

# Optimizing Converged Cisco Networks

---

## Volume 1

Version 1.0

## Student Guide

EPGS Production Services: 07.25.06

**Corporate Headquarters**

Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA 95134-1706  
USA  
www.cisco.com  
Tel: 408 526-4000  
800 553-NETS (6387)  
Fax: 408 526-4100

**European Headquarters**

Cisco Systems International BV  
Haarlerbergpark  
Haarlerbergweg 13-19  
1101 CH Amsterdam  
The Netherlands  
www-europe.cisco.com  
Tel: 31 0 20 357 1000  
Fax: 31 0 20 357 1100

**Americas Headquarters**

Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA 95134-1706  
USA  
www.cisco.com  
Tel: 408 526-7660  
Fax: 408 527-0883

**Asia Pacific Headquarters**

Cisco Systems, Inc.  
168 Robinson Road  
#28-01 Capital Tower  
Singapore 068912  
www.cisco.com  
Tel: +65 6317 7777  
Fax: +65 6317 7799

Cisco Systems has more than 200 offices in the following countries and regions. Addresses, phone numbers, and fax numbers are listed on the **Cisco.com Website at [www.cisco.com/go/offices](http://www.cisco.com/go/offices).**

Argentina • Australia • Austria • Belgium • Brazil • Bulgaria • Canada • Chile • China PRC • Colombia • Costa Rica • Croatia • Cyprus • 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



© 2006 Cisco Systems, Inc. All rights reserved. CCSP, CCVP, the Cisco Square Bridge logo, Follow Me Browsing, and StackWise are trademarks of Cisco Systems, Inc.; Changing the Way We Work, Live, Play, and Learn, and iQuick Study are service marks of Cisco Systems, Inc.; and Access Registrar, Aironet, BPX, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Enterprise/Solver, EtherChannel, EtherFast, EtherSwitch, Fast Step, FormShare, GigaDrive, GigaStack, HomeLink, Internet Quotient, IOS, IP/TV, iQ Expertise, the iQ logo, iQ Net Readiness Scorecard, LightStream, Linksys, MeetingPlace, MGX, the Networkers logo, Networking Academy, Network Registrar, Packet, PIX, Post-Routing, Pre-Routing, ProConnect, RateMUX, ScriptShare, SlideCast, SMARTnet, The Fastest Way to Increase Your Internet Quotient, and TransPath are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or Website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0601R)

**DISCLAIMER WARRANTY:** THIS CONTENT IS BEING PROVIDED "AS IS." CISCO MAKES AND YOU RECEIVE NO WARRANTIES IN CONNECTION WITH THE CONTENT PROVIDED HEREUNDER, EXPRESS, IMPLIED, STATUTORY OR IN ANY OTHER PROVISION OF THIS CONTENT OR COMMUNICATION BETWEEN CISCO AND YOU. CISCO SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES, INCLUDING WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE, OR ARISING FROM A COURSE OF DEALING, USAGE OR TRADE PRACTICE. This learning product may contain early release content, and while Cisco believes it to be accurate, it falls subject to the disclaimer above.



***Students, this letter describes important  
course evaluation access information!***

Welcome to Cisco Systems Learning. Through the Cisco Learning Partner Program, Cisco Systems is committed to bringing you the highest-quality training in the industry. Cisco learning products are designed to advance your professional goals and give you the expertise you need to build and maintain strategic networks.

Cisco relies on customer feedback to guide business decisions; therefore, your valuable input will help shape future Cisco course curricula, products, and training offerings. We would appreciate a few minutes of your time to complete a brief Cisco online course evaluation of your instructor and the course materials in this student kit. On the final day of class, your instructor will provide you with a URL directing you to a short post-course evaluation. If there is no Internet access in the classroom, please complete the evaluation within the next 48 hours or as soon as you can access the web.

On behalf of Cisco, thank you for choosing Cisco Learning Partners for your Internet technology training.

Sincerely,

*Cisco Systems Learning*





# Table of Contents

## Volume 1

<b><u>Course Introduction</u></b>	<b><u>1</u></b>
Overview	1
Learner Skills and Knowledge	1
Course Goal and Objectives	3
Course Flow	4
Additional References	5
Cisco Glossary of Terms	5
Your Training Curriculum	6
<b><u>Describe Network Requirements</u></b>	<b><u>1-1</u></b>
Overview	1-1
Module Objectives	1-1
<b><u>Describing Network Requirements</u></b>	<b><u>1-3</u></b>
Overview	1-3
Objectives	1-3
IIN and Cisco SONA Framework	1-4
Intelligent Information Network	1-4
Cisco SONA Framework	1-5
Cisco SONA Layers	1-6
Cisco Network Models	1-7
Cisco Enterprise Architectures	1-7
Cisco Hierarchical Network Model	1-9
Example: Enterprise Network	1-10
Traffic Conditions in a Converged Network	1-11
Network Traffic Mix and Requirements	1-11
Example: Converged Network	1-12
Summary	1-13
References	1-13
Module Summary	1-14
Module Self-Check	1-15
Module Self-Check Answer Key	1-16
<b><u>Describe Cisco VoIP Implementations</u></b>	<b><u>2-1</u></b>
Overview	2-1
Module Objectives	2-1
<b><u>Introducing VoIP Networks</u></b>	<b><u>2-3</u></b>
Overview	2-3
Objectives	2-3
Benefits of Packet Telephony Networks	2-4
Packet Telephony Components	2-6
Two Basic Methods for VoIP	2-7
Analog Interfaces	2-8
Digital Interfaces	2-10
Stages of a Phone Call	2-11
Distributed vs. Centralized Call Control	2-12
Distributed Call Control	2-12
Centralized Call Control	2-14
Summary	2-16
<b><u>Digitizing and Packetizing Voice</u></b>	<b><u>2-17</u></b>
Overview	2-17
Objectives	2-17
Basic Voice Encoding: Converting Analog to Digital	2-18
Analog-to-Digital Conversion Steps	2-19
Basic Voice Encoding: Converting Digital to Analog	2-20
Digital-to-Analog Conversion Steps	2-21

The Nyquist Theorem	2-22
Example: Sampling of Voice	2-23
Quantization	2-24
Quantization Techniques	2-25
Example: Quantization of Voice	2-26
Digital Voice Encoding	2-27
Compression Bandwidth Requirements	2-28
Mean Opinion Score	2-29
What Is a DSP?	2-30
Example: DSP Used for Conferencing	2-31
Example: Transcoding Between Low-Bandwidth Codecs Used in the WAN and a Voice-Mail System Supporting Only G.711	2-32
Summary	2-33
<b>Encapsulating Voice Packets for Transport</b>	<b>2-35</b>
Overview	2-35
Objectives	2-35
End-to-End Delivery of Voice Packets	2-36
Voice Transport in Circuit-Based Networks	2-36
Voice Transport in IP Networks	2-37
Explaining Protocols Used in Voice Encapsulation	2-38
Voice Encapsulation Examples	2-39
Reducing Header Overhead	2-41
Voice Encapsulation Overhead	2-41
RTP Header Compression	2-42
When to Use RTP Header Compression	2-44
Summary	2-45
<b>Calculating Bandwidth Requirements</b>	<b>2-47</b>
Overview	2-47
Objectives	2-47
Impact of Voice Samples and Packet Size on Bandwidth	2-48
Bandwidth Implications of Codecs	2-49
How the Packetization Period Affects VoIP Packet Size and Rate	2-50
VoIP Packet Size and Packet Rate Examples	2-52
Data-Link Overhead	2-53
Security and Tunneling Overhead	2-54
Extra Headers in Security and Tunneling Protocols	2-55
Example: VoIP over IPsec VPN	2-56
Calculating the Total Bandwidth for a VoIP Call	2-57
Total Bandwidth Calculation Procedure	2-58
Illustration of the Bandwidth Calculation	2-60
Quick Bandwidth Calculation	2-62
Effects of VAD on Bandwidth	2-63
VAD Characteristics	2-63
VAD Bandwidth Reduction Examples	2-64
Summary	2-65
<b>Implementing Voice Support in an Enterprise Network</b>	<b>2-67</b>
Overview	2-67
Objectives	2-67
Enterprise Voice Implementations	2-68
Voice Gateway Functions on a Cisco Router	2-69
Cisco Unified CallManager Functions	2-71
Example of Cisco Unified CallManager Functions	2-73
Enterprise IP Telephony Deployment Models	2-74
Example: Single Site	2-75
Example: Multisite with Centralized Call Processing	2-76
Example: Multisite with Distributed Call Processing	2-77
Example: Clustering over WAN	2-78
Identifying Voice Commands in Cisco IOS Configurations	2-79

What Is CAC?	2-82
Example: CAC Deployment	2-83
Summary	2-84
Module Summary	2-85
Module Self-Check	2-86
Module Self-Check Answer Key	2-90
<b><i>Introduction to IP QoS</i></b>	<b><i>3-1</i></b>
Overview	3-1
Module Objectives	3-1
<b><i>Introducing QoS</i></b>	<b><i>3-3</i></b>
Overview	3-3
Objectives	3-3
Converged Network Quality Issues	3-4
Available Bandwidth	3-6
Bandwidth Availability	3-7
Example: Efficient Use of Available Bandwidth	3-9
End-to-End Delay	3-10
The Impact of Delay on Quality	3-11
Ways to Reduce Delay	3-12
Example: Efficient Use of Ways to Reduce Delay	3-14
Packet Loss	3-15
Ways to Prevent Packet Loss	3-17
Example: Packet Loss Solution	3-18
QoS Defined	3-19
Implementing QoS	3-20
QoS Traffic Classes—The Requirements of Different Traffic Types	3-21
Identify Traffic and Its Requirements	3-21
The Requirements of Different Traffic Types	3-22
Example: Traffic Classification	3-22
QoS Policy	3-23
Example: Defining QoS Policies	3-23
Summary	3-24
<b><i>Identifying Models for Implementing QoS</i></b>	<b><i>3-25</i></b>
Overview	3-25
Objectives	3-25
QoS Models	3-26
Best-Effort Model	3-27
Benefits and Drawbacks	3-28
IntServ Model	3-29
IntServ Functions	3-32
Benefits and Drawbacks	3-33
RSVP and the IntServ QoS Model	3-34
Resource Reservation Protocol	3-34
RSVP Operation	3-36
Example: RSVP in Action	3-37
DiffServ Model	3-38
Benefits and Drawbacks	3-39
Summary	3-40
<b><i>Identifying Methods for Implementing QoS</i></b>	<b><i>3-41</i></b>
Overview	3-41
Objectives	3-41
Methods for Implementing QoS Policy	3-42
Legacy CLI	3-43
Legacy CLI Usage Guidelines	3-44
Legacy CLI Example	3-45
Modular QoS CLI	3-46
Modular QoS CLI Components	3-47

