lastly, you will use a policy map and an 802.1q/p priority at a switch port to mark traffic.

In subsequent labs, you will use these tools to classify and mark traffic so that more dramatic service policies can be applied.

Task 1: Configure NBAR Protocol Discovery to gather statistics on your router.

Step 1 Telnet to your distribution router and log in at the privileged level. The telnet password is "cisco" and the enable password is "san-fran."

Step 2 Enable Cisco Express Forwarding (CEF) on your router. This is required for most QoS features, but is not enabled by default.

Step 3 Enable "ip nbar protocol-discovery" on the FastEthernet interface.

Step 4 At the user prompt, issue a "show ip nbar protocol-discovery" command. This command sorts all traffic except "unknown" by volume.

What type of traffic is most common? _____

Step 5 Use a web browser to go to the URL http://10.99.1.15/qosdemo.htm and "refresh" or "<shift>-reload" a few times. At the user prompt, issue a "show ip nbar protocol-discovery" command.

What type of traffic is most common? _____

Step 6 Open up the IPTV viewer application and choose a "program" to watch. At the user prompt, issue a "show ip nbar protocol-discovery" command.

What type of traffic is most common? _____

(Hint: Scroll far down the output of the command.)

Task 2: Define a Class Map to classify traffic based on an Access List.

Step 1 In global configuration on your distribution router, define a named extended access-list called "multicast."

Step 2 Have this access list permit any packet with a destination IP address of 224.0.0.0 – 239.255.255.255. (Hint: use the command "permit ip any 224.0.0.0 15.255.255.255," which matches any packet with a destination address in the multicast range.)

Step 3 Define a class map called "video," which uses the "multicast" access list as match criteria.

Step 4 From user mode, enter the "show class-map" and the "show access-lists" commands to confirm your configuration.

Task 3: Define Class Maps to classify traffic with NBAR.

Step 1 Still on your distribution router, define a class map called "ftp," which uses NBAR to match FTP traffic.

Step 2 Define a class map called "http1" to match HTTP traffic, with the string "important" anywhere in the URL.

Step 3 Define a class map called "http2" to match HTTP traffic, with the string "not_so" anywhere in the URL.

Step 4 Confirm your configuration.

Task 4: Configure a policy map to mark DSCP values based on classification.

Step 1 On your distribution router, define a policy map named "ingress."

Step 2 Issue the "class video" command to define a policy for the "video" class.

Step 3 Enter the command "set ip dscp ?" to see the DSCP values.

Step 4 Set a DSCP value of "AF41" for all "video" traffic.

Step 5 For classes "ftp," "http1," and "http2," assign DSCP values of "AF31," "AF21," and "AF22," respectively.

Step 6 Apply your "ingress" service policy to inbound traffic on interface FastEthernet 0/0.

Step 7 From user mode, enter the "show policy-map interface fastethernet0/0" command to confirm your configuration.

Task 5: Test your configuration

Step 1 Close any open IPTV or web browser sessions on your PC.

Step 2 Clear the counters on your distribution router's FastEthernet 0/0 interface.

Step 3 Use the "show policy-map interface" command to see how many matches you've had for each class.

Step 4 Use your web browser to open a session to 10.99.1.15/qosdemo.htm. Refresh the page to ensure traffic from the server.

Step 5 Use the "show policy-map interface" command to see how many matches you've had for the "http1" and "http2" classes.

Step 6 Open the IPTV viewer and enjoy a movie.

Step 7 Use the "show policy-map interface" command to see how many matches you've had for the "video" class.

Task 6: Configure the 2900XL switch to apply a COS.

Step 1 Telnet to your 2900XL switch.

Step 2 Configure 802.1q/p priority of 1 on the PC's interface (FastEthernet 0/1.)

Step 3 From user mode, enter the "show interface fastethernet0/1 switchport" command to confirm your configuration.

Task 7: Define a Policy to map Layer Two COS to Layer Three DSCP.

Step 1 In global configuration on your access router, define a class map named "cos1." (Remember to "match-any.")

Step 2 Configure class map "cos1" to match any traffic with a COS of 1.

Step 3 Define a policy map "l2tol3" to set a DSCP value of 'AF11' for all 'cos1' class traffic.

Step 4 Issue the "service-policy l2tol3" command to implement the policy on the proper sub-interface of the Fa0/0 interface.

Step 5 From user mode, enter the "show policy-map interface <interface>" command to confirm your configuration.

Task 8: Clean up

Step 1 Remove the "ingress" service policy from the distribution router's Fa0/0 interface, then save the policies you have built on both of your routers. You will be using them in future labs.

Completion Criteria:

You have successfully completed this laboratory exercise if the appropriate traffic was matched by the appropriate class maps.

Configuration Examples (arranged by Task):

```
Router#config terminal

Router(config)#ip cef

Router(config)#interface FastEthernet0/0

Router(config-if)#ip nbar protocol-discovery



Router(config)#ip access-list extended multicast

Router(config-ext-nacl)#permit ip any 224.0.0.0 15.255.255.255

Router(config)# class-map match-any video

Router(config-cmap)# match access-group name video



Router(config)# class-map match-any ftp
```



```
Router(config-cmap)# match protocol ftp

Router(config-cmap)# class-map match-any http1

Router(config-cmap)# match protocol http url "*important*"

Router(config-cmap)# class-map match-any http2

Router(config-cmap)# match protocol http url "*not_so*"


Router(config)#policy-map ingress

Router(config-pmap)#class http1

Router(config-pmap-c)#set ip dscp af21

Router(config-pmap-c)#class http2

Router(config-pmap-c)#set ip dscp af22

Router(config-pmap-c)#class ftp

Router(config-pmap-c)#set ip dscp af31

Router(config-pmap-c)#class video

Router(config-pmap-c)#set ip dscp af41

Router(config-pmap-c)#interface FastEthernet0/0

Router(config-if)#service-policy input ingress


Switch(config)#interface FastEthernet0/1

Switch(config-if)#switchport priority default 1


Router(config)#class-map match-any cos1

Router(config-cmap)#match cos  1


Router(config)#policy-map l2tol3

Router(config-pmap)#class cos1

Router(config-pmap-c)#set ip dscp af11
```

```
Router(config)#interface FastEthernet0/0.20

Router(config-if)#service-policy input l2tol3
```


Lab Guide for Chapter 4: Congestion Management

Overview

This chapter contains two laboratory exercises that implement your knowledge of Congestion Management.

The first lab, “WFQ Operation,” has you configure Weighted Fair Queuing, then configure IP Precedence on a voice over IP Dial Peer to take advantage of the “weighted” facet of Weighted Fair Queuing.

The second lab, “CBWFQ/LLQ Operation,” has you configure Class Based Weighted Fair Queuing and Low Latency Queuing.

Laboratory Exercise 4.1: WFQ Operation

Complete the following laboratory exercise to practice what you learned so far in this chapter.

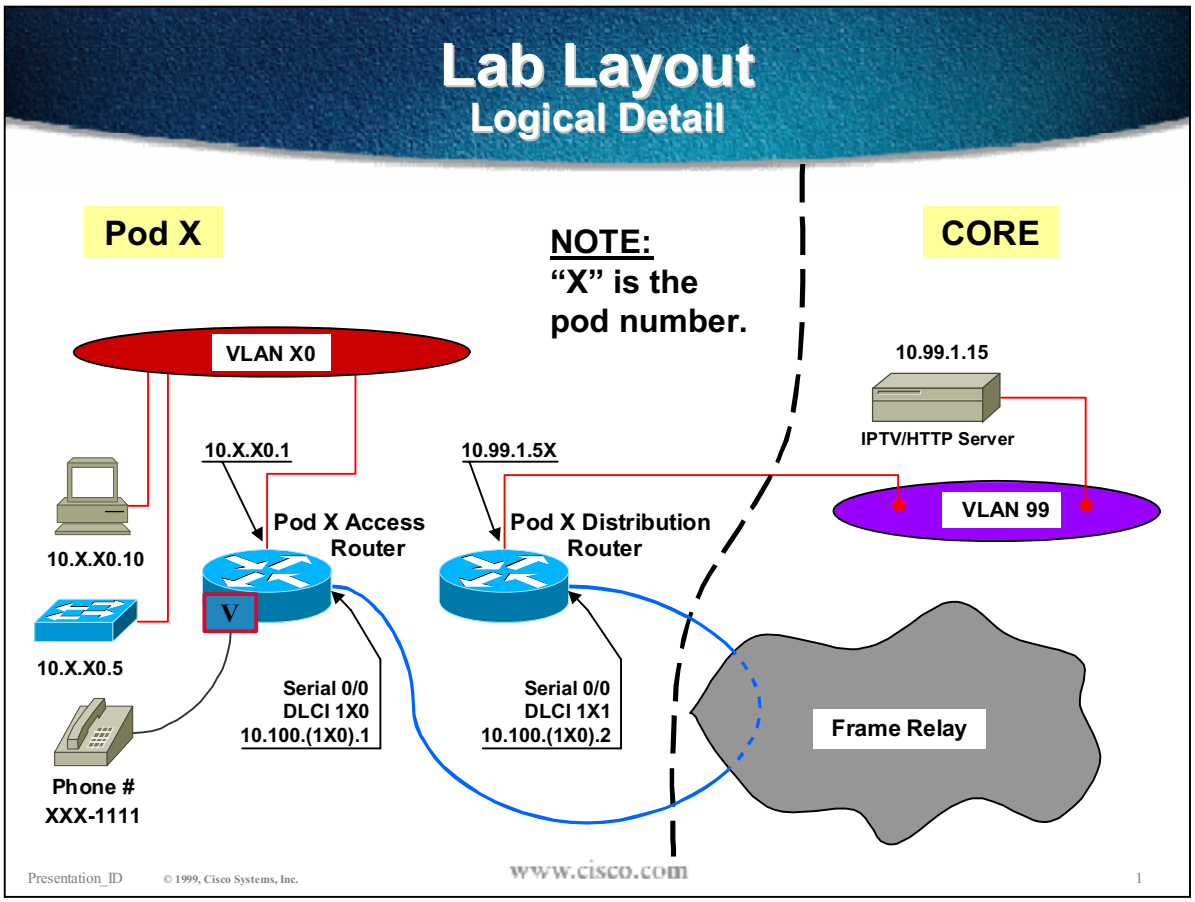
Objectives

In this lab you will complete the following tasks:

- Configure Weighted Fair Queuing.
- Apply a DSCP level to video traffic to improve its operation.

Visual Objective

The following figure illustrates the configuration you will be working with in this exercise.



WAN Connections

In this exercise, you will be connecting to the Core over your WAN links, and applying QoS to that serial connection. The following table provides the specifics of the WAN connections.