Class Maps	3-48
Configuring Class Maps	3-49
ACLs for Traffic Classification	3-50
Policy Maps	3-51
Configuring Policy Maps	3-52
Service Policy	3-53
Attaching Service Policies to Interfaces	3-54
MQC Example	3-55
Basic Verification Commands	3-57
Cisco AutoQoS	3-58
The Features of Cisco AutoQoS	3-60
Cisco AutoQoS Usage Guidelines	3-61
Cisco AutoQoS Example	3-62
Cisco SDM QoS Wizard	3-64
QoS Features	3-65
Getting Started with Cisco SDM	3-66
Creation of a QoS Policy	3-67
QoS Wizard	3-68
Interface Selection	3-69
QoS Policy Generation	3-70
Command Delivery Status	3-74
QoS Status	3-75
QoS Implementation Methods Compared	3-76
Summary	3-77
Module Summary	3-78
Module Self-Check	3-79
Module Self-Check Answer Key	3-82

---

## ***Implement the DiffServ QoS Model*** **4-1**

Overview	4-1
Module Objectives	4-1

## **Introducing Classification and Marking** **4-3**

Overview	4-3
Objectives	4-3
Classification	4-5
Marking	4-6
Classification and Marking at the Link Layer	4-7
Classification and Marking in the Enterprise	4-9
DiffServ Model	4-11
IP Precedence and DSCP Compatibility	4-13
Per-Hop Behaviors	4-14
EF PHB	4-15
AF PHB	4-16
DSCP Summary	4-18
Mapping CoS to Network Layer QoS	4-19
QoS Service Class Defined	4-20
Example: Defining QoS Service Class	4-21
Implementing QoS Policy Using a QoS Service Class	4-22
Example: Application Service Classes	4-24
Trust Boundaries	4-27
Trust Boundaries: IP Phones and PCs	4-29
Summary	4-31

## **Using NBAR for Classification** **4-33**

Overview	4-33
Objectives	4-33
Network-Based Application Recognition	4-34
NBAR Application Support	4-36
Packet Description Language Module	4-40
Protocol Discovery	4-43

Configuring and Monitoring NBAR Protocol Discovery	4-44
Configuring NBAR for Static Protocols	4-46
Example	4-48
Configuring Stateful NBAR for Dynamic Protocols	4-49
Example: Classification of RTP Session	4-53
Summary	4-54
References	4-54
<b>Introducing Queuing Implementations</b>	<b>4-55</b>
Overview	4-55
Objectives	4-55
Congestion and Queuing	4-56
Example: Congestion Caused by Speed Mismatch	4-57
Example: Congestion Caused by Aggregation	4-58
Queuing Algorithms	4-59
Congestion and Queuing	4-60
Queuing Algorithm Introduction	4-61
FIFO	4-62
Priority Queuing	4-63
Round Robin	4-65
Weighted Round Robin	4-66
Router Queuing Components	4-68
The Software Queue	4-70
The Hardware Queue	4-71
Congestion on Software Interfaces	4-73
Summary	4-74
<b>Configuring WFQ</b>	<b>4-75</b>
Overview	4-75
Objectives	4-75
Weighted Fair Queuing	4-76
WFQ Architecture and Benefits	4-77
WFQ Classification	4-78
WFQ Insertion and Drop Policy	4-80
Benefits and Drawbacks of WFQ	4-81
Configuring and Monitoring WFQ	4-82
Additional WFQ Configuration Parameters	4-83
Monitoring WFQ	4-84
Summary	4-86
<b>Configuring CBWFQ and LLQ</b>	<b>4-87</b>
Overview	4-87
Objectives	4-87
Describing Advanced Queuing Mechanisms	4-88
Class-Based Weighted Fair Queuing	4-89
CBWFQ Architecture and Benefits	4-90
Classification	4-91
Scheduling	4-92
Available Bandwidth	4-93
CBWFQ Benefits and Drawbacks	4-94
Configuring and Monitoring CBWFQ	4-95
Example of CBWFQ	4-98
Monitoring CBWFQ	4-99
Low Latency Queuing	4-101
LLQ Architecture and Benefits	4-102
LLQ Benefits	4-103
Configuring and Monitoring LLQ	4-104
Monitoring LLQ	4-107
Summary	4-108

<b>Introducing Congestion Avoidance</b>	<b>4-109</b>
Overview	4-109
Objectives	4-109
Managing Interface Congestion with Tail Drop	4-110
Tail Drop Limitations	4-111
TCP Synchronization	4-112
TCP Delay, Jitter, and Starvation	4-113
Random Early Detection	4-114
RED Profiles	4-115
RED Modes	4-116
TCP Traffic Before and After RED	4-117
Weighted Random Early Detection	4-118
WRED Building Blocks	4-120
Class-Based WRED	4-121
WRED Profiles	4-122
DSCP-Based WRED (Expedited Forwarding)	4-123
Configuring CBWRED	4-124
Changing the WRED Traffic Profile	4-125
CBWFQ Using IP Precedence with CBWRED: Example	4-127
WRED Profiles: DSCP-Based WRED (AF)	4-129
Configuring DSCP-Based CBWRED	4-130
Changing the WRED Traffic Profile	4-131
CBWRED Using DSCP with CBWFQ: Example	4-132
Monitoring CBWRED	4-134
Summary	4-135
<b>Introducing Traffic Policing and Shaping</b>	<b>4-137</b>
Overview	4-137
Objectives	4-137
Traffic Policing and Shaping Overview	4-138
Why Use Policing?	4-139
Why Use Shaping?	4-140
Why Use Traffic Conditioners?	4-141
Traffic Policing and Shaping: Example	4-142
Policing vs. Shaping	4-143
Measuring Traffic Rates	4-144
Single Token Bucket Class-Based Policing	4-146
Cisco IOS Traffic Policing and Shaping Mechanisms	4-147
Cisco IOS Traffic-Shaping Mechanisms	4-148
Applying Traffic Conditioners	4-149
Summary	4-150
<b>Understanding WAN Link Efficiency Mechanisms</b>	<b>4-151</b>
Overview	4-151
Objectives	4-151
Link Efficiency Mechanisms Overview	4-152
Compression	4-152
Link Efficiency Mechanisms	4-154
Layer 2 Payload Compression	4-156
Layer 2 Payload Compression Results	4-157
Header Compression	4-158
Header Compression Results	4-160
Large Packets “Freeze Out” Voice on Slow WAN Links	4-161
Link Fragmentation and Interleaving	4-162
Applying Link Efficiency Mechanisms	4-163
Example	4-164
Summary	4-165

<b>Implementing QoS Preclassify</b>	<b>4-167</b>
Overview	4-167
Objectives	4-167
Virtual Private Networks	4-168
VPN Types	4-169
Encryption Overview	4-170
VPN Protocols	4-171
Implementing QoS with Preclassification	4-172
QoS Preclassify Applications	4-173
GRE Tunneling	4-174
IPsec AH	4-175
IPsec ESP	4-176
QoS Preclassification Deployment Options	4-177
Configuring QoS Preclassify	4-179
Example	4-180
Summary	4-181
<b>Deploying End-to-End QoS</b>	<b>4-183</b>
Overview	4-183
Objectives	4-183
QoS SLAs	4-184
Enterprise Network with Traditional Layer 2 Service	4-185
Enterprise Network with IP Service	4-186
Know the SLA Offered by Your Service Provider	4-187
Typical SLA Requirements for Voice	4-188
Deploying End-to-End QoS	4-189
Enterprise Campus QoS Implementations	4-191
Campus Access and Distribution Layer QoS Implementation	4-193
WAN Edge QoS Implementations	4-195
Traffic Leaving Enterprise Network	4-196
Traffic Leaving Service Provider Network	4-198
Example: Managed Customer Edge with Three Service Classes	4-200
WAN Edge Design	4-202
Customer Edge-to-Provider Edge QoS for Frame Relay Access:	
Customer Edge Outbound	4-203
Customer Edge-to-Provider Edge QoS for Frame Relay Access:	
Provider Edge Inbound	4-205
What Is CoPP?	4-206
Cisco Router Planes	4-207
CoPP Deployment	4-208
CoPP Example	4-209
Summary	4-210
Module Summary	4-211
Module Self-Check	4-213
Module Self-Check Answer Key	4-215





# Course Introduction

---

## Overview

As converged networks and mobility are used more and more in daily business, these technologies need to be optimized to support business requirements.

You will learn about the new Cisco Intelligent Information Network (IIN) model and the Cisco Service-Oriented Network Architecture (SONA) as architectural frameworks for converged networks.

You will review VoIP network essentials and focus on the VoIP-related challenges in such networks. To ensure quality in a converged network, you will deal with concepts and implementation methods for quality of service (QoS). Finally, you will learn about the evolution of wireless security standards and the elements of the Cisco wireless LAN (WLAN) network. You will work on case studies and several lab activities based on Cisco Integrated Services Routers (ISRs) and the converged network topics.

## Learner Skills and Knowledge

This subtopic lists the skills and knowledge that learners must possess to benefit fully from the course. The subtopic also includes recommended Cisco learning offerings that learners should first complete to benefit fully from this course.

## Learner Skills and Knowledge

### **Skills and knowledge:**

- Completed initial configuration of a switch
- Basic interswitch connections
- Completed initial configuration of a router
- Routing (static routing, default routing, default router, default gateway, and basic NAT and PAT)
- Concepts linked to routing protocols (classful versus classless, single-area OSPF, RIP, EIGRP, administrative distance, and interoperations)
- Standard WAN technologies (Frame Relay, PPP, and HDLC)
- Fundamental security knowledge, including the presence of hackers, viruses, and other security threats

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—3

## Learner Skills and Knowledge (Cont.)

- **Skills and knowledge:**
  - Fundamental knowledge of IP addressing including the format of IPv4 addresses, the concept of subnetting, VLSM, and CIDR, as well as static and default routing
  - Standard and extended ACLs
  - Client utilities including Telnet, IPCONFIG, trace route, ping, FTP, TFTP, and HyperTerminal
  - Basic familiarity with Cisco IOS software, including accessing the CLI on a Cisco device and implementing the debug and show commands
- **Cisco learning offerings:**
  - Introduction to Cisco Networking Technologies (INTRO)
  - Interconnecting Cisco Network Devices (ICND)

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—4

# Course Goal and Objectives

This topic describes the course goal and objectives.

**“The goal of the ONT course is to teach learners how to optimize their converged enterprise networks.”**

Optimizing Converged Cisco Networks



Upon completing this course, you will be able to meet these objectives:

- Describe the converged network requirements within the Cisco conceptual network models, with a focus on performance and wireless security
- Describe Cisco VoIP implementations
- Describe the need to implement QoS and the methods for implementing QoS on a converged network using Cisco routers and Catalyst switches
- Explain the key IP QoS mechanisms used to implement the DiffServ QoS model
- Configure Cisco AutoQoS for Enterprise
- Describe and configure wireless security and basic wireless management

# Course Flow

This topic presents the suggested flow of the course materials.