	WAN Link				Miscellaneous			
	Access Router		Distribution Router		Access Router Loop 0 Address	Distribution Router Ethernet Address	Analog Phone#	IP Phone#
	DLCI	IP	DLCI	IP				
Pod 1	110	10.100.110.1	111	10.100.110.2	10.1.200.1	10.99.1.51	111-1111	111-2222
Pod 2	120	10.100.120.1	121	10.100.120.2	10.2.200.1	10.99.1.52	222-1111	222-2222
Pod 3	130	10.100.130.1	131	10.100.130.2	10.3.200.1	10.99.1.53	333-1111	333-2222
Pod 4	140	10.100.140.1	141	10.100.140.2	10.4.200.1	10.99.1.54	444-1111	444-2222
Pod 5	150	10.100.150.1	151	10.100.150.2	10.5.200.1	10.99.1.55	555-1111	555-2222
Pod 6	160	10.100.160.1	161	10.100.160.2	10.6.200.1	10.99.1.56	666-1111	666-2222
Pod 7	170	10.100.170.1	171	10.100.170.2	10.7.200.1	10.99.1.57	777-1111	777-2222
Pod 8	180	10.100.180.1	181	10.100.180.2	10.8.200.1	10.99.1.58	888-1111	888-2222

Command List

The following table lists the new commands you will use in this exercise in logical order. Refer to this list, or lists in previous exercises, if you need command assistance.

ROUTER COMMANDS		
Prompt	**Command**	**Description**
`Router(config-if)#`	**fair-queue**	Enables weighted fair queuing on an interface.
`Router(config)#`	**class-map match-any** *name*	Defines a class map with a specified *name* to match any contents.
`Router(config-cmap)#`	**match access-group** {*number* | **name** *name*}	Specifies an access list to be used for matching criteria.
`Router(config)#`	**ip access-list extended** *name*	Defines an extended, named IP access list.
`Router(config-ext-nacl)#`	**permit ip** *source destination*	Permits any packet with a given source and destination address.
`Router(config)#`	**policy-map** policy-name	Specifies the name of the service policy to configure.
`Router(config-pmap)#`	**class** class-name	Specifies the name of a predefined class, which was defined with the **class-map** command, included in the service policy.
`Router(config-pmap-c)#`	**set ip [precedence | dscp]** *level*	Assigns an IP Precedence or DSCP level to packets that match the class-map criteria.
`Router(config-if)#`	**service-policy input** *name*	Applies a policy map to inbound traffic on an interface.
`Router#`	**show policy-map interface** *type number*	Displays configurations and statistics of all input and output policies that are attached to an interface.

Setup

Your instructor has configured the core so that all pods are connected to the "backbone" (VLAN 99) via your distribution routers. You will be watching a 600Kbps video stream, loading large web pages, and your WAN link is 730Kbps. See the "Logical Topology" diagram for interconnection details.

Scenario

You are the network administrator for Cowville Chip Industries (CCI,) a company specializing in wood chip products. CCI uses their network for document sharing, e-mail, video, and IP telephony, and has a 730Kbps frame relay connection between facilities and to the internet.

Task1: Configure Weighted Fair Queuing

Step 1 Open up your IPTV viewer and watch a "movie." Then, open the "qosdemo.htm" web page. How is the quality of the movie?

Step 2 As the web page loads, attempt to place a phone call to another pod. Was it successful? If so, how was the voice quality?

Step 3 Telnet to your **distribution** router. Because of the congestion, you may want to close the video session first.

Note WFQ is enabled by default on most WAN links under 2Mbps. In this case, your instructor has disabled it to make your lives more interesting.

Step 4 Enable Weighted Fair Queuing on the serial port of the distribution router. Confirm the configuration with a "show interface serial0/0" command.

Step 5 Telnet to your access router and enable WFQ on its serial interface. Execute a "show interface serial0/0" command to confirm the configuration.

Step 6 Once WFQ has been enabled on both routers, again open up a movie and place an analog phone call.

Has the audio quality improved? _____

Why or why not? _____

(Remember, Fair Queuing gives preference to low volume, bursty traffic. Also be aware that your movie is actually two streams, one for video and one for audio.)

Task 2: Apply a DSCP value to Video traffic to take advantage of the "weighted" facet of WFQ

Step 1 Re-apply your "ingress" service policy to the FastEthernet 0/0 port of your distribution router. (Refer to the Classification and Marking lab if you need details.)

Step 2 Again, open up a movie.

Has the video quality improved? _____

Why or why not? _____

Task 3: Clean up.

Step 1 Remove WFQ from your access and distribution routers.

Step 2 Save your router configuration.

Completion Criteria:

You have successfully completed this laboratory exercise if you observed a noticeable improvement in the network performance for audio and video.

Configuration Examples (arranged by Task):

```
Router#config terminal

Router(config)#interface Serial0/0

Router(config-if)#fair-queue



Router(config-pmap-c)#interface FastEthernet0/0

Router(config-if)#service-policy input ingress
```

Laboratory Exercise 4.2: CBWFQ/LLQ Operation

Complete the following laboratory exercise to practice what you have learned so far in this chapter.

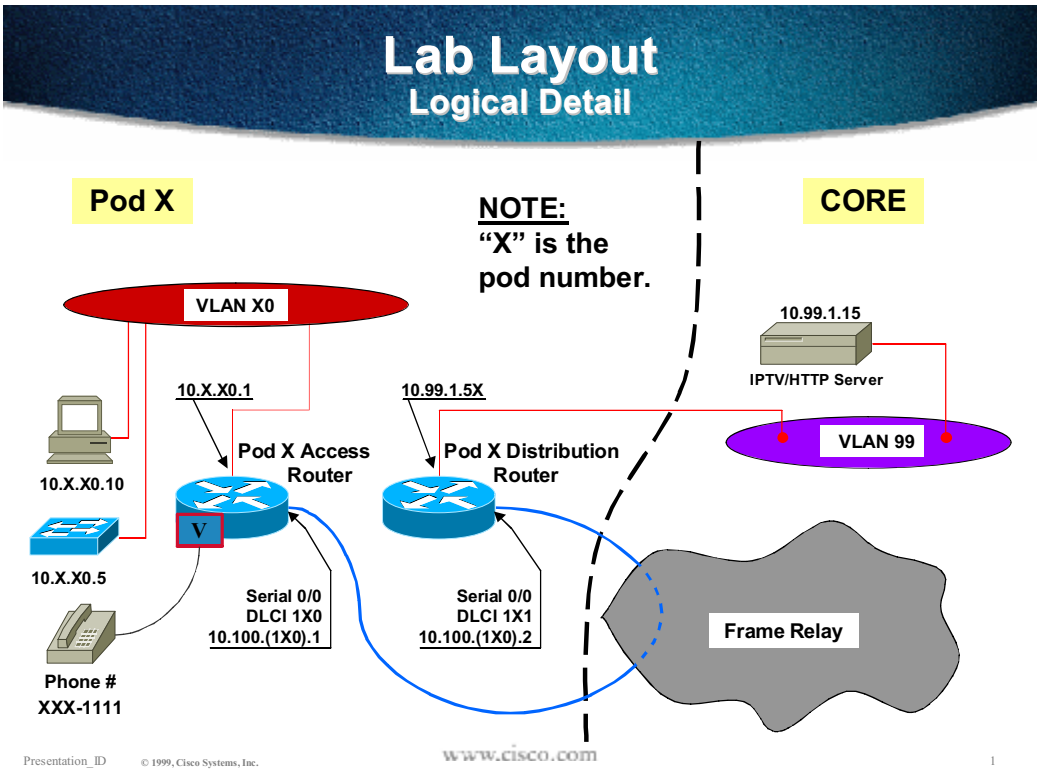
Objectives

In this lab you will complete the following tasks:

- Configure Class Based Weighted Fair Queuing.
- Configure Low Latency Queuing.

Visual Objective

The figure below illustrates the configuration you will be working with in this exercise.



WAN Connections

In this exercise, you will be connecting to the Core over your WAN links, and applying QoS to that serial connection. The following table provides the specifics of the WAN connections.

	WAN Link				Miscellaneous			
	Access Router		Distribution Router		Access Router Loop 0 Address	Distribution Router Ethernet Address	Analog Phone#	IP Phone#
	DLCI	IP	DLCI	IP				
Pod 1	110	10.100.110.1	111	10.100.110.2	10.1.200.1	10.99.1.51	111-1111	111-2222
Pod 2	120	10.100.120.1	121	10.100.120.2	10.2.200.1	10.99.1.52	222-1111	222-2222
Pod 3	130	10.100.130.1	131	10.100.130.2	10.3.200.1	10.99.1.53	333-1111	333-2222
Pod 4	140	10.100.140.1	141	10.100.140.2	10.4.200.1	10.99.1.54	444-1111	444-2222
Pod 5	150	10.100.150.1	151	10.100.150.2	10.5.200.1	10.99.1.55	555-1111	555-2222
Pod 6	160	10.100.160.1	161	10.100.160.2	10.6.200.1	10.99.1.56	666-1111	666-2222
Pod 7	170	10.100.170.1	171	10.100.170.2	10.7.200.1	10.99.1.57	777-1111	777-2222
Pod 8	180	10.100.180.1	181	10.100.180.2	10.8.200.1	10.99.1.58	888-1111	888-2222

Command List

This table lists the commands you will use in this exercise, in logical order. Refer to this list, or lists in previous exercises, if you need command assistance.

ROUTER CONFIGURATION COMMANDS		
Prompt	**Command**	**Description**
`Router(config)#`	**class-map match-any** *name*	Defines a class map with a specified *name* to match any contents.
`Router(config-cmap)#`	**match ip dscp** *level*	Specifies IP Precedence matching criteria for the class map.
`Router(config)#`	**dial-peer voice** *tag* **voip**	Enter voice over IP dial-peer configuration mode. The *tag* is an administrative number for tracking.
`Router(config-dial-peer)#`	**ip precedence** *level*	Apply an IP Precedence level to packets generated by a given dial peer.
`Router(config)#`	**policy-map** policy-name	Specifies the name of the service policy to configure.
`Router(config-pmap)#`	**class** class-name	Specifies the name of a class, predefined with the **class-map** command, included in the service policy.
`Router(config-pmap-c)#`	**bandwidth** *bandwidth*	Allows a specific amount of bandwidth, in Kilobits/second, to traffic that matches the class-map criteria.
`Router(config-pmap-c)#`	**priority** *bandwidth*	Places in the priority queue, and allows a specific amount of bandwidth in Kilobits/second, to traffic that matches the class-map criteria.
`Router(config-if)#`	**service-policy output** *name*	Applies a policy map to outbound traffic on an interface.

Setup

In a previous lab, you configured class maps called “video,” “ftp,” “http1” and “http2.” You then defined an “ingress” policy to assign DSCP values of “AF41,” “AF31,” “AF21,” and “AF22,” respectively, to these classes. If it isn’t still there, re-apply the “ingress” policy to the distribution router’s Fasterthernet 0/0 interface to inbound traffic from the servers (the “ingress” point) to “mark” the traffic appropriately.

Scenario

You are still the network administrator for CCI.

The Weighted Fair Queuing you configured has made you a hero and resulted in a massive bonus, but with your elevated “hero” status the users are expecting even more granular control over network resources.

The CEO feels that some web pages are more important than other web pages.

Your mission: Give video enough bandwidth for good quality viewing. Provide twice as much WAN bandwidth for the “important” web traffic than for the “not so” important traffic. Ensure timely forwarding of voice traffic.

Task 1: Configure Class Based Weighted Fair queuing (CBWFQ)

Step 1 Open the IPTV viewer and select a movie.

How would you rate the video quality? _____

Step 2 Using a web browser, open an http session to 10.99.1.15/qosdemo.htm. This web page, in turn, opens two additional web pages, “important” and “not_so.”

How is the video quality now? _____

Which one of the web pages load faster? _____

Step 3 Close your IPTV viewer and all web browsers.

Step 4 As you are already marking traffic with your “ingress policy, the next step is to define classes for and apply a congestion management service policy to traffic outbound on the serial 0/0 interface. Define four map classes named “af41,” “af31,” “af21,” and “af22.” For each of these classes, match the corresponding DSCP value.

Step 5 Define a policy map named “egress” which provides 600Kbps to video (af41), 30Kbps to FTP (af31), 20Kbps to the “important” HTTP (af21), and 10Kbps to the “not so” HTTP traffic.

Step 6 Apply your service policy to outbound traffic on the serial 0/0 port of your distribution router.

Step 7 As before, open a movie on the IPTV viewer and web browser sessions to the two web pages.

How is the video quality? _____

How do the web pages compare?_____

Task 2: Configure Low Latency Queuing (LLQ) to improve voice performance.

Step 1 Place a call from your analog phone to another pod and observe the quality.

Step 2 On your Access router, apply an IP precedence level of 5 to each VOIP dial peer configured on your router. The tag numbers can be identified with a "show dial-peer voice summary."

Step 3 On your Distribution router, define a class called "voice" and assign to it all traffic with an IP Precedence of 5.

Step 4 In the "egress" policy, place class "voice" traffic in the priority queue with 85 Kbps of bandwidth.

Step 5 Place a call from your analog phone to the other pod and again observe the quality.

Task 3: Clean up.

Step 6 Save your router configuration. We will be adding to it in future labs.

Completion Criteria:

You have successfully completed this laboratory exercise if you observed a noticeable improvement in video and voice performance, and superior performance for the "important" web traffic over the "not so" web traffic.

Configuration Examples (arranged by Task):

```
class-map match-any http1
  match protocol http url "*important*"
class-map match-any http2
  match protocol http url "*not_so*"
class-map match-any ftp
  match protocol ftp
policy-map ingress
  class http1
   set ip dscp 18
  class http2
   set ip dscp 20
```

```
  class ftp
   set ip dscp 26
  class video
   set ip dscp 34
!
class-map match-any af21
  match ip dscp af21
class-map match-any af22
  match ip dscp af22
class-map match-any af31
  match ip dscp af31
!
policy-map egress
  class af41
   bandwidth 600
  class af31
   bandwidth 30
  class af21
   bandwidth 20
!
interface serial 0/0
 service-policy out egress



dial-peer voice 10 voip
 ip precedence 5
dial-peer voice 20 voip
 ip precedence 5
```



```
dial-peer voice 30 voip
 ip precedence 5
dial-peer voice 40 voip
 ip precedence 5
dial-peer voice 50 voip
 ip precedence 5
dial-peer voice 60 voip
 ip precedence 5
dial-peer voice 70 voip
 ip precedence 5
dial-peer voice 80 voip
 ip precedence 5



class-map match-any voice
  match ip precedence 5
!
policy-map egress
  class voice
   priority 64
```

Lab Guide for Chapter 5: Congestion Avoidance

Overview

This document contains a laboratory exercise that identifies the effects of Weighted Red on TCP and UDP traffic.

Laboratory Exercise 5.1: WRED Operation

Complete the following laboratory exercise to practice what you have learned so far in this chapter.

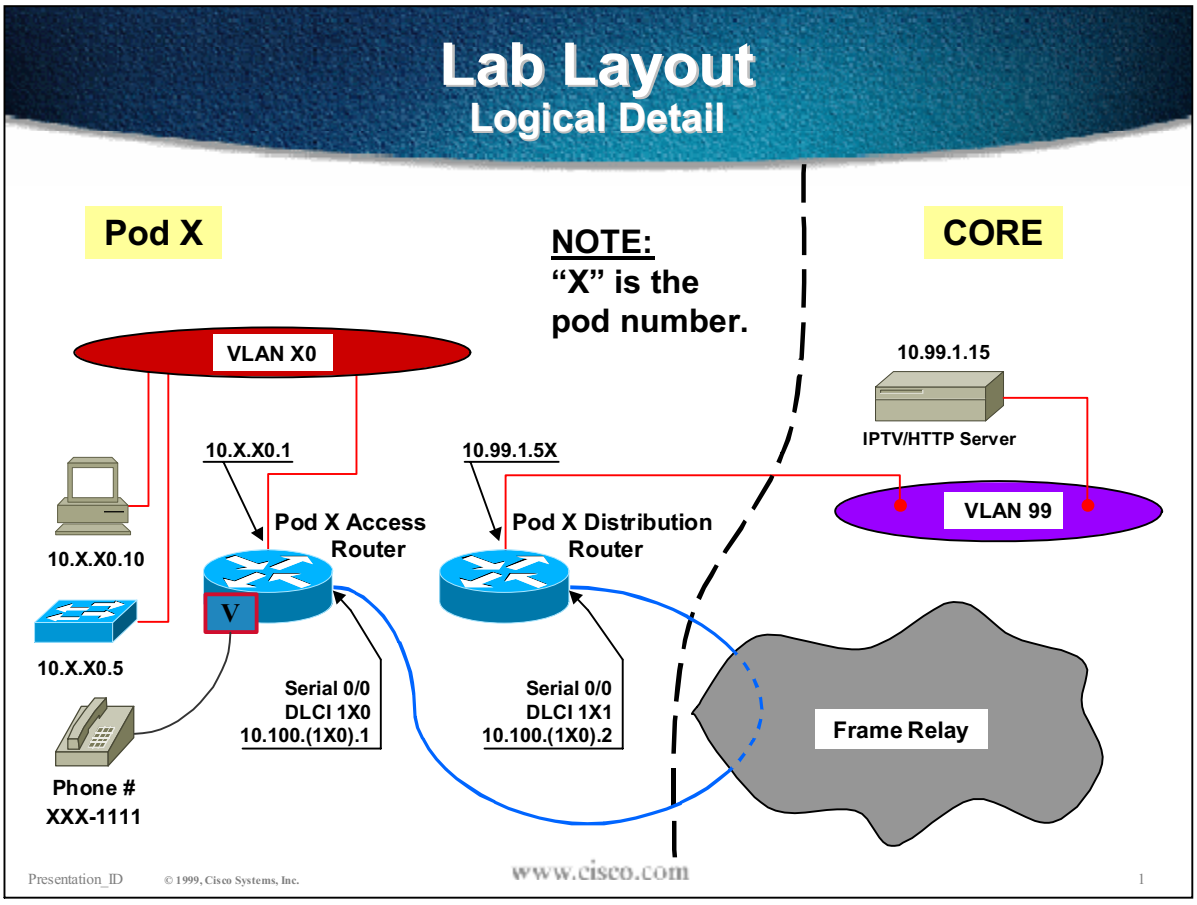
Objectives

In this lab you will complete the following task:

- Identify the effects of Weighted Red on TCP and UDP traffic.

Visual Objective

The figure below illustrates the configuration you will be working with in this exercise.



WAN Connections

In this exercise, you will be connecting to the Core over your WAN links, and applying QoS to that serial connection. The following table provides the specifics of the WAN connections.

	WAN Link				Miscellaneous			
	Access Router		Distribution Router		Access Router Loop 0 Address	Distribution Router Ethernet Address	Analog Phone#	IP Phone#
	DLCI	IP	DLCI	IP				
Pod 1	110	10.100.110.1	111	10.100.110.2	10.1.200.1	10.99.1.51	111-1111	111-2222
Pod 2	120	10.100.120.1	121	10.100.120.2	10.2.200.1	10.99.1.52	222-1111	222-2222
Pod 3	130	10.100.130.1	131	10.100.130.2	10.3.200.1	10.99.1.53	333-1111	333-2222
Pod 4	140	10.100.140.1	141	10.100.140.2	10.4.200.1	10.99.1.54	444-1111	444-2222
Pod 5	150	10.100.150.1	151	10.100.150.2	10.5.200.1	10.99.1.55	555-1111	555-2222
Pod 6	160	10.100.160.1	161	10.100.160.2	10.6.200.1	10.99.1.56	666-1111	666-2222
Pod 7	170	10.100.170.1	171	10.100.170.2	10.7.200.1	10.99.1.57	777-1111	777-2222
Pod 8	180	10.100.180.1	181	10.100.180.2	10.8.200.1	10.99.1.58	888-1111	888-2222

Command List

This table lists the commands you will use in this exercise, in logical order. Refer to this list, or lists in previous exercises if you need command assistance.

ROUTER CONFIGURATION COMMANDS		
Prompt	**Command**	**Description**
`Router(config)#`	**`policy-map`** `policy-name`	Specifies the name of the service policy to configure.
`Router(config-pmap)#`	**`class`** `class-name`	Specifies the name of a class, predefined with the **class-map** command, included in the service policy.
`Router(config-pmap-c)#`	**`randon-detect dscp-based`**	Applies WRED to a class of traffic.
`Router(config-pmap-c)#`	**`priority`** *bandwidth*	Places in the priority queue, and allows a specific amount of bandwidth in Kilobits/second, to traffic that match the class-map criteria.
`Router(config-if)#`	**`service-policy output`** *name*	Applies a policy map to outbound traffic on an interface.
`Router#`	**`Show policy-map interface`**	Displays configured classes on a given interface and counters reflect the action taken by WRED

Setup

You will be using the same topology and adding to the “egress” policy defined in the CBWFQ/LLQ lab.

Scenario

You are still the network administrator for CCI.

Numerous studies have indicated improved performance with RED, so you have decided to implement it in your own network.

Task 1: Configure Class Based Weighted RED (Random Early Detection)

Step 1 Telnet to your distribution router. In the "egress" policy map, apply DSCP based WRED to the "af21," "af22," and "af41" classes. (Make sure the policy is still applied to the serial interface.)

Step 2 Open up the IPTV viewer and enjoy a movie, and, using a web browser open up the "qosdemo.htm" page on 10.99.1.15.

How is the video quality? _____

Step 3 From privileged user mode on the core router, show the service policy applied to the serial port and observe the number of drops associated with each class of traffic. Looking at the "minimum threshold" value for each DSCP level, is AF21 traffic more likely to be dropped than AF41?

Step 4 Judging from the video performance, is it a good idea to apply WRED to UDP traffic?

Task 2: Clean up.

Step 1 Remove the WRED from the policy maps. We will keep the service policies themselves for future labs.

Step 2 Save your router configuration.

Completion Criteria:

You have successfully completed this laboratory exercise if you observed the dropped video traffic.

Configuration Example:

```
policy-map podX
  class af41
   bandwidth 600
   random-detect dscp-based
  class af31
   bandwidth 30
```

```
 random-detect dscp-based
class af21
 bandwidth 20
 random-detect dscp-based
class af22
 bandwidth 10
 random-detect dscp-based
```


Lab Guide for Chapter 6: Link Efficiency

Overview

This document contains two laboratory exercises that implement what you have learned about link efficiency.

The first lab, “Configure Frame Relay Fragmentation,” has you configure frame relay fragmentation and observe the reduction in voice quality.

The second lab, “Configuring RTP Header Compression,” has you configure RTP header compression and observe the reduction in voice bandwidth requirements.

Laboratory Exercise 6.1: Configuring Frame Relay Fragmentation

Complete the following laboratory exercise to practice what you have learned so far in this chapter.