# Course Flow

	Day 1	Day 2	Day 3	Day 4	Day 5
A M	Course Introduction	Introduction to IP QoS	Implement the DiffServ QoS Model	Implement the DiffServ QoS Model	Implement Wireless Scalability
	Describe Network Requirements		Lab 4-1	Lab 4-6	Lab 6-1
	Describe Cisco VoIP Implementations		Implement the DiffServ QoS Model Lab 4-2	Implement AutoQoS	Lab 6-2
	Lunch				
P M	Lab 2-1	Case Study: 3-1	Implement the DiffServ QoS Model	Lab 5-1	Lab 6-3
	Describe Cisco VoIP Implementations	Lab 3-2	Lab 4-3	Lab 5-2	Implement Wireless Scalability
			Implement the DiffServ QoS Model		
			Lab 4-4		
Lab 2-2	Implement the DiffServ QoS Model	Lab 4-5	Lab 5-3	Lab 6-4	

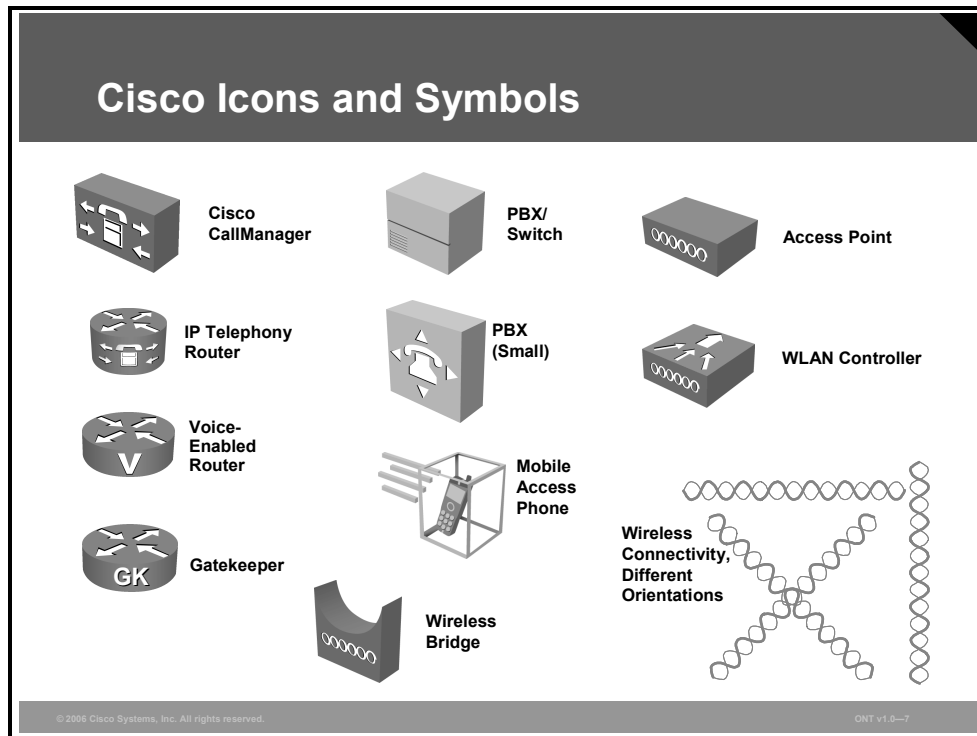
© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0-6

The schedule reflects the recommended structure for this course. This structure allows enough time for the instructor to present the course information and for you to work through the lab activities. The exact timing of the subject materials and labs depends on the pace of your specific class.

# Additional References

This topic presents the Cisco icons and symbols that are used in this course, as well as information on where to find additional technical references.



## Cisco Glossary of Terms

For additional information on Cisco terminology, refer to the *Cisco Internetworking Terms and Acronyms* glossary of terms at <http://www.cisco.com/univercd/cc/td/doc/cisintwk/ita/index.htm>.

# Your Training Curriculum

This topic presents the training curriculum for this course.

## Cisco Career Certifications



[www.cisco.com/go/certifications](http://www.cisco.com/go/certifications)

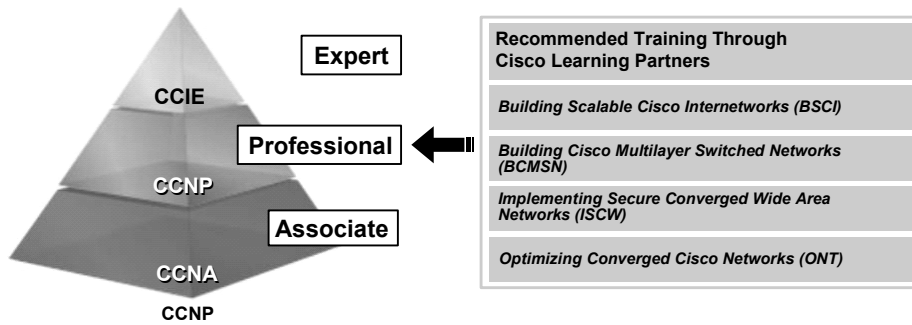
© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—8

You are encouraged to join the Cisco Certification Community, a discussion forum open to anyone holding a valid Cisco Career Certification (such as Cisco CCIE<sup>®</sup>, CCNA<sup>®</sup>, CCDA<sup>®</sup>, CCNP<sup>®</sup>, CCDP<sup>®</sup>, CCIP<sup>®</sup>, CCVP<sup>™</sup>, or CCSP<sup>™</sup>). It provides a gathering place for Cisco certified professionals to share questions, suggestions, and information about Cisco Career Certification programs and other certification-related topics. For more information, visit [www.cisco.com/go/certifications](http://www.cisco.com/go/certifications).

# Cisco Career Certifications: CCNP

**Expand Your Professional Options  
and Advance Your Career**

Professional-level recognition in CCNP



[www.cisco.com/go/certifications](http://www.cisco.com/go/certifications)

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0-9





# Describe Network Requirements

---

## Overview

This module introduces the concept of converged networks that carry voice, video, and data and explains various architectures and network models that accommodate the integrated services within converged networks. The concepts of the Intelligent Information Network (IIN), Cisco Service-Oriented Network Architecture (SONA), and Cisco Enterprise Architectures are explained, with a focus on the traffic conditions and requirements imposed by converged applications. The stress is on performance and security requirements in fixed and wireless networks.

## Module Objectives

Upon completing this module, you will be able to describe the converged network requirements within the Cisco conceptual network models with a focus on performance and wireless security.



# Describing Network Requirements

---

## Overview

The convergence of voice, video, and data has not only changed conceptual network models, but it has also affected the way in which networks support services and applications. This lesson starts by introducing the Cisco vision of the Intelligent Information Network (IIN) and the Cisco Service-Oriented Network Architecture (SONA). This architectural framework shifts the view of the network from pure traffic transport toward a services and applications orientation. The Cisco Enterprise Architectures are explained and aligned with the traditional three-layer hierarchical network model. The traffic patterns in converged networks are discussed, and the lesson concludes with the requirements for integrated services (including performance and security) that converged applications impose on these networks.

## Objectives

Upon completing this lesson, you will be able to describe the converged network requirements of various network and networked applications within the Cisco network architectures. This ability includes being able to meet these objectives:

- Describe the IIN and the Cisco SONA framework
- Explain the Cisco conceptual network models, such as Cisco Enterprise Architectures and Cisco hierarchical network model
- Describe the traffic conditions in a converged network

# IIN and Cisco SONA Framework

This topic describes the IIN and its features and Cisco SONA, which guides an evolution of enterprise networks toward IIN.

## Intelligent Information Network

- **IIN integrates networked resources and information assets.**
- **IIN extends intelligence across multiple products and infrastructure layers.**
- **IIN actively participates in the delivery of services and applications.**
- **Three phases in building an IIN are:**
  - Integrated transport
  - Integrated services
  - Integrated applications

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—1.3

## Intelligent Information Network

The Cisco vision of the Intelligent Information Network (IIN) encompasses these features:

- **Integration of networked resources and information assets that have been largely **unlinked**:** The modern converged networks with integrated voice, video, and data require that IT departments more closely link the IT infrastructure with the network.
- **Intelligence across multiple products and infrastructure layers:** The intelligence built into each component of the network is extended networkwide and applies end to end.
- **Active participation of the network in the delivery of services and applications:** With added intelligence within the network devices, the IIN makes it possible for the network to actively manage, monitor, and optimize service and application delivery across the entire IT environment.

With the listed features, the IIN offers much more than basic connectivity, bandwidth for users, and access to applications. The IIN offers end-to-end functionality and centralized, unified control that promotes true business transparency and agility.

The IIN technology vision offers an evolutionary approach that consists of three phases in which functionality can be added to the infrastructure as required:

- **Integrated transport:** Everything—data, voice, and video—consolidates onto an IP network for secure network convergence. By integrating data, voice, and video transport into a single, standards-based, modular network, organizations can simplify network management and generate enterprisewide efficiencies. Network convergence also lays the foundation for a new class of IP-enabled applications delivered through Cisco IP Communications solutions.

- **Integrated services:** After the network infrastructure has been converged, IT resources can be pooled and shared or “virtualized” to flexibly address the changing needs of the organization. Integrated services help to unify common elements, such as storage and data center server capacity. By extending virtualization capabilities to encompass server, storage, and network elements, an organization can transparently use all of its resources more efficiently. Business continuity is also enhanced because shared resources across the IIN provide services in the event of a local systems failure.
- **Integrated applications:** With Cisco Application-Oriented Networking (AON) technology, Cisco has entered the third phase of building the IIN. This phase focuses on making the network “application aware,” so it can optimize application performance and more efficiently deliver networked applications to users. In addition to capabilities such as content caching, load balancing, and application-level security, Cisco AON makes it possible for the network to simplify the application infrastructure by integrating intelligent application message handling, optimization, and security into the existing network.

## Cisco SONA Framework

With its vision of the IIN, Cisco is helping organizations to address new IT challenges, such as the deployment of service-oriented architectures, web services, and virtualization.