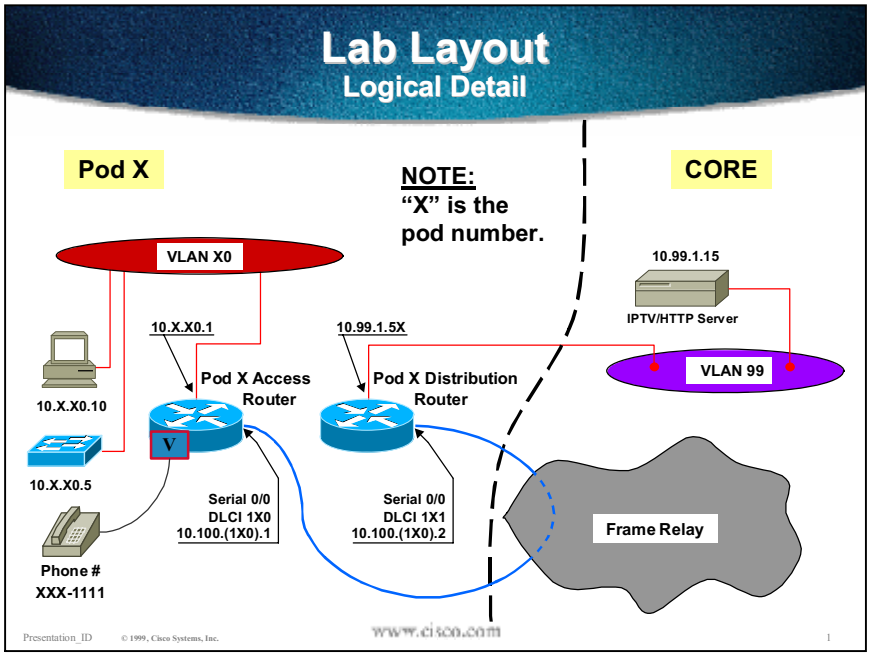
Objectives

In this lab you will complete the following task:

- Configure Frame Relay Fragmentation and observe the reduction in voice delay.

Visual Objective

The following figure illustrates the configuration you will be working with in this exercise.



WAN Connections

In this exercise, you will be connecting to the Core over your WAN links, and applying QoS to that serial connection. The following table provides the specifics of the WAN connections.

	WAN Link				Miscellaneous			
	Access Router		Distribution Router		Access Router Loop 0 Address	Distribution Router Ethernet Address	Analog Phone#	IP Phone#
	DLCI	IP	DLCI	IP				
Pod 1	110	10.100.110.1	111	10.100.110.2	10.1.200.1	10.99.1.51	111-1111	111-2222
Pod 2	120	10.100.120.1	121	10.100.120.2	10.2.200.1	10.99.1.52	222-1111	222-2222
Pod 3	130	10.100.130.1	131	10.100.130.2	10.3.200.1	10.99.1.53	333-1111	333-2222
Pod 4	140	10.100.140.1	141	10.100.140.2	10.4.200.1	10.99.1.54	444-1111	444-2222
Pod 5	150	10.100.150.1	151	10.100.150.2	10.5.200.1	10.99.1.55	555-1111	555-2222
Pod 6	160	10.100.160.1	161	10.100.160.2	10.6.200.1	10.99.1.56	666-1111	666-2222
Pod 7	170	10.100.170.1	171	10.100.170.2	10.7.200.1	10.99.1.57	777-1111	777-2222
Pod 8	180	10.100.180.1	181	10.100.180.2	10.8.200.1	10.99.1.58	888-1111	888-2222

Command List

This table lists the commands you will use in this exercise, in logical order. Refer to this list, lists in previous exercises, or the example configuration at the end of the exercise if you need further command assistance.

ROUTER COMMANDS		
Prompt	**Command**	**Description**
`router(config)#`	**`map-class frame-relay`** *`string`*	Defines a frame relay map class called *string*.
`router(config-map-class)#`	**`frame-relay fair-queue`**	Enables WFQ for any PVC the map-class is attached to.
`router(config-map-class)#`	**`frame-relay ip rtp priority`** *`start-port range bw`*	Enables RTP priority, where *start-port* is the bottom of the UDP port range, *range* is the number of ports in the range, and *bw* is the allocated bandwidth.
`router(config-map-class)#`	**`frame-relay fragment`** *`size`*	Enables frame relay fragmentation, where size is the size of the fragments.
`router(config-if)#`	**`frame-relay traffic-shaping`**	Enables frame relay traffic shaping.
`router(config-fr-dlci)#`	**`class`** *`string`*	Attaches the map-class called *string* to the DLCI.
`router#`	**`show frame-relay PVC`** *`dlci`*	Verifies FRTS and fragmentation.

Setup

Your instructor has reduced the clock rate on the frame relay switch to 64Kbps. To see how voice reacts at this clock rate when mixed with large data frames, you will be sending 1500 byte pings from your PC to another PC. This will not cause congestion, as the pings from both laptops combined with a voice call will only require about 50Kbps of bandwidth, but the serialization delay for these large frames will be roughly 200ms.

Scenario

The customer is complaining because even though the WAN links are not congested, voice quality is poor because of delay.

Task 1: Configure frame relay fragmentation to reduce delay and jitter for voice.

Note	Pods 1, 3, 5, and 7 will be working with pods 2, 4, 6, and 8 respectively. This exercise will have to be closely coordinated with the other pod.

Step 1 From a DOS prompt, send a continuous stream of 1500 byte pings to the other pod's PC. This can be done with the "ping <destination address> –t –l 1500 –w 5000" command.

Step 2 Place an analog phone call to the pod you are working with.

How is the call quality?_____

Step 3 In order to configure FRF.12, we must define a frame relay map class and apply that to the PVC. Define a frame relay map-class called "voice," and in that class enable WFQ, fragmentation with a packet size of 200, IP RTP priority for the 16383 UDP ports after and including 16384, and an allowable bandwidth of 85 Kbps.

Step 4 On the serial 0/0 interfaces of both your distribution and access routers, remove any service policies that are currently applied and enable frame relay traffic shaping (required for Frame Relay Fragmentation.)

Step 5 Apply the "voice" class to the DLCI's (PVC number) on both your distribution and access routers.

Note	Frame Relay Fragmentation MUST be applied to both ends of the connection for the circuit to work. Therefore, apply it first to your distribution router, then your access router. If you apply it first to the access router, connection to the distribution router will be lost and you will not be able to apply it there.

Step 6 Confirm your configuration with the "show frame-relay pvc" command.

Step 7 Once both pods have completed these steps, place another phone call to the other pod.

How noticeable is delay now? _____

Task 2: Clean up.

Step 1 Do not save your router configuration. Stop the pings.

Completion Criteria:

You have successfully completed this laboratory exercise if you observed the reduced delay resulting from Frame Relay Fragmentation.

Configuration Examples (arranged by Task):

```
map-class frame-relay voice
  frame-relay fair-queue
  frame-relay fragment 200
```

(Enables FRF.12 fragmentation with a fragment size of 200 bytes)

```
  frame-relay ip rtp priority 16384 16383 85
```

(Assigns a priority queue for voice (based on UDP ports) with max bandwidth = 85 Kbps.

```
interface Serial1/0
  frame-relay traffic-shaping
interface Serial1/0.120 point-to-point
   frame-relay interface-dlci 120
    class voice
```

(Attaches the "voice" class to the PVC)

Laboratory Exercise 6.2: Configuring RTP Header Compression

Complete the following laboratory exercise to practice what you learned so far in this chapter.

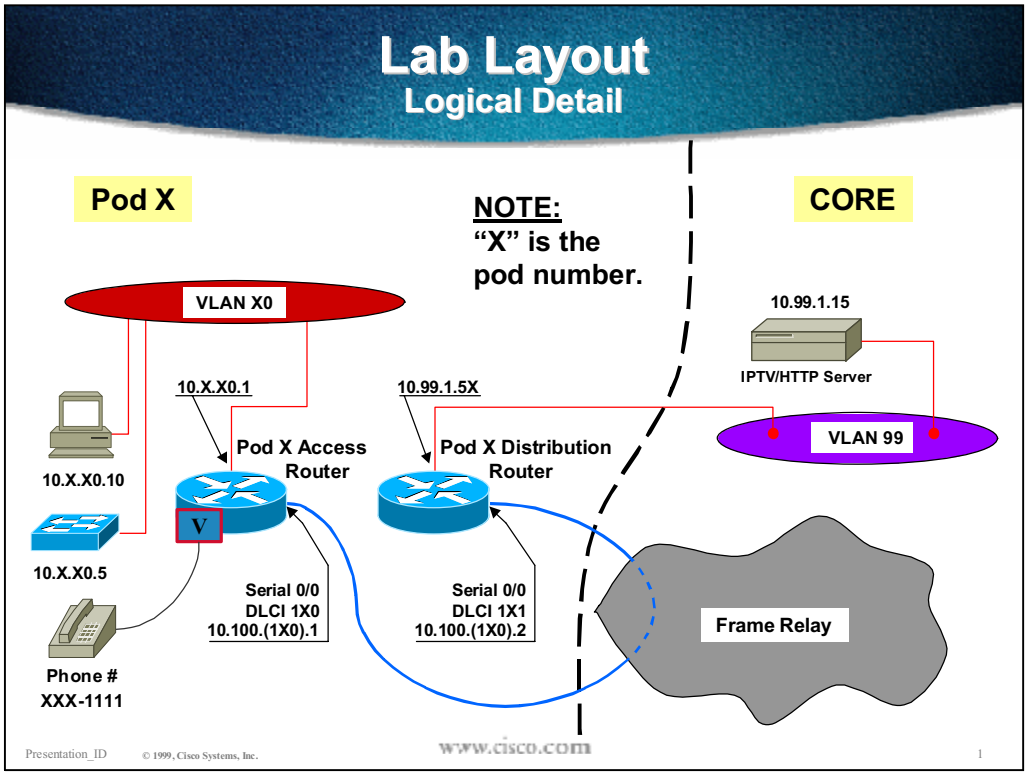
Objectives

In this lab you will complete the following task:

- Configure RTP header compression and observe the reduction in voice bandwidth requirements.

Visual Objective

The following figure illustrates the configuration you will be working with in this exercise.



WAN Connections

In this exercise, you will be connecting to the Core over your WAN links, and applying QoS to that serial connection. The following table provides the specifics of the WAN connections.

	WAN Link				Miscellaneous			
	Access Router		Distribution Router		Access Router Loop 0 Address	Distribution Router Ethernet Address	Analog Phone#	IP Phone#
	DLCI	IP	DLCI	IP				
Pod 1	110	10.100.110.1	111	10.100.110.2	10.1.200.1	10.99.1.51	111-1111	111-2222
Pod 2	120	10.100.120.1	121	10.100.120.2	10.2.200.1	10.99.1.52	222-1111	222-2222
Pod 3	130	10.100.130.1	131	10.100.130.2	10.3.200.1	10.99.1.53	333-1111	333-2222
Pod 4	140	10.100.140.1	141	10.100.140.2	10.4.200.1	10.99.1.54	444-1111	444-2222
Pod 5	150	10.100.150.1	151	10.100.150.2	10.5.200.1	10.99.1.55	555-1111	555-2222
Pod 6	160	10.100.160.1	161	10.100.160.2	10.6.200.1	10.99.1.56	666-1111	666-2222
Pod 7	170	10.100.170.1	171	10.100.170.2	10.7.200.1	10.99.1.57	777-1111	777-2222
Pod 8	180	10.100.180.1	181	10.100.180.2	10.8.200.1	10.99.1.58	888-1111	888-2222

Command List

The following table lists the commands you will use in this exercise, in logical order. Refer to this list, or lists in previous exercises, if you need command assistance.

ROUTER COMMANDS		
Prompt	**Command**	**Description**
`Router(config-if)#`	**`load-interval`** *`interval`*	Specifies the interval, in seconds, used to gather interface statistics.
`Router(config-dial-peer)#`	**`no vad`**	Disables Voice Activity Detection. VAD saves bandwidth by not sending data during periods of silence on the call.
`Router(config-subif)#`	**`frame-relay ip rtp header-compression`**	Enable RTP header compression on a frame relay interface.
`Router#`	**`sh ip rtp header-compression`**	Displays cRTP statistics.

Setup

No advanced configuration is needed for this exercise.

Scenario

CCI is relying more and more on voice over IP to reduce their long-distance phone bills. The "powers-that-be" have offered you a lifetime supply of mulch if you can reduce the load these calls place on the network.

Task 1: Configure RTP header compression (cRTP) to reduce voice bandwidth requirements.

Note	Pods 1, 3, 5, and 7 will be working with pods 2, 4, 6, and 8 respectively. This exercise needs to be closely coordinated with the other pod.

Step 1 Change the load interval on the serial 0/0 interface on both your access and distribution routers to 30 seconds. (The default is 300 seconds, which is longer than we'd like to wait to see results.)

Step 2 On your access router, disable VAD on the voice over IP dial peer pointing to the other pod. This tells the dial peer to send a constant stream of data whether anyone is actually talking or not. Also set the CODEC to G.729r8 for this dial peer.

Note	Reducing the interface load interval increases processor utilization. Disabling VAD increases bandwidth utilization. Neither of these commands is recommended in "real-life," and each is used here for demonstration purposes only.

Step 3 Place a voice call to the other pod, and note the utilization of the serial link after 30 seconds.

Utilization: _____bits/second

Step 4 Enable RTP header compression on both the access and distribution end of your serial link.

Step 5 Again, place a call to the other pod and note the utilization on the serial link after 30 seconds.

Utilization: _____bits/second

How much bandwidth has cRTP saved? _____

Task 2: Clean up.

Step 1 Set the load interval on serial 0/0 back to 300 seconds, remove the rtp header compression, and re-enable VAD on your access and distribution routers.

Step 2 Save your router configuration.

Completion Criteria:

You have successfully completed this laboratory exercise if you observed the bandwidth savings resulting from cRTP.

Lab Guide for Chapter 7: Policing and Shaping

Overview

This chapter contains two laboratory exercises that implement your knowledge of Policing and Shaping.

The first lab, “Configuring Class Based Policing,” has you configure class-based policing to limit specific traffic.

The second lab, “Configuring Frame Relay Traffic Shaping,” has you configure and identify the effects of FRTS.

Laboratory Exercise 7.1: Configuring Class Based Policing

Complete the following laboratory exercise to practice what you have learned so far in this chapter.

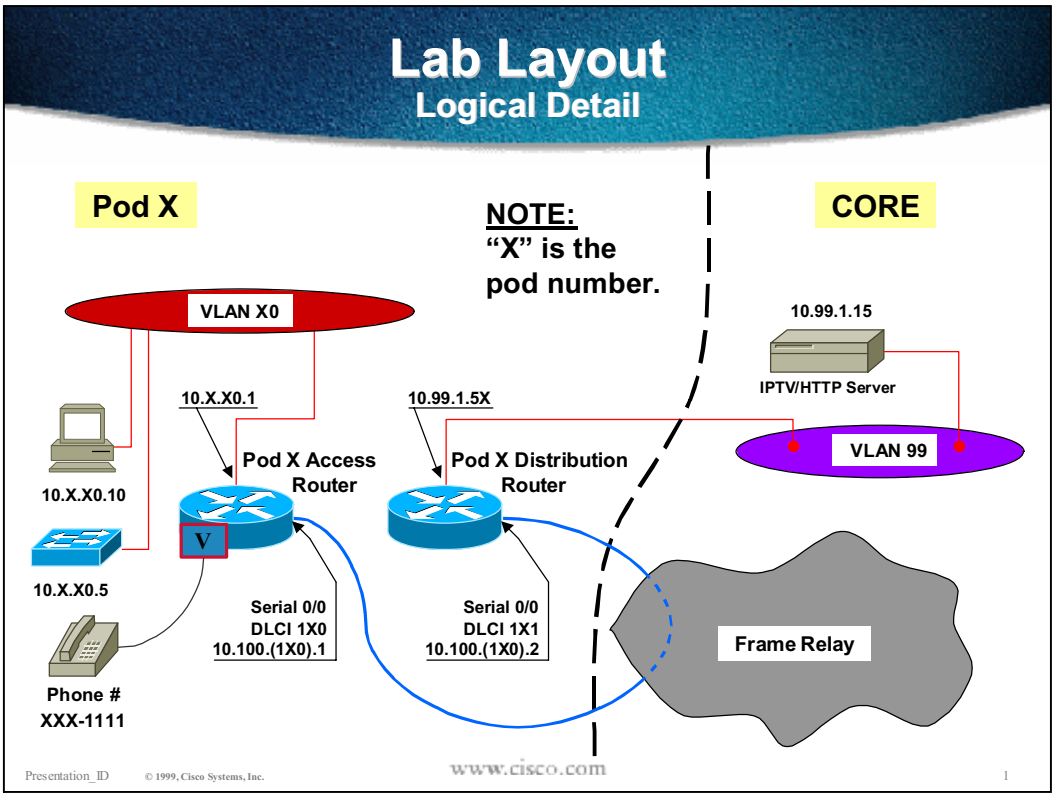
Objectives

In this lab you will complete the following task:

- Configure class-based policing to limit specific traffic.

Visual Objective

The following figure shows the configuration you will be working with in this exercise.



WAN Connections

In this exercise, you will be connecting to the Core over your WAN links, and applying QoS to that serial connection. The following table provides the specifics of the WAN connections.

	WAN Link				Miscellaneous			
	Access Router		Distribution Router		Access Router Loop 0 Address	Distribution Router Ethernet Address	Analog Phone#	IP Phone#
	DLCI	IP	DLCI	IP				
Pod 1	110	10.100.110.1	111	10.100.110.2	10.1.200.1	10.99.1.51	111-1111	111-2222
Pod 2	120	10.100.120.1	121	10.100.120.2	10.2.200.1	10.99.1.52	222-1111	222-2222
Pod 3	130	10.100.130.1	131	10.100.130.2	10.3.200.1	10.99.1.53	333-1111	333-2222
Pod 4	140	10.100.140.1	141	10.100.140.2	10.4.200.1	10.99.1.54	444-1111	444-2222
Pod 5	150	10.100.150.1	151	10.100.150.2	10.5.200.1	10.99.1.55	555-1111	555-2222
Pod 6	160	10.100.160.1	161	10.100.160.2	10.6.200.1	10.99.1.56	666-1111	666-2222
Pod 7	170	10.100.170.1	171	10.100.170.2	10.7.200.1	10.99.1.57	777-1111	777-2222
Pod 8	180	10.100.180.1	181	10.100.180.2	10.8.200.1	10.99.1.58	888-1111	888-2222

Command List

This table lists the commands you will use in this exercise in logical order. Refer to this list, lists in previous exercises, or the sample configuration at the end of the lab if you need command assistance.

ROUTER COMMANDS		
Prompt	**Command**	**Description**
`Router(config-if)#`	**`service-policy output`** *`name`*	Applies a policy map to outbound traffic on an interface.
`Router(config)#`	**`policy-map`** `policy-name`	Specifies the name of the service policy to configure.
`Router(config-pmap)#`	**`class`** `class-name`	Specifies the name of a class, predefined with the **class-map** command, included in the service policy.
`Router(config-pmap-c)#`	**`police`** *`average-rate burst-rate excess-burst-rate`* **`conform-action`** *`action`* **`exceed-action`** *`drop`*	Defines policing parameters and actions to be taking for a given class of traffic.

Setup

In the CBWFQ/LLQ lab, you defined class maps and a service policy on the distribution router for traffic coming toward your access router. In this exercise, you will add policing to the "not so" important HTTP traffic.

Scenario

In the CBWFQ/LLQ exercise, you allowed 600Kbps of bandwidth for AF41 (video,) 20Kbps for AF21 (“important” HTTP traffic,) and 12Kbps for AF22 (“not so” HTTP traffic). Without video, the “important” traffic will be allowed roughly two-thirds of the available bandwidth, and the “not so” will get the other third. In this exercise, you will limit the “not so” traffic to no more than 12Kbps of bandwidth regardless of how much bandwidth is available, so that all of the unused bandwidth will go to the “important” traffic.

Task 1: Configure Traffic Policing.

Step 1 Telnet to you distribution router and apply your “egress” service policy to outbound traffic on the serial port coming to you pod (You may have to remove FRTS first.)

Step 2 To confirm its operation, open a movie on the IPTV viewer and the “qosdemo” web page on your browser. As before, you should observer that the web pages open rather slowly, but the “important” page loads roughly twice as fast as the “not so” page.

Step 3 Close the IPTV viewer and the web browser. Wait about thirty seconds for the IPTV multicast traffic to clear, and again open the “qosdemo” web page on your browser. You should observe that both frames load much faster, yet the “important” page still loads roughly twice as fast as the “not so” page.

Step 4 Apply the “police” command to the “af22” class in the “egress” policy map to limit bandwidth to 12Kbps with a 12Kbps burst.

Step 5 Close the web browser, and again open the “qosdemo” web page. You should observe that “important” frame opens even faster than before, but the “not so” frame opens at about the same speed it did while you were watching a movie. This illustrates that the “not so” web page is being limited so that the “important” page gets all of the surplus bandwidth on the serial 0/0 interface.

Step 6 Use the ‘show policy-map interface <interface>’ command to confirm the actions of WRED.

Task 2: Clean up.

Step 1 Remove the “egress” policy from the distribution router interface.

Completion Criteria:

You have successfully completed this laboratory exercise if you observed the policing on the “not so” HTTP traffic in action.

Configuration Example:

```
policy-map podx

  class af22

     police 12000 12000 12000 conform-action transmit exceed-action drop
```



```
.
.
.
interface Serial0/0
 load-interval 30
 service-policy output podx
```

Laboratory Exercise 7.2: Configuring Frame Relay Traffic Shaping

Complete the following laboratory exercise to practice what you have learned so far in this chapter.