### Cisco SONA Framework

- **Cisco SONA is an architectural framework.**
- **Cisco SONA brings several advantages to enterprises:**
  - **Outlines how enterprises can evolve toward the IIN**
  - **Illustrates how to build integrated systems across a fully converged intelligent network**
  - **Improves flexibility and increases efficiency**

© 2006 Cisco Systems, Inc. All rights reserved.
ONT v1.0—1-4

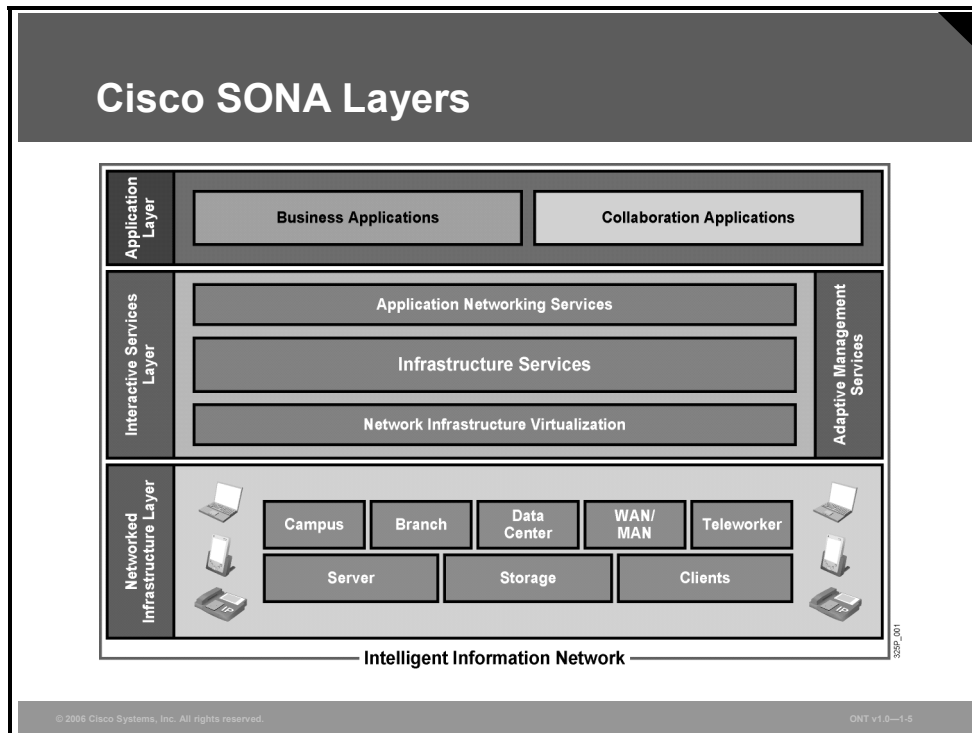
Cisco Service-Oriented Network Architecture (SONA) is an architectural framework that guides the evolution of enterprise networks to an IIN. The Cisco SONA framework provides several advantages to enterprises:

- Outlines the path toward the IIN
- Illustrates how to build integrated systems across a fully converged IIN
- Improves flexibility and increases efficiency, which results in optimized applications, processes, and resources

Cisco SONA uses the extensive product line services, proven architectures, and experience of Cisco and its partners to help enterprises achieve their business goals.

# Cisco SONA Layers

The Cisco SONA framework shows how integrated systems can both allow a dynamic, flexible architecture and provide for operational efficiency through standardization and virtualization. In this framework, the network is the common element that connects and enables all components of the IT infrastructure.

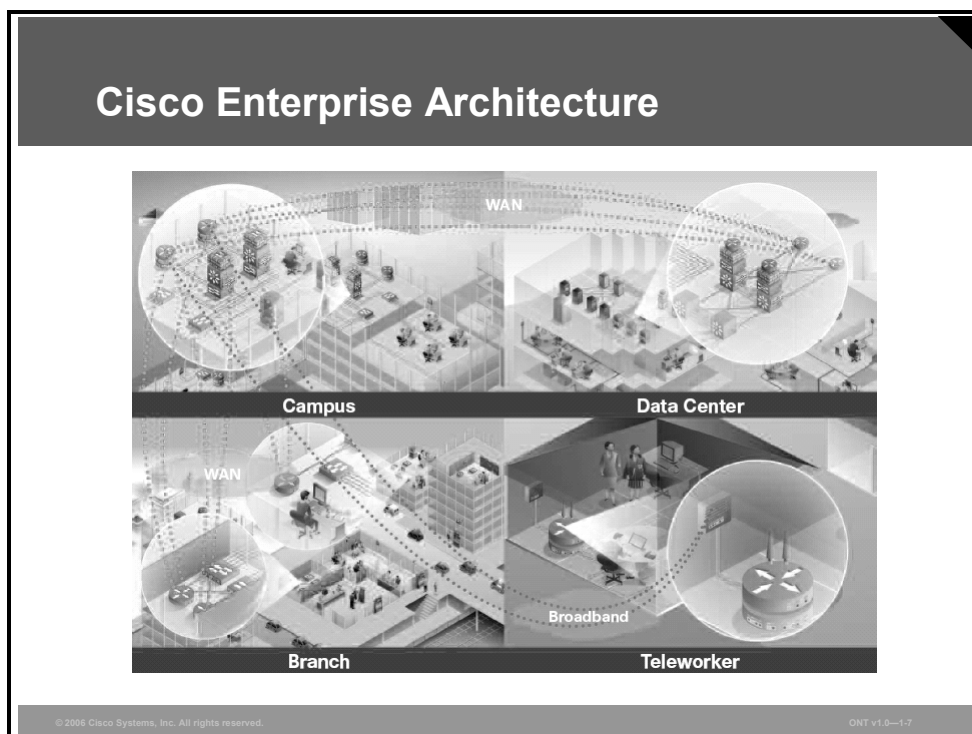


Cisco SONA outlines three layers of the IIN:

- The *networked infrastructure layer*, where all IT resources are interconnected across a converged network foundation. IT resources include servers, storage, and clients. The network infrastructure layer represents how these resources exist in different places in the network, including the campus, branch, data center, WAN, metropolitan area network (MAN), and teleworker. The objective for customers in this layer is to have the connectivity anywhere and anytime.
- The *interactive services layer*, which enables efficient allocation of resources to applications and business processes delivered through the networked infrastructure. This layer comprises these elements:
  - Voice and collaboration services
  - Mobility services
  - Security and identity services
  - Storage services
  - Computer services
  - Application networking services
  - Network infrastructure virtualization
  - Services management
  - Adaptive management services
- The *application layer* includes business applications and collaboration applications. The objective for customers in this layer is to meet business requirements and achieve efficiencies by leveraging the interactive services layer.

# Cisco Network Models

This topic describes Cisco network models, with the Cisco Enterprise Architectures and their mapping to the traditional three-layer hierarchical network model.



## Cisco Enterprise Architectures

Cisco provides the enterprisewide systems architecture that helps companies to protect, optimize, and grow the infrastructure that supports business processes. The architecture provides for integration of the entire network—campus, data center, WAN, branches, and teleworkers—offering staff secure access to tools, processes, and services.

The Cisco Enterprise *Campus* Architecture combines a core infrastructure of intelligent switching and routing with tightly integrated productivity-enhancing technologies, including IP communications, mobility, and advanced security. The architecture provides the enterprise with high availability through a resilient multilayer design, redundant hardware and software features and automatic procedures for reconfiguring network paths when failures occur. Multicast provides optimized bandwidth consumption, and quality of service (QoS) prevents oversubscription to ensure that real-time traffic, such as voice and video, or mission-critical data is not dropped or delayed. Integrated security protects against and mitigates the impact of worms, viruses, and other attacks on the network—even at the switch port level. The Cisco enterprisewide architecture extends authentication support using standards such as 802.1x and Extensible Authentication Protocol (EAP). It also provides the flexibility to add IPsec and Multiprotocol Label Switching (MPLS) virtual private networks (VPNs), identity and access management, and VLANs to compartmentalize access. This structure helps improve performance and security and decreases costs.

The Cisco Enterprise *Data Center* Architecture is a cohesive, adaptive network architecture that supports the requirements for consolidation, business continuance, and security while enabling emerging service-oriented architectures, virtualization, and on-demand computing. IT staff can easily provide departmental staff, suppliers, or customers with secure access to applications and resources. This structure simplifies and streamlines management, significantly reducing overhead. Redundant data centers provide backup using synchronous and asynchronous data and application replication. The network and devices offer server and application load balancing to maximize performance. This solution allows the enterprise to scale without major changes to the infrastructure.

The Cisco Enterprise *Branch* Architecture allows enterprises to extend head-office applications and services, such as security, IP communications, and advanced application performance, to thousands of remote locations and users or to a small group of branches.

---

<b>Note</b>	Cisco integrates security, switching, network analysis, caching, and converged voice and video services into a series of integrated services routers (ISRs) in the branch—so that the enterprises can deploy new services when they are ready, without buying new equipment.
-------------	--

---

This solution provides secure access to voice, mission-critical data, and video applications—anywhere, anytime. Advanced network routing, VPNs, redundant WAN links, application content caching, and local IP telephony call processing provide a robust architecture with high levels of resilience for all branch offices. An optimized network leverages the WAN and LAN to reduce traffic, and save bandwidth and operational expenses. The enterprise can easily support branch offices with the ability to centrally configure, monitor, and manage devices located at remote sites, including tools such as Cisco AutoQoS or the Security Device Manager (SDM) GUI QoS Wizard, that proactively resolve congestion and bandwidth issues before they affect network performance.

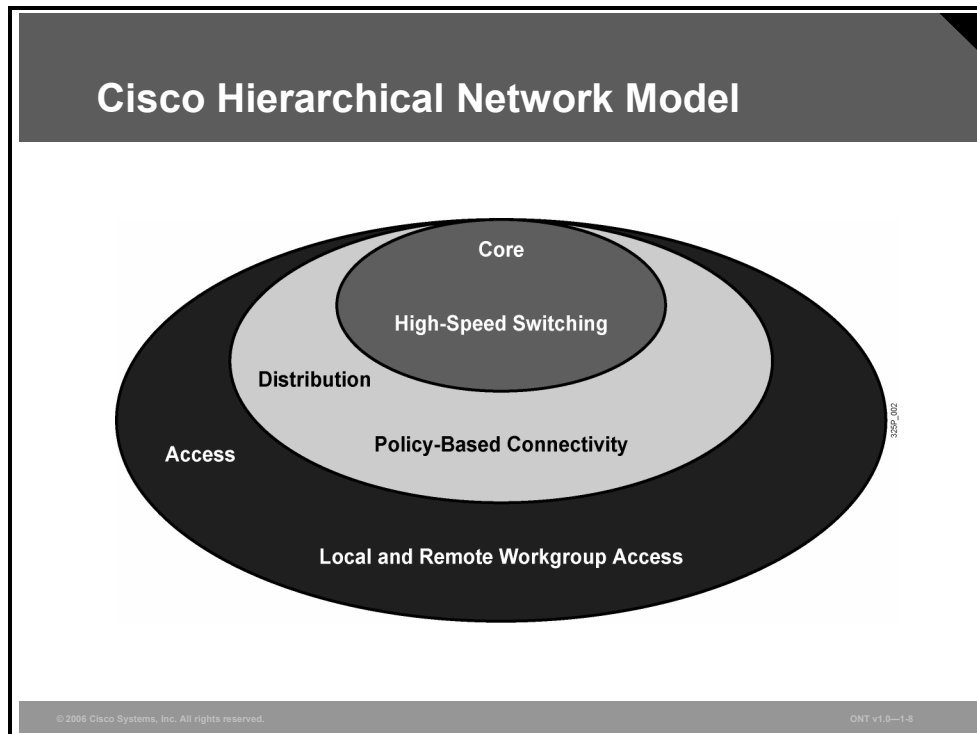
The Cisco Enterprise *Teleworker* Architecture allows enterprises to securely deliver voice and data services to remote small or home offices over a standard broadband access service, providing a business-resiliency solution for the enterprise and a flexible work environment for employees. Centralized management minimizes IT support costs, and robust integrated security addresses the unique security challenges of this environment. Integrated security and identity-based networking services (comprising authentication, access control, and user policies to secure network connectivity and resources) enable the enterprise to help extend campus security policies to the teleworker. Staff can securely log into the network over an always-on VPN and gain access to authorized applications and services from a single cost-effective platform. Productivity can further be enhanced by adding an IP phone, providing cost-effective access to a centralized IP communications system with voice and unified messaging services.