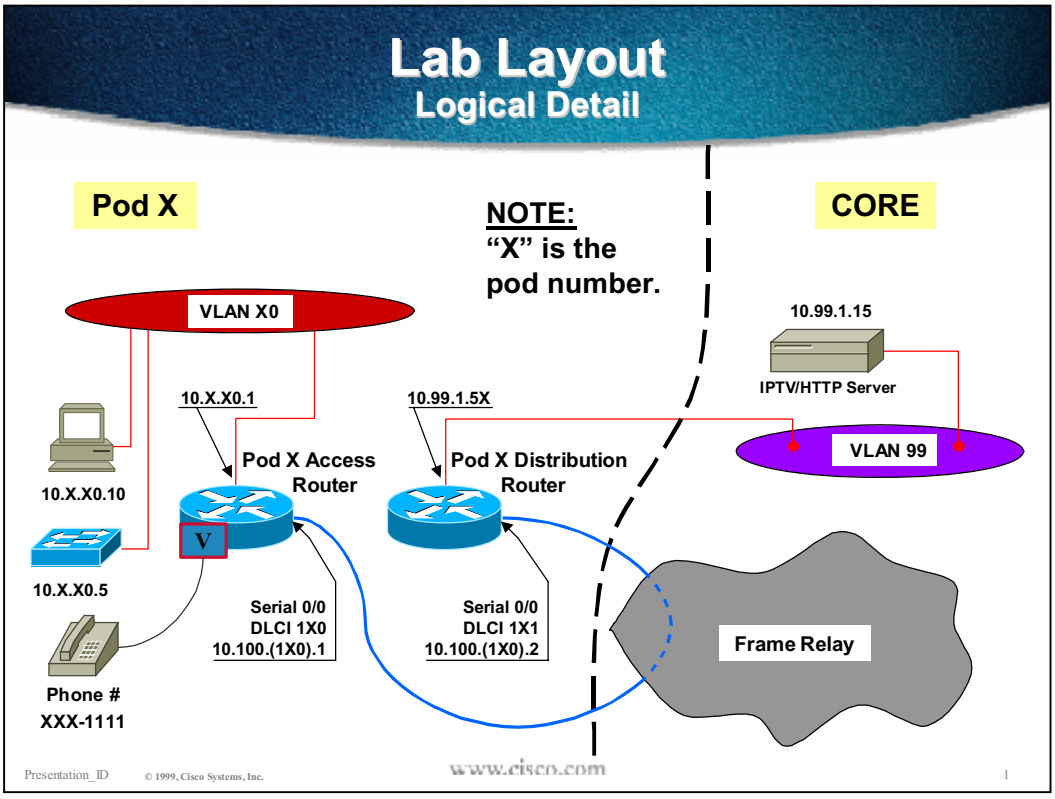
Objectives

In this lab you will complete the following task:

- Configure and identify the effects of FRTS.

Visual Objective

The figure below illustrates the configuration you will be working with in this exercise.



WAN Connections

In this exercise, you will be connecting to the Core over your WAN links, and applying QoS to that serial connection. The following table provides the specifics of the WAN connections.

	WAN Link				Miscellaneous			
	Access Router		Distribution Router		Access Router Loop 0 Address	Distribution Router Ethernet Address	Analog Phone#	IP Phone#
	DLCI	IP	DLCI	IP				
Pod 1	110	10.100.110.1	111	10.100.110.2	10.1.200.1	10.99.1.51	111-1111	111-2222
Pod 2	120	10.100.120.1	121	10.100.120.2	10.2.200.1	10.99.1.52	222-1111	222-2222
Pod 3	130	10.100.130.1	131	10.100.130.2	10.3.200.1	10.99.1.53	333-1111	333-2222
Pod 4	140	10.100.140.1	141	10.100.140.2	10.4.200.1	10.99.1.54	444-1111	444-2222
Pod 5	150	10.100.150.1	151	10.100.150.2	10.5.200.1	10.99.1.55	555-1111	555-2222
Pod 6	160	10.100.160.1	161	10.100.160.2	10.6.200.1	10.99.1.56	666-1111	666-2222
Pod 7	170	10.100.170.1	171	10.100.170.2	10.7.200.1	10.99.1.57	777-1111	777-2222
Pod 8	180	10.100.180.1	181	10.100.180.2	10.8.200.1	10.99.1.58	888-1111	888-2222

Command List

This table lists the commands you will use in this exercise, in logical order. Refer to this list, lists in previous exercises, or the sample configuration at the end of the lab if you need command assistance.

ROUTER COMMANDS		
Prompt	**Command**	**Description**
`router(config-if)#`	**`frame-relay traffic-shaping`**	Enables frame relay traffic shaping.
`router(config-if)#`	**`bandwidth`** *`Kbps`*	Specifies the throughput of and interface or PVC in Kbps.
`router(config-if)#`	**`load-interval`** *`seconds`*	Specifies the number of seconds over which the router computer throughput and other values.
`Router#`	**`show interface`** *`interface`*	Specifies the name of the service policy to configure.
`Router#`	**`| include`** *`string`*	Placed after a "show" command, tells the router to only include lines of the output which include *string*.
`Router#`	**`Show policy-map interface`**	Displays configured classes on a given interface and counters reflect the action taken by shaper

Setup

Refer to the "Lab Layout: Logical Detail" diagram for the topology you will be working with in this lab.

Scenario

In a frame relay connection, the physical speed of the connection to the provider is often much faster than the CIR of the individual PVCs. It is generally in your best interest to limit your throughput to the CIR, though this is not the default behavior on the routers. For this lab, we will assume that the physical speed of the interface is 730K and the CIR is 256K.

Task 1: Configure Traffic Shaping.

Step 1 Open up the IPTV viewer to generate traffic.

Step 2 Telnet to your distribution router and remove your "egress" service policy, or existing frame relay traffic shaping from the outbound traffic on the serial port coming to your pod.

Step 3 If not already done, set the load interval on the interface to 30 seconds so you won't have to wait too long to see results in changes in throughput.

Step 4 Open the "qosdemo.htm" web page on your browser. After approximately 30 seconds, issue the "show interface" command for your interface on the distribution router, and observe the 30 second output rate in bits per second. Alternately, do a "show interface (interface number) | include output rate" to just see the specific information you're looking for.

What is the output rate? _____

Step 5 On your distribution router serial port, change the bandwidth to 256 Kbps (this is normally set to match your CIR).

Step 6 Close your web browser and re-open the qos-demo page.

Is there any change in the speed of the web page loading? _____

The video stream requires roughly 600Kbps. How does it look now? _____

What is the 30 second output rate of the interface now? _____

Step 7 Despite the fact that you reduced the "bandwidth" to 256Kbps, you should see no change because the router, by default, will pass traffic as fast as it physically can. The "bandwidth" command is only used for metrics by some routing protocols, accounting, and QoS. The clock rate on the interface is 730Kbps. Now enable class-based shaping by creating a policy called 'egress-shaping' and apply it outbound to the serial interface on the distribution router. Define the policy so your traffic is shaped to the following rates:

Video (af41):	128Kbps
FTP (af31):	32Kbps
HTTP1 (af21):	64Kbps
HTTP2 (af22):	32Kbps

Step 8 Close and re-open the web page. Use the 'show policy-map interface' command to confirm that the traffic is being shaped to the specified rates.

Task 2: Clean up.

Step 1 Remove the shaping service policy from your Distribution router serial interface, and set the bandwidth back to 900Kbps.

Completion Criteria:

You have successfully completed this laboratory exercise if you observed the action of class-based shaping.

Configuration Example:

```
!
policy-map egress-shaping
  class af41
   shape average 128000
  class af31
   shape average 32000
  class af21
   shape average 64000
  class af22
   shape average 32000
!
interface serial 0/0
 service-policy out egress-shaping
```

Lab Guide for Chapter 8: Call Admission Control

Overview

This chapter contains two laboratory exercises that implement your knowledge of Call Admission Control.

The first lab, “Configuring Local Voice BusyOut and H.323 Gatekeeper” implements LVBO and CAC for H.323 Gatekeeper on your lab configuration.

The second lab, “RSVP,” implements CAC using RSVP.

Laboratory Exercise 8.1: Configuring Local Voice BusyOut and H.323 Gatekeeper

Complete the following laboratory exercise to practice what you have learned so far in this chapter.

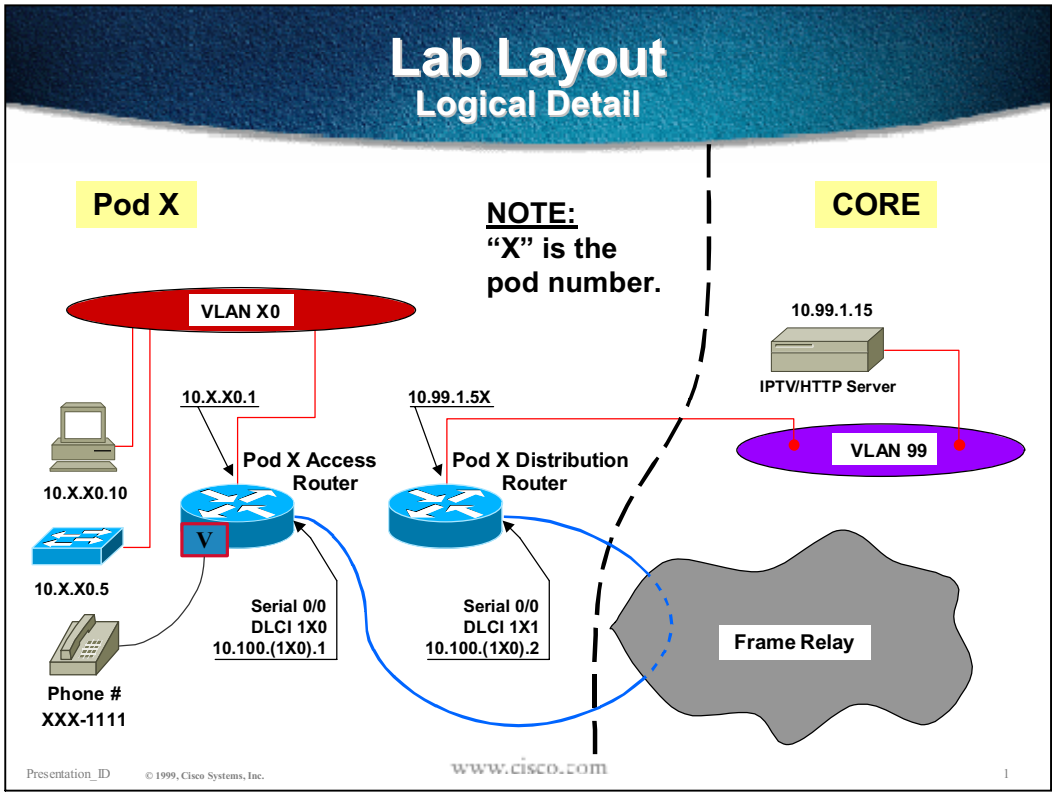
Objectives

In this lab you will complete the following tasks:

- Configure Local Voice BusyOut.
- Configure CAC with H.323 Gatekeeper.

Visual Objective

The figure below illustrates the configuration you will be working with in this exercise.



WAN Connections

In this exercise, you will be connecting to the Core over your WAN links, and applying QoS to that serial connection. The following table provides the specifics of the WAN connections.

	WAN Link				Miscellaneous			
	Access Router		Distribution Router		Access Router Loop 0 Address	Distribution Router Ethernet Address	Analog Phone#	IP Phone#
	DLCI	IP	DLCI	IP				
Pod 1	110	10.100.110.1	111	10.100.110.2	10.1.200.1	10.99.1.51	111-1111	111-2222
Pod 2	120	10.100.120.1	121	10.100.120.2	10.2.200.1	10.99.1.52	222-1111	222-2222
Pod 3	130	10.100.130.1	131	10.100.130.2	10.3.200.1	10.99.1.53	333-1111	333-2222
Pod 4	140	10.100.140.1	141	10.100.140.2	10.4.200.1	10.99.1.54	444-1111	444-2222
Pod 5	150	10.100.150.1	151	10.100.150.2	10.5.200.1	10.99.1.55	555-1111	555-2222
Pod 6	160	10.100.160.1	161	10.100.160.2	10.6.200.1	10.99.1.56	666-1111	666-2222
Pod 7	170	10.100.170.1	171	10.100.170.2	10.7.200.1	10.99.1.57	777-1111	777-2222
Pod 8	180	10.100.180.1	181	10.100.180.2	10.8.200.1	10.99.1.58	888-1111	888-2222

Command List

This table lists the commands you will use in this exercise, in logical order. Refer to this list, lists in previous exercises, or the sample configuration at the end of the lab, if you need command assistance.

ROUTER COMMANDS		
Prompt	**Command**	**Description**
`Router(config)#`	**`voice-port`** *`port`*	Gets the voice port configuration mode.
`p2r(config-voiceport)#`	**`busyout monitor serial`** *`interface`*	Monitors a specific serial interface, and provide a busy signal to the phone if that interface goes down..
`Router(config-if)#`	**`h323-gateway voip interface`**	Defines the interface whose IP address will be used for the gatekeeper function.
`Router(config-dial-peer)`	**`session target ras`**	Enables the dial peer to use H.323 RAS (Registration, Admission, and Status) signaling to determine the IP address of the session target.
`Router(config)#`	**`gatekeeper`**	Defines the router as an h323 gatekeeper and gets to gatekeeper configuration mode.
`Router(config-gk)#`	**`zone local`** *`zone-name domain-name`*	Defines the local zone and domain name.
`Router(config-gk)#`	**`zone remote`** *`zone-name domain-name ip-address`*	Defines parameters for a remote zone .
`Router(config-gk)#`	**`zone prefix`** *`name number`*	Specifies the prefix for numbers reachable at the remote zone.
`Router(config-gk)#`	**`bandwidth remote`** *`kbps`*	Specifies the amount of available bandwidth for calls to remote zones.
`Router(config-gk)#`	**`no shut`**	Enables the gatekeeper function
`Router(config)#`	**`gateway`**	

ROUTER COMMANDS		
Prompt	**Command**	**Description**
Router#	**debug RAS**	Shows the RAS interactions as they occur.
Router#	**show gatekeeper endpoints**	Verifies the gatekeeper/gateway configuration.
Router#	**show voice port summary**	Provides a brief list of all the configured voice ports.

Setup

The "Lab Layout: Logical Detail" diagram illustrates the configuration that you will be working with in this exercise. You will be working with one other pod in this exercise. It is suggested that you pair together pods 1 and 2, pods 3 and 4, pods 5 and 6, and pods 7 and 8.

Scenario

You have been asked to configure your phones so that a busy signal is heard if no connection to the internet is available, and to configure so that too many simultaneous voice calls can't be made for the available resources.

Task 1: Configure LVBO

Note Do not save your configurations in this lab.

Step 2 Verify that a dial tone is available on the handset in your pod. If no dial tone is present, do not proceed until you have fixed the problem.

Step 3 Implement the local voice busyout feature on the first available FXS voice port on your router. Identity the available voice ports with the "show voice port summary" command.

Step 4 Disable (shutdown) the serial interface that is being monitored, then lift your telephone.

What do you hear? _____

Task 2: Configure Call Admission Control with the H.323 Gatekeeper functionality

Step 5 Define the interface that will be used as the H.323 gateway interface (there can be only one). The loopback interface is recommended because it never goes down.

Step 6 Enable H.323 RAS (Registration, Admission, and Status) or your VOIP dial peer. When there is a match on the specified dial string, the gatekeeper will be consulted to translate the dialed E.164 (phone number) to the destination IP address.

Step 7 Configure the MCM Gatekeeper; The local zone will be podX, where X is your pod number, and the domain name will be cisco.com. The remote zone will be named podY where Y is the other pod's number, the domain will be cisco.com, and the ip

address will be their loopback address. The prefix for the remote zone will be YYY*, where Y is the other pod's number. Specify 64Kbps of bandwidth, and do a "no shut."

Step 8 Enable the voice gateway to register with the gatekeeper.

Step 9 Confirm the gatekeeper configuration.

Step 10 Test the configuration by placing a call to the remote pod that you have configured. The call should proceed normally.

Step 11 To demonstrate CAC, change the available bandwidth for remote calls to 63Kbps and again try to place a call. (Shouldn't work.)

Completion Criteria:

You have successfully completed this laboratory exercise if you observed the busy signal after shutting down the serial interface, and again after reducing the remote bandwidth to 63Kbps.

Configuration Example:

```
voice-port 1/0/0
 busyout monitor serial 0/0


serial 0/0
 shutdown




interface Loopback0
 h323-gateway voip id pod1 ipaddr 10.1.200.1


dial-peer voice 20 voip
 destination-pattern 222....
 session target ras


gatekeeper
 zone local pod1 cisco.com
```



```
 zone remote pod2 cisco.com 10.2.200.2
 zone prefix pod2 222*
 bandwidth remote 64
 no shutdown

gateway
```

Reference:

RAS messages initiated by your gateway and gatekeeper begin with the string "RASLib::". Note the source and destination addresses to help decipher the message flow. The following is a partial list of RAS messages for your reference:

```
ARQ             Admission Request
ARJ             Admission Rejection
LRQ             Locate Request
LCF             Locate Confirmation
RIP             Request In Progress
```

Laboratory Exercise 8.2: RSVP

Complete the following laboratory exercise to practice what you have learned so far in this chapter.

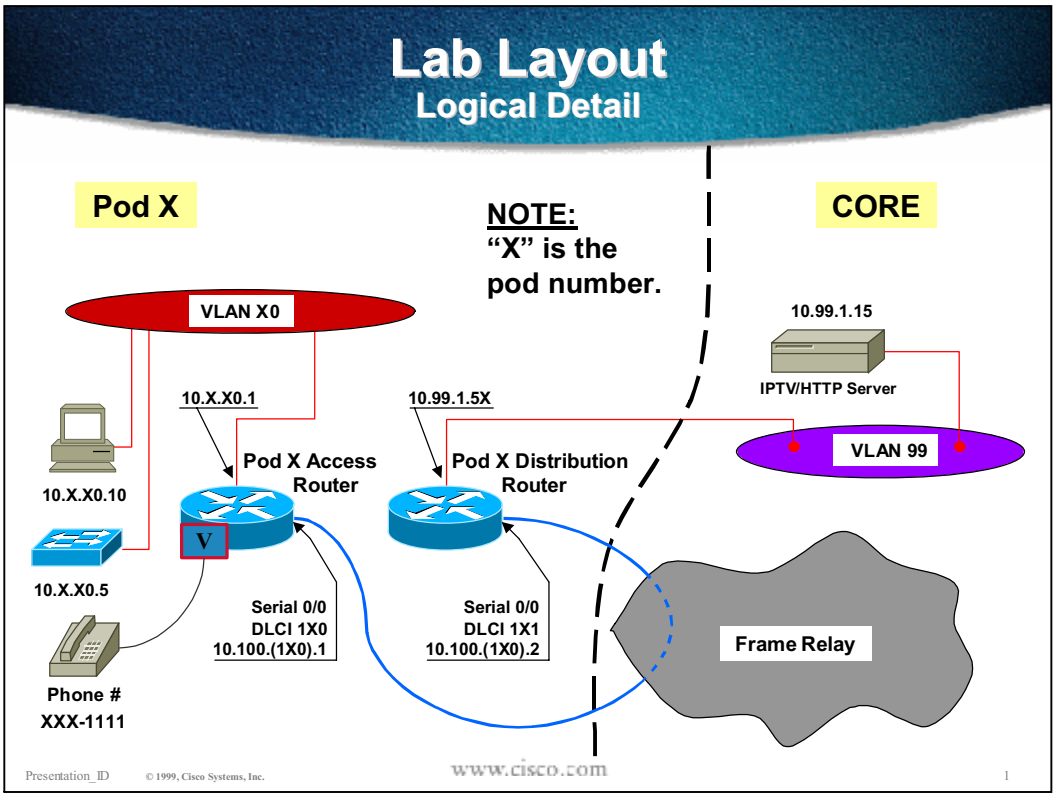
Objectives

In this lab you will complete the following task:

- Implement CAC using RSVP.

Visual Objective

The figure below shows the configuration you will be working with in this exercise.



WAN Connections

In this exercise, you will be connecting to the Core over your WAN links, and applying QoS to that serial connection. The following table provides the specifics of the WAN connections.

	WAN Link				Miscellaneous			
	Access Router		Distribution Router		Access Router Loop 0 Address	Distribution Router Ethernet Address	Analog Phone#	IP Phone#
	DLCI	IP	DLCI	IP				
Pod 1	110	10.100.110.1	111	10.100.110.2	10.1.200.1	10.99.1.51	111-1111	111-2222
Pod 2	120	10.100.120.1	121	10.100.120.2	10.2.200.1	10.99.1.52	222-1111	222-2222
Pod 3	130	10.100.130.1	131	10.100.130.2	10.3.200.1	10.99.1.53	333-1111	333-2222
Pod 4	140	10.100.140.1	141	10.100.140.2	10.4.200.1	10.99.1.54	444-1111	444-2222
Pod 5	150	10.100.150.1	151	10.100.150.2	10.5.200.1	10.99.1.55	555-1111	555-2222
Pod 6	160	10.100.160.1	161	10.100.160.2	10.6.200.1	10.99.1.56	666-1111	666-2222
Pod 7	170	10.100.170.1	171	10.100.170.2	10.7.200.1	10.99.1.57	777-1111	777-2222
Pod 8	180	10.100.180.1	181	10.100.180.2	10.8.200.1	10.99.1.58	888-1111	888-2222

Command List

This table lists the commands you will use in this exercise, in logical order. Refer to this list, lists in previous exercises, or the example configuration at the end of the exercise, if you need further command assistance.

ROUTER COMMANDS		
Prompt	**Command**	**Description**
`router(config)#`	**`call rsvp-sync`**	Enables RSVP globally (enabled by default.)
`router(config-if)#`	**`ip rsvp bandwidth`** *`X Y`*	Enables RSVP on an interface (or sub-interface) with X maximum bandwidth and Y bandwidth per connection.
`router(config-dial-peer)#`	**`acc-qos controlled-load`**	Accepts calls with a guaranteed delay.
`router(config-dial-peer)#`	**`req-qos controlled-load`**	Requests a guaranteed delay.
`router#`	**`show ip rsvp installed`**	Shows bandwidth reservations in place.
`router#`	**`show frame-relay PVC`** *`dlci`*	Verifies FRTS and fragmentation.

Setup

Your instructor has set up the WAN topology per the Lab Layout: Logical Topology Diagram.

Scenario

You have been asked to configure your network so that too many simultaneous voice calls can't be made for the available resources.

Task 1: Implement RSVP

Step 1 On your access and distribution routers, check that "rsvp-sync" is turned on in the configuration.