The Cisco Enterprise *WAN* Architecture offers the convergence of voice, video, and data services over a single IP communications network. This solution enables the enterprise to cost-effectively span large geographic areas. QoS, granular service levels, and comprehensive encryption options help ensure the secure delivery of high-quality corporate voice, video, and data resources to all corporate sites, enabling staff to work productively and efficiently wherever they are located. Security is provided with multiservice VPNs (IPsec and MPLS) over Layer 2 or Layer 3 WANs, hub-and-spoke topologies, or full-mesh topologies.



# Cisco Hierarchical Network Model

Traditionally, the three-layer hierarchical model has been used in network design.



The model provides a modular framework that allows flexibility in network design and facilitates ease of implementation and troubleshooting. The hierarchical model divides networks or their modular blocks into the access, distribution, and core layers, with these features:

- The *access layer* is used to grant user access to network devices. In a network campus, the access layer generally incorporates switched LAN devices with ports that provide connectivity to workstations and servers. In the WAN environment, the access layer at remote sites or at a teleworker location may provide access to the corporate network across WAN technology.
- The *distribution layer* aggregates the wiring closets, and uses switches to segment workgroups and isolate network problems in a campus environment. Similarly, the distribution layer aggregates WAN connection at the edge of the campus and provides policy-based connectivity.
- The *core layer* (also referred to as the backbone) is a high-speed backbone and is designed to switch packets as fast as possible. Because the core is critical for connectivity, it must provide a high level of availability and adapt to changes very quickly.

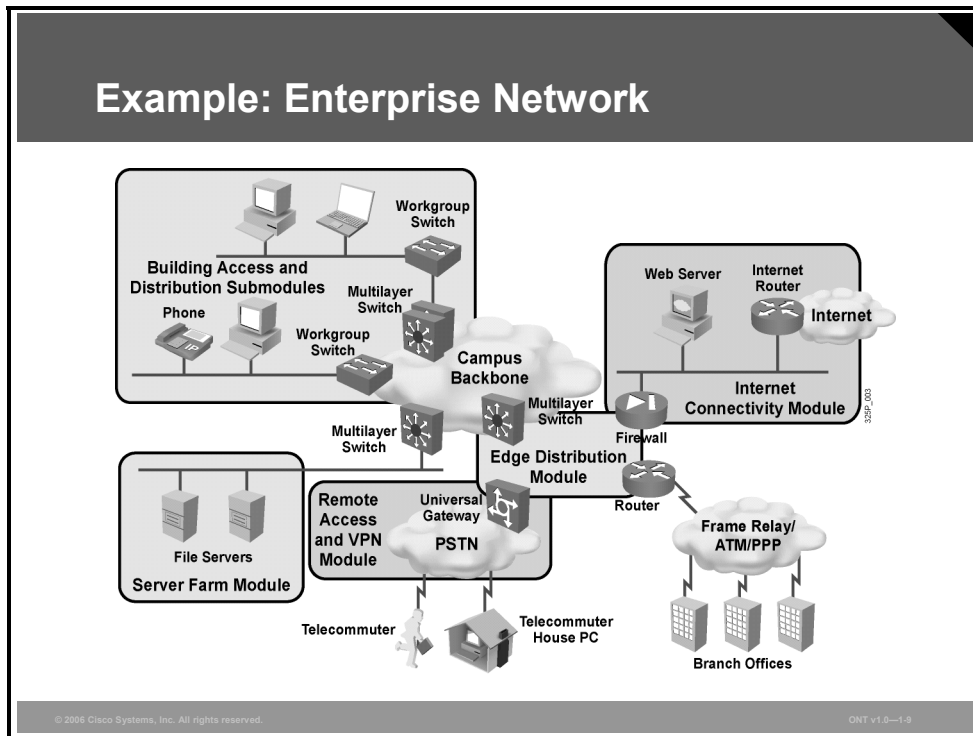
---

**Note** The hierarchical model can be applied to any network type, such as LANs, WANs, wireless LANs (WLANs), MANs, and VPNs, and to any modular block of the Cisco networking model.

---

# Example: Enterprise Network

The example shows a network that was deployed following the Cisco Enterprise Architectures and the Cisco hierarchical model design.



Various architectures and submodules form an integrated converged network that supports business processes.

The campus comprises five submodules:

- Building access with access switches and end devices (PCs and IP phones)
- Building distribution with distribution multilayer switches
- Backbone
- Edge distribution that concentrates all branches and teleworkers accessing the campus via WAN or Internet
- Server farm that represents the data center

Additional submodules represent remote access and VPN, Internet, and traditional WAN (Frame Relay, ATM, and leased lines with PPP).

# Traffic Conditions in a Converged Network

This topic describes the traffic types and requirements in converged networks.

## Network Traffic Mix and Requirements

- **Converged network traffic mix:**
  - **Voice and video traffic**
  - **Voice applications traffic**
  - **Mission-critical applications traffic**
  - **Transactional traffic**
  - **Routing update traffic**
  - **Network management traffic**
  - **Bulk transfer (best-effort) and scavenger (less-than-best-effort) traffic**
- **Key requirements:**
  - **Performance (bandwidth, delay, and jitter)**
  - **Security (access and transmission)**

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—1-11

## Network Traffic Mix and Requirements

Converged networks with integrated voice, video, and data contain various traffic patterns:

- Voice and video traffic (for example, IP telephony, and video broadcast and conferencing); video traffic frequently carried as IP multicast traffic
- Voice applications traffic, generated by voice-related applications (such as contact centers)
- Mission-critical traffic, generated, for example, by stock exchange applications
- Transactional traffic, generated by e-commerce applications
- Routing update traffic, from routing protocols such as Routing Information Protocol (RIP), Open Shortest Path First (OSPF), Enhanced Interior Gateway Routing Protocol (EIGRP), Intermediate System-to-Intermediate System (IS-IS), and Border Gateway Protocol (BGP)
- Network management traffic
- Bulk transfer (such as file transfer or HTTP), considered best-effort traffic
- Scavenger (casual entertainment, rogue traffic), considered less-than-best-effort traffic

The diversity of the traffic mix imposes stringent performance and security requirements on the network. The requirements differ significantly, depending on the traffic type. For example, voice and video require constant bandwidth with low delay and jitter, while transactional traffic requires high reliability and security with relatively low bandwidth. Also, voice applications, such as IP telephony, require high reliability and availability, because users expect the “dial tone” in the IP network to be exactly the same as in traditional phone network. To meet the traffic requirements in the network, voice and video traffic must be treated differently from other traffic, such as web-based (HTTP) traffic. QoS mechanisms are mandatory in converged networks.

Security is a key issue in fixed networks but is even more important in wireless mobility, where access to the network is possible from virtually anywhere. Several security strategies, such as device hardening with strict access control and authentication, intrusion protection, intrusion detection, and traffic protection with encryption, can mitigate network security threats.

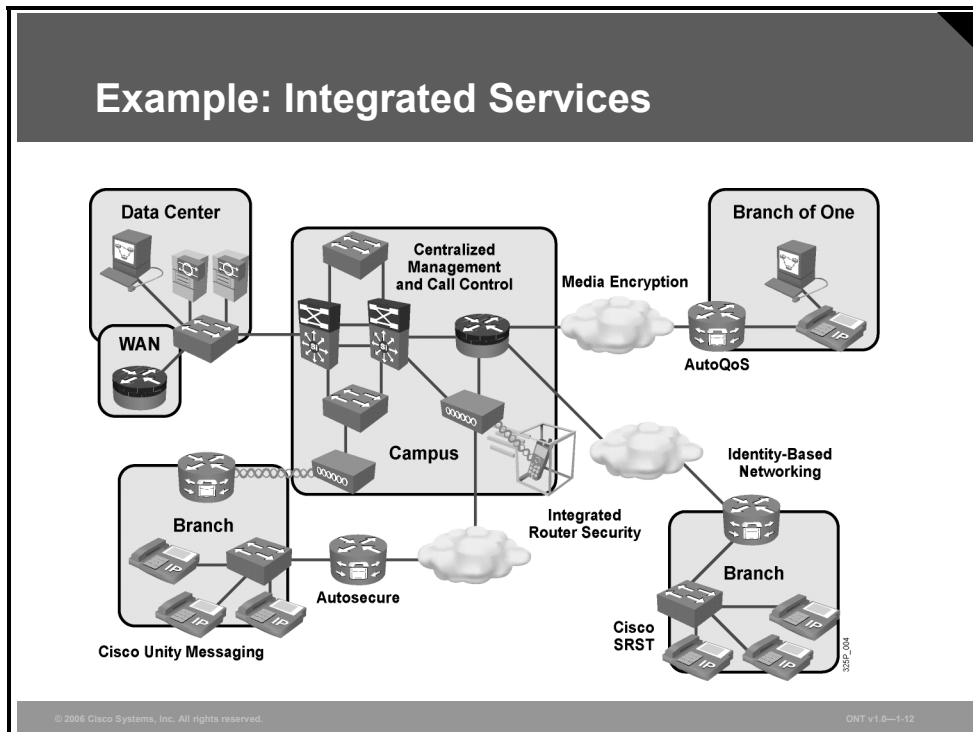
---

**Note** Converged networks span the entire range of access options, from fixed networks to WLANs and mobile wireless networks.

---

## Example: Converged Network

The figure shows a sample converged network with integrated secured services where advanced technologies, such as IP communications (IP telephony and unified messaging), wireless mobility, and security, have been deployed.



The clouds represent the Cisco Enterprise WAN Architecture. The links in this area can easily become a bottleneck that affects the performance of IP telephony. One of the solutions to this issue, Cisco AutoQoS, is shown on the voice-enabled router in a simple branch consisting of one user. To increase IP telephony reliability and availability, Cisco Survivable Remote Site Telephony (SRST) has been deployed at one of the branches.

To increase the security of the deployed services, encryption and identity-based networking (including authentication and access control) have been implemented, in addition to device hardening with integrated router security. The AutoSecure feature is used to provide simple and straightforward “one touch” device lockdown. The need for increased security is driven also by the wireless mobility that has been deployed in the sample network.

---

**Note** Network performance, which is based on application types and network security, is a key part of enabling the network to provide business value to the enterprise.

---

# Summary

This topic summarizes the key points that were discussed in this lesson.

## Summary

- **Cisco SONA framework guides the evolution of the enterprise network toward IIN.**
- **Cisco Enterprise Architectures with a hierarchical network model facilitate the deployment of converged networks.**
- **Converged networks with their traffic mix make higher demands on the network and its resources.**
- **Performance and security are key requirements in converged networks with wireless mobility.**

© 2006 Cisco Systems, Inc. All rights reserved.

QNT v1.0—1-13

## References

For additional information, refer to these resources:

- Cisco Systems, Inc. Intelligent Information Network page at [http://www.cisco.com/en/US/netsol/ns650/networking\\_solutions\\_market\\_segment\\_solution.html](http://www.cisco.com/en/US/netsol/ns650/networking_solutions_market_segment_solution.html).
- Cisco Systems, Inc. Service-Oriented Network Architecture page at [http://www.cisco.com/en/US/netsol/ns629/networking\\_solutions\\_market\\_segment\\_solutions\\_home.html](http://www.cisco.com/en/US/netsol/ns629/networking_solutions_market_segment_solutions_home.html).
- Cisco Systems, Inc. Enterprise Architectures page at [http://www.cisco.com/en/US/netsol/ns517/networking\\_solutions\\_market\\_segment\\_solutions\\_home.html](http://www.cisco.com/en/US/netsol/ns517/networking_solutions_market_segment_solutions_home.html).

# Module Summary

This topic summarizes the key points that were discussed in this module.

## Module Summary

- **The requirements of converged networks carrying voice, video, and data have changed the conceptual network models.**
- **The IIN, Cisco SONA, and Cisco Enterprise Architectures models provide a framework for deployment of converged networks.**
- **The service layer of the Cisco SONA architecture addresses the performance and security requirements of converged networks.**

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—1-1

The requirements of converged networks carrying voice, video, and data have changed the conceptual network models that have been used as a framework for deploying traditional data networks. Cisco models, such as the Intelligent Information Network (IIN), Cisco Service-Oriented Network Architecture (SONA), and Cisco Enterprise Architectures, serve as architectural models for deploying converged networks. The shift from an infrastructure perspective to an applications perspective requires that services be integrated into the conceptual models. The service layer of the Cisco SONA architecture, with all its integrated services, addresses the key requirements—performance and security—of converged networks.

# Module Self-Check

Use the questions here to review what you learned in this module. The correct answers and solutions are found in the Module Self-Check Answer Key.

- Q1) Which Cisco SONA interactive service plays a key role in wireless mobility? (Source: Describing Network Requirements)
- A) storage services
  - B) security and identity services
  - C) computer services
  - D) network infrastructure virtualization
  - E) voice and collaboration services
- Q2) Which of the architectures within Cisco Enterprise Architectures model is the most appropriate for deploying integrated services as implemented in Cisco ISRs? (Source: Describing Network Requirements)
- A) Campus
  - B) Data Center
  - C) Branch
  - D) Teleworker
  - E) WAN
- Q3) Which two Cisco features solve the performance and security requirements of the converged enterprise networks in a simple yet efficient manner? (Choose two.) (Source: Describing Network Requirements)
- A) Cisco SRST
  - B) Cisco AutoQoS
  - C) CDP
  - D) AutoSecure
  - E) 802.1x

## Module Self-Check Answer Key

- Q1) B
- Q2) C
- Q3) B, D



# Describe Cisco VoIP Implementations

---

## Overview

This module lists the advantages of converged networks and discusses the principles of Cisco VoIP implementations. The module describes the process of digitizing voice and IP encapsulation and provides information about bandwidth consumption and various aspects of voice network implementation.

## Module Objectives

Upon completing this module, you will be able to describe Cisco VoIP implementations. This ability includes being able to meet these objectives:

- Describe basic principles of VoIP networks
- Describe the process by which voice is digitized and packetized for transport on a data network
- Explain the encapsulation of voice into IP packets
- List the bandwidth requirements for various codecs and data links, and, given the formula to calculate total bandwidth for a VoIP call, list the methods to reduce bandwidth consumption
- Understand various aspects of voice network implementation



# Introducing VoIP Networks

---

## Overview

Converged networks allow voice and data to be transmitted over the same network. When operating VoIP networks, you should understand some principles of VoIP networks and know the components and their roles. This lesson describes basic principles of VoIP networks, and explains various call control methods.

## Objectives

Upon completing this lesson, you will be able to describe basic principles of VoIP networks. This ability includes being able to meet these objectives:

- Explain the benefits of VoIP compared to traditional circuit-switched telephony
- Describe the components of a VoIP network
- Describe analog connectivity options for legacy equipment to connect to a VoIP network
- Describe digital interface options to connect VoIP equipment to PBXs or the PSTN
- Describe the three stages of a call
- Compare the concept of distributed call control, where a voice gateway provides call control functions, to that of centralized call control, where the call control process is run by a call agent, such as Cisco Unified CallManager

# Benefits of Packet Telephony Networks

This topic explains the benefits of VoIP compared to traditional circuit-switched telephony.

## Benefits of Packet Telephony Networks

- **More efficient use of bandwidth and equipment**
- **Lower transmission costs**
- **Consolidated network expenses**
- **Improved employee productivity through features provided by IP telephony:**
  - **IP phones are complete business communication devices**
    - **Directory lookups and database applications (XML)**
    - **Integration of telephony into any business application**
  - **Software-based and wireless phones offer mobility.**
- **Access to new communications devices (such as, PDAs and cable set-top boxes)**

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0-2.3

Most modern companies are using converged networks, serving both data and telephony with a single IP network infrastructure. The benefits of packet telephony networks include these:

- **More efficient use of bandwidth and equipment:** Traditional telephony networks use a 64-kbps channel for every voice call. Packet telephony shares bandwidth among multiple logical connections.
- **Lower transmission costs:** A substantial amount of equipment is needed to combine 64-kbps channels into high-speed links for transport across the network. Packet telephony statistically multiplexes voice traffic alongside data traffic. This consolidation provides substantial savings on capital equipment and operations costs.
- **Consolidated network expenses:** Instead of operating separate networks for voice and data, voice networks are converted to use the packet-switched architecture to create a single integrated communications network with a common switching and transmission system. The benefit is significant cost savings on network equipment and operations.
- **Improved employee productivity through features provided by IP telephony:** IP phones are not only phones, they are complete business communication devices. They offer directory lookups and access to databases through Extensible Markup Language (XML) applications. These applications allow simple integration of telephony into any business application. For instance, employees can use the phone to look up information about a customer who called in, search for inventory information, and enter orders. The employee can be notified of a issue (for example, a change of the shipment date), and with a single click can call the customer about the change. In addition, software-based phones or wireless phones offer mobility to the phone user.

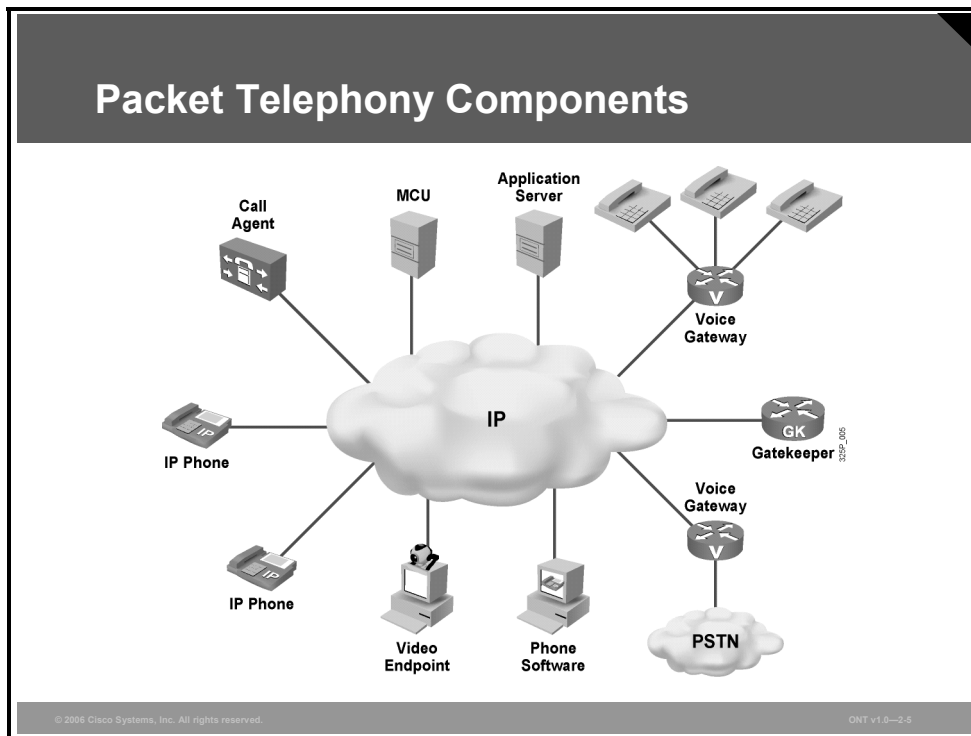
- **Access to new communications devices:** Packet technology can reach devices that are largely inaccessible to the modern time-division multiplexing (TDM) infrastructures. Examples of such devices are computers, wireless devices, household appliances, personal digital assistants (PDAs), and cable set-top boxes. Intelligent access to such devices enables companies and service providers to increase the volume of communications that they deliver, the breadth of services that they offer, and the number of subscribers that they serve. Packet technology, therefore, enables companies to market new devices, including videophones, multimedia terminals, and advanced IP phones.

Although packet technology has clear benefits, you should carefully consider these points before migrating to this technology:

- Return on investment (ROI), when based on the new system features, can be difficult to prove.
- Generally, voice and data personnel do not “speak the same language.”
- Current voice telephony components may not yet have fully depreciated.

# Packet Telephony Components

This topic introduces the basic components of a packet voice network.