Step 2 Configure WFQ on both ends of the serial interface.

Step 3 Enable RSVP on both ends of the serial interface. Allow 30Kbps maximum per link and 24Kbps per session.

Step 4 Enable RSVP QoS request and acceptance on the dial-peer pointing to the other pod, and point this dial peer to the other access router's serial IP address vice the loop-back address.

Step 5 Make a voice call, and do a "show ip rsvp installed" on your router. You should see a 24K reservation for the voice call between the pods.

Step 6 Change the RSVP configuration on the PVC to 15Kbps allowed and 10Kbps per session.

Step 7 Attempt to make a call to the other pod. Was it successful?

Task 2: Cleanup.

Remove the RSVP parameters from the PVC and the dial peers.

Completion Criteria:

You have successfully completed this exercise if you saw that RSVP would not allow a call if the reservable bandwidth was insufficient.

Lab Guide for Chapter 9: Management Tools

Overview

This document contains a laboratory exercise that uses QDM to monitor traffic and to configure QoS policy.

Laboratory Exercise 9.1: Traffic Monitoring and QoS Policy Configuration with QDM.

Complete the following laboratory exercise to practice what you learned far in this chapter.

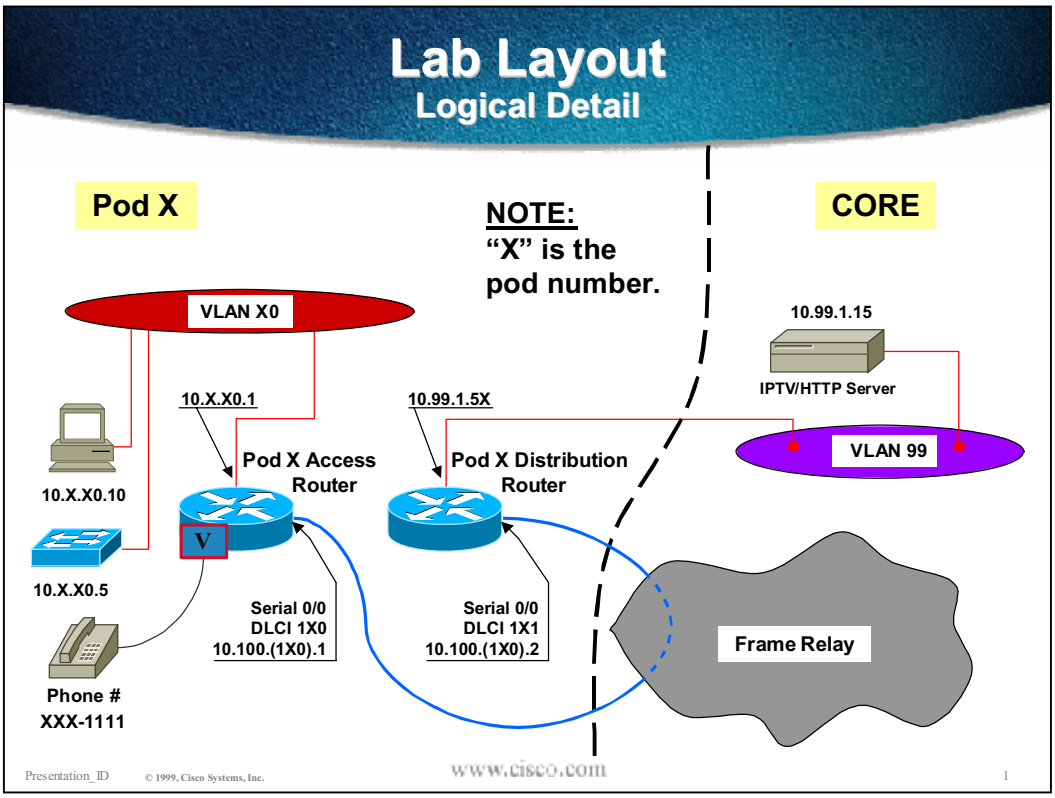
Objectives

In this lab you will complete the following tasks:

- Enable QDM on your router.
- Graphically monitor bandwidth utilization.
- Implement a service policy with QDM.

Visual Objective

The figure below illustrates the configuration you will be working with in this exercise.



WAN Connections

In this exercise, you will be connecting to the Core over your WAN links, and applying QoS to that serial connection. The following table provides the specifics of the WAN connections.

	WAN Link				Miscellaneous			
	Access Router		Distribution Router		Access Router Loop 0 Address	Distribution Router Ethernet Address	Analog Phone#	IP Phone#
	DLCI	IP	DLCI	IP				
Pod 1	110	10.100.110.1	111	10.100.110.2	10.1.200.1	10.99.1.51	111-1111	111-2222
Pod 2	120	10.100.120.1	121	10.100.120.2	10.2.200.1	10.99.1.52	222-1111	222-2222
Pod 3	130	10.100.130.1	131	10.100.130.2	10.3.200.1	10.99.1.53	333-1111	333-2222
Pod 4	140	10.100.140.1	141	10.100.140.2	10.4.200.1	10.99.1.54	444-1111	444-2222
Pod 5	150	10.100.150.1	151	10.100.150.2	10.5.200.1	10.99.1.55	555-1111	555-2222
Pod 6	160	10.100.160.1	161	10.100.160.2	10.6.200.1	10.99.1.56	666-1111	666-2222
Pod 7	170	10.100.170.1	171	10.100.170.2	10.7.200.1	10.99.1.57	777-1111	777-2222
Pod 8	180	10.100.180.1	181	10.100.180.2	10.8.200.1	10.99.1.58	888-1111	888-2222

Command List

This table lists the command you will use in this exercise. Refer to lists in previous exercises, or the sample configuration at the end of the lab if you need further command assistance.

ROUTER COMMANDS		
Prompt	**Command**	**Description**
`Router(config)#`	**`ip http server`**	Enables HTTP services on the router.

Setup

The lab topology is operational per the previous exercises.

Scenario

You are looking for an “easier” way to manage QoS on your network.

Task 1: Enable QDM on your router.

Step 1 Telnet to your access router and enable HTTP services.

Step 2 Close the telnet session.

Step 3 Open up your web browser, and go to an IP address of your access router. The username is “cisco” and password is “san-fran.”

Step 4 Select the QDM link.

Task 2: Graphically monitor the utilization of your WAN link.

Step 1 On the QDM page, select the “monitor” tab.

Step 2 Add a graph for the serial 0/0 interface monitoring bytes per second.

Task 3: Deploy a service policy using QDM.

Step 1 On the QDM page, select the “configure” tab.

Step 2 Reproduce one of the MQC based exercises from a previous lab (your choice.)

Step 3 Observe the changes with a second graph.

Task 4: Clean up.

None.

Completion Criteria:

You have successfully completed this laboratory exercise if you are comfortable with the QDM interface.

Lab Guide for Chapter 10: QoS Design

Overview

This document contains a laboratory exercise for you to practice what you have learned about QoS Design, and practice what you have learned throughout this course.

Laboratory Exercise 10.1: QoS Design

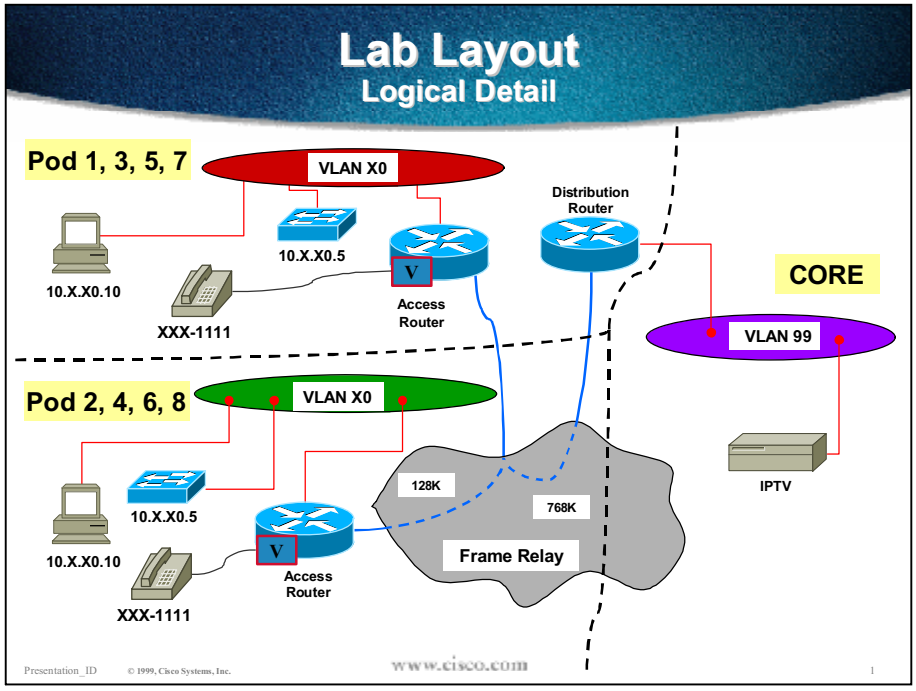
Complete the following laboratory exercise to practice what you have learned.

Objectives

- Given a network topology and a set of performance requirements design and implement a comprehensive QoS strategy to meet those requirements.

Visual Objective

The figure below illustrates the configuration you will be working with in this exercise.



WAN Connections

In this exercise, the odd numbered pods will be connected as before. The even numbered pods access routers will be connected to the odd numbered pods access routers. The following table provides the specifics of the WAN connections.

	"Odd" Access to Distribution WAN Links				"Even" to "Odd" Access Router WAN Links			
	Access Router		Distribution Router		"Even" Router		"Odd" Router	
	DLCI	IP	DLCI	IP	DLCI	IP	DLCI	IP
Pod 1	110	10.100.110.1	111	10.100.110.2	121	10.200.121.1	112	10.200.121.2
Pod 3	130	10.100.130.1	131	10.100.130.2	143	10.200.143.1	134	10.200.143.2
Pod 5	150	10.100.150.1	151	10.100.150.2	165	10.200.165.1	156	10.200.165.2
Pod 7	170	10.100.170.1	171	10.100.170.2	187	10.200.187.1	178	10.200.187.2

Setup

Your instructor has deleted all existing QoS policies and changed the frame relay mapping on the core WAN router to match the topology and addressing above.

Scenario

"California Pocket Keepers" is the world's leading manufacturer of stylish, yet functional pocket protectors. From CPK's headquarters in San Jose, they have 768Kbps connections to regional offices throughout the world. Each regional office has 128Kbps connections to several branch offices.

CPK uses video broadcast from their headquarters only to their regional offices, and estimate the video streams to be around 600Kbps. They have two internal web pages, an "important" one for items that require immediate attention, and a "not_so" one for informational purposes. They want the "important" page to have at least 40Kbps of bandwidth, and the "not_so" one to have 20Kbps.

They use VOIP, but the branch offices have always complained about delay on the calls. Lately over-subscription has been a problem for everyone. They wish to limit the calls to no more than two at any given time over any connection. Finally, they have a custom-developed mission critical application called "vonk," which is not too sensitive to delay but is very sensitive to drops. "Vonk" uses UDP port 1234 as a destination, and requires exactly 6Kbps of bandwidth.

Task: Develop and implement a comprehensive QoS solution to meet CPK's needs. Be prepared to present and justify your solution to the class.