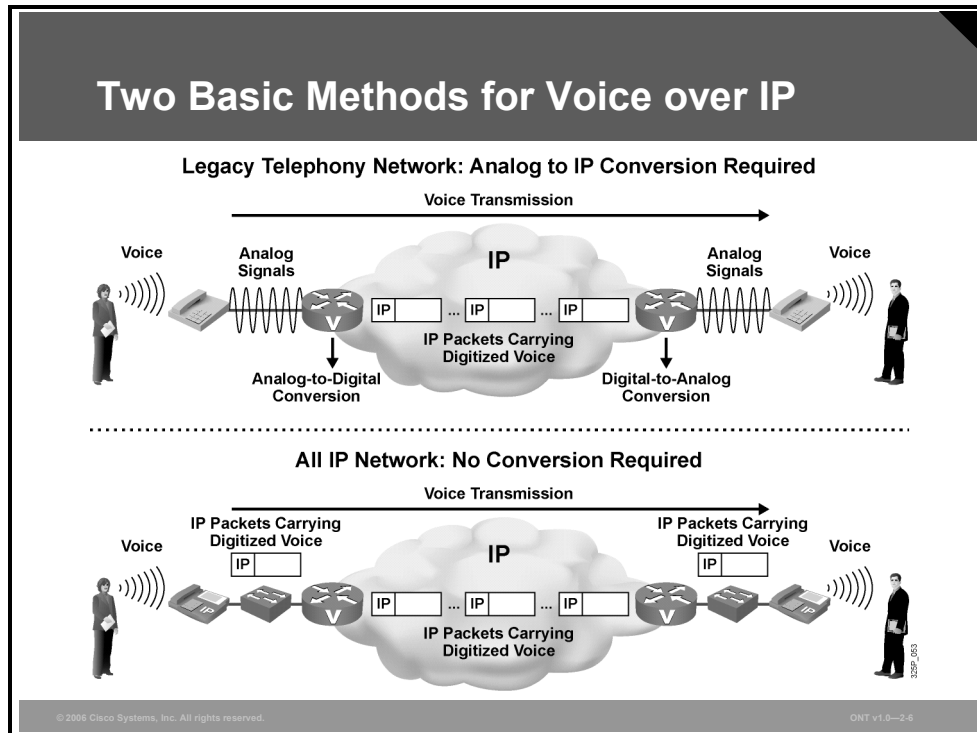
A packet telephony solution consists of several components:

- **Phones:** Phones provide telephony features to users. Phones can be IP phones, software-based phones operated on PCs, or traditional phones (analog or ISDN).
- **Gateways:** Gateways interconnect the packet telephony world with traditional telephony devices. These can be analog or ISDN phones, faxes, circuit-based PBX systems, or public switched telephone network (PSTN) switches.
- **Multipoint control units:** A multipoint control unit is required for conferences. If more than two parties are in a call, all members of the conference send their media to the multipoint control unit, where they are mixed and then sent back to all participants.
- **Application servers:** Application servers provide XML-based services to IP phones. IP phone users have access to directories and databases through XML applications.
- **Gatekeepers:** Gatekeepers provide Call Admission Control (CAC) and translate telephone numbers or names to IP addresses for call routing in an H.323 network.
- **Call agents:** Call agents provide call control, CAC, and bandwidth control and address translation services to IP phones or Media Gateway Control Protocol (MGCP) gateways.
- **Video endpoints:** Video endpoints provide video telephony features to users. As with audio-only calls, video calls need a multipoint control unit for conferences. For videoconferences, the multipoint control unit has to be capable of mixing video and audio streams.

In an IP telephony network, some functions rely on digital signal processors (DSPs). DSPs are used for converting analog voice signals into digital format and vice versa. They also provide functions such as voice compression, transcoding (changing between different formats of digitized voice), and conferencing. DSPs are hardware components located on voice modules inside gateways.

## Two Basic Methods for VoIP

A VoIP network can include legacy devices that require analog-to-IP conversion, or it can be an all-IP network, as shown in the figure.

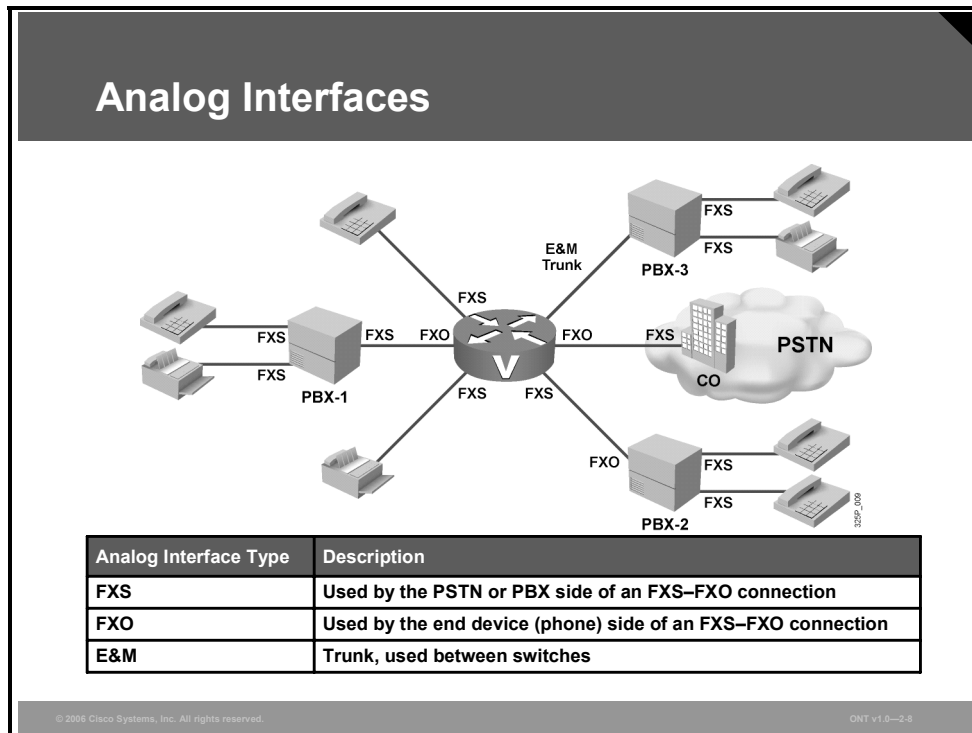


In a VoIP network that includes legacy equipment, such as analog phones, gateways are needed that convert the analog signals into digital format and encapsulate them into IP packets.

If all devices in the network are IP enabled, no conversion is needed.

# Analog Interfaces

This topic defines three analog interfaces (Foreign Exchange Station [FXS], Foreign Exchange Office [FXO], and ear and mouth [E&M]) and discusses how each of these interfaces is used.



Gateways use different types of interfaces to connect to analog devices, such as phones, fax machines, or PBX or public switched telephone network (PSTN) switches. Analog interfaces used at the gateways include these three types:

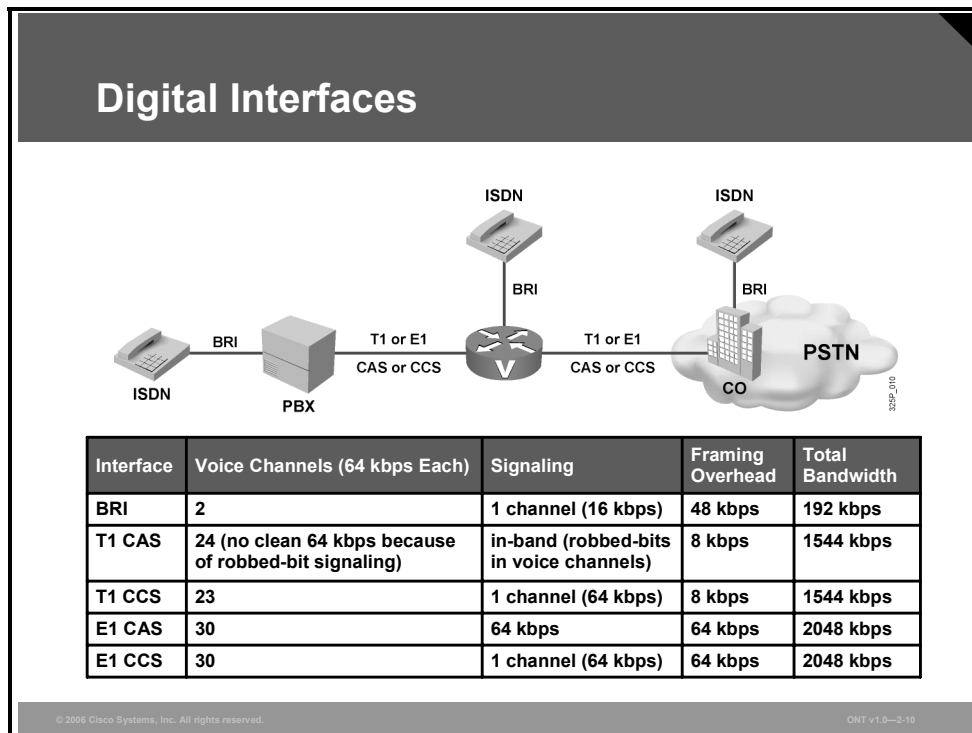
- **FXS:** The FXS interface connects to analog end systems, such as analog phones or analog faxes, which on their side use the FXO interface. The router FXS interface behaves like a PSTN or a PBX, serving phones, answering machines, or fax machines with line power, ring voltage, and dial tones. If a PBX uses an FXO interface, it can also connect to a router FXS interface. In this case, the PBX acts like a phone.
- **FXO:** The FXO interface connects to analog systems, such as a PSTN or a PBX, which on their side use the FXS interface. The router FXO interface behaves like a phone, getting line power, ring voltage, and dial tones from the other side. As mentioned, a PBX can also use an FXO interface toward the router (which will then use an FXS interface), if the PBX takes the role of the phone.
- **E&M:** The E&M interface provides signaling for analog trunks. Analog trunks interconnect two PBX-style devices, such as any combination of a gateway (acting as a PBX), a PBX, and a PSTN switch. E&M is often defined to as “ear and mouth,” but it derives from the term “earth and magneto.” “Earth” represents the electrical ground, and “magneto” represents the electromagnet used to generate tones.



In the figure, the gateway serves a phone and a fax machine using two FXS interfaces. For these two devices, the router acts like a PBX or a PSTN switch. The router connects to the PSTN using an FXO interface. For this connection, the router acts like a phone toward the PSTN. Another FXO interface is used to connect to a PBX (PBX-1). Again, the router acts like an end system toward the PBX, and hence uses the same port type as the phone and fax, which are also connected to PBX-1. A second PBX (PBX-2) connects to the router FXS interface. For this connection, it is the PBX that behaves like a phone toward the router, which acts like a PSTN switch. Finally, the router connects to another PBX (PBX-3), this time using an E&M interface. On this trunk connection, both the router and PBX-3 act as a PBX.

# Digital Interfaces

This topic describes channel associated signaling (CAS), and basic rate and PRIs.



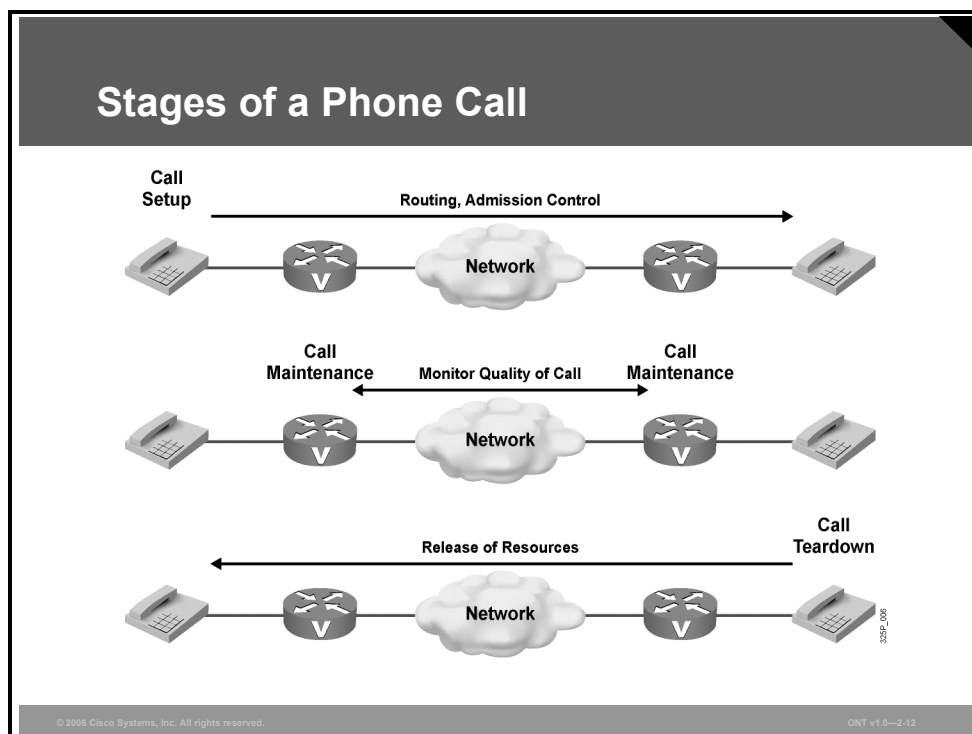
Gateways can use digital interfaces to connect to voice equipment. From a hardware perspective, there are BRIs and T1 and E1 interfaces available. All of them use TDM to support multiple logical channels. T1 and E1 interfaces can use either CAS or common channel signaling (CCS), while a BRI always uses CCS. ISDN PRIs use T1 or E1 CCS.

The figure illustrates a router that serves an ISDN phone using a BRI voice interface. In addition, the router has two T1 or E1 lines: one to a PBX and one to the PSTN. Depending on the interface type (T1 or E1) and the signaling method (CAS versus CCS), a maximum of 23, 24, or 30 voice channels will be available on these trunks. The PSTN and the PBX also serve ISDN phones through BRI connections.

The table shows the number of voice channels, the signaling bandwidth, and the total bandwidth, considering the framing overhead, for all available interface and signaling options.

# Stages of a Phone Call

Call control allows phones to establish, maintain, and disconnect a voice flow across a network. This topic describes the stages of a phone call.

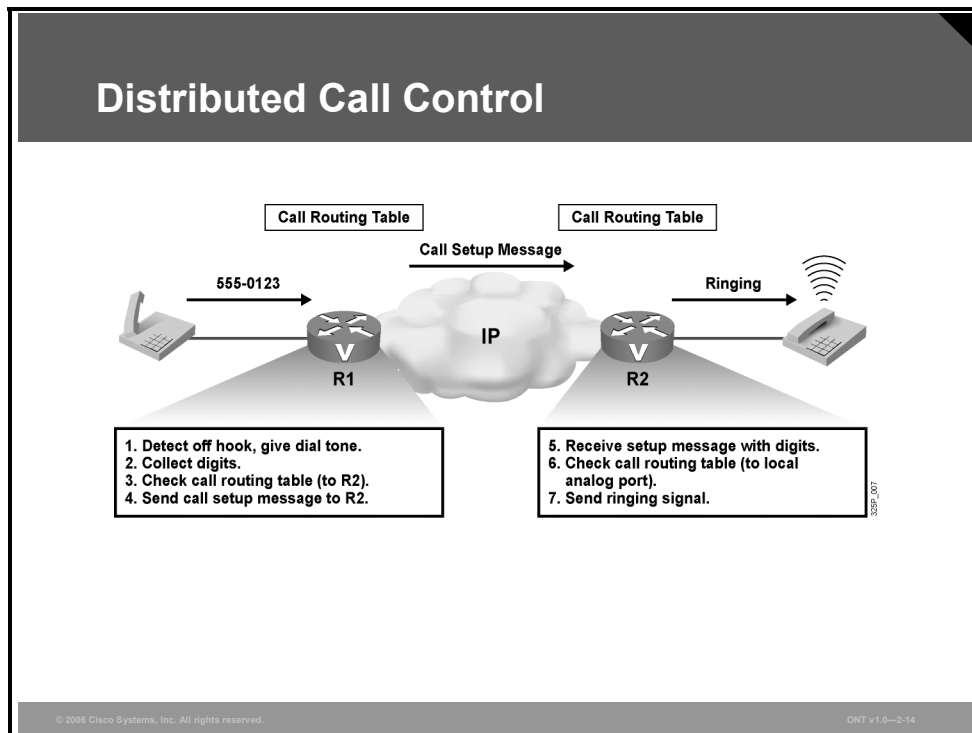


Although different protocols address call control in different ways, they all provide a common set of services. The basic components of call control are these:

- **Call setup:** Call setup checks the call-routing configuration to determine the destination of a call. The configuration specifies the bandwidth requirements for the call. When the bandwidth requirements are known, CAC determines whether sufficient bandwidth is available to support the call. If bandwidth is available, call setup generates a setup message and sends it to the destination. If bandwidth is not available, call setup notifies the initiator by presenting a busy signal. Different call control protocols, such as H.323, MGCP, and session initiation protocol (SIP), define different sets of messages to be exchanged during setup. However, all of them negotiate the same basic information:
  - The IP addresses of the two devices that will exchange VoIP
  - The User Datagram Protocol (UDP) port numbers that will be used for the Real-Time Transport Protocol (RTP) streams
  - The format (for example, the compression algorithm) used for the digitized voice
- **Call maintenance:** Call maintenance tracks packet count, packet loss, jitter, and delay during the call. Information passes to the voice-enabled devices to determine whether connection quality is good or has deteriorated to the point where the call should be dropped.
- **Call teardown:** Call teardown notifies voice-enabled devices to free resources and make them available for other calls when either side terminates a call.

# Distributed vs. Centralized Call Control

This topic compares distributed and centralized call control.



There are two types of call control: distributed and centralized.

## Distributed Call Control

The figure shows an environment where call control is handled by multiple components in the network. Distributed call control is possible where the voice-capable device is configured to support call control directly. This is the case when protocols such as H.323 or SIP are enabled on the end devices. With distributed call control, the devices perform the call setup, call maintenance, and call teardown on their own:

- **Process dialed digits and route the call:** The figure shows an example of this sequence.
  1. After detecting a service request (the phone goes off hook), the first gateway (R1) plays a dial tone.
  2. Next, R1 collects the digits dialed by the caller.
  3. R1 then looks up the called number in its local call-routing table. According to the call routing table, the called number can be reached via the second gateway (R2).
  4. R1 now enters the first stage of a call, call setup, by sending the appropriate message to R2.
  5. R2 receives the call setup message from R1.
  6. R2 then looks up the called number in its local call routing table. According to the call routing table, the called number can be reached on a local voice port.
  7. Therefore, R2 sends the call to that port by applying the ring voltage.

In this example of the distributed call control model, R1 made a local decision to send the call setup message to R2 based on the call routing table of R1. R2 again made a local decision (using its call routing table) that the called device could be reached on a certain physical port.

- **Supervise the call:** During the call, R1 and R2 both monitor the quality of the call. In the distributed call control model, if one of the gateways detects that the quality is no longer acceptable, it locally terminates the call.

Supervision of the call occurs during the second stage of the call, call maintenance.

- **Terminate the call:** If the caller that is connected to R1 finishes the call, R1 informs R2 that the VoIP call should be terminated and resources should be released.

In the distributed call control model, the gateway itself initiates the third stage of a call, call teardown.

As illustrated in the example, with distributed call control, each gateway makes its own, autonomous decisions and does not depend on the availability of another (centralized) device to provide call routing services to the gateway. Because each gateway has its own intelligence, there is no single point of failure. However, each gateway needs to have a local call routing table, which has to be configured manually. Therefore, administration of the distributed call control model is less scalable.

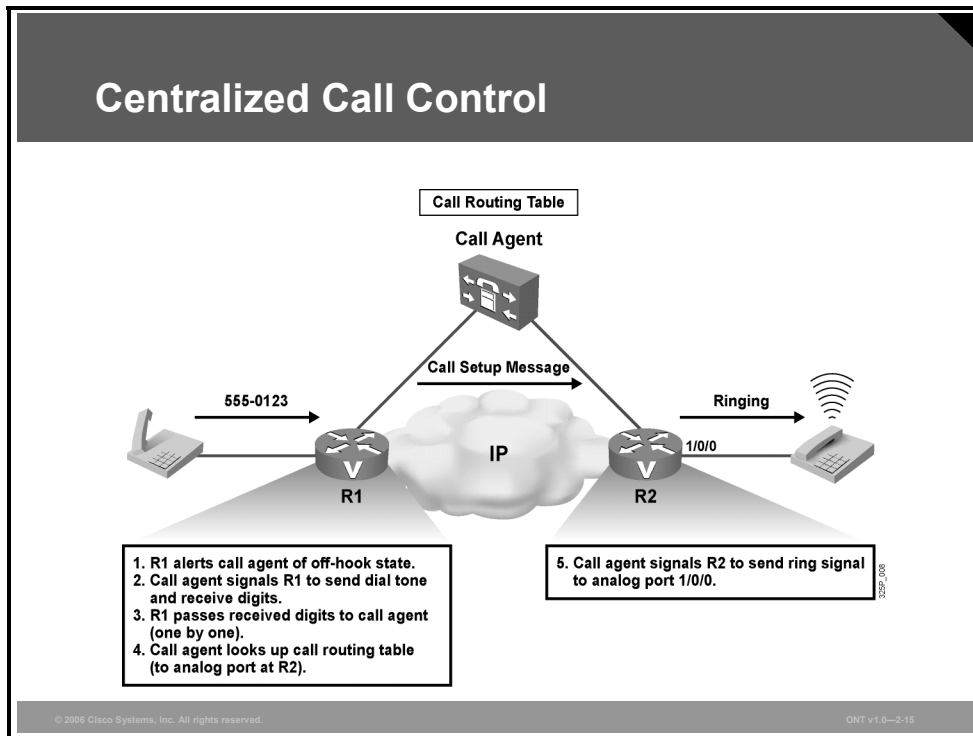
---

<b>Note</b>	For larger deployments using the distributed call control model, special devices can be added for centralized number lookup. H.323 gatekeepers or SIP network servers can be used by a gateway or endpoint to find numbers that are not known locally. In such a deployment, there is no need for all (or even any) numbers to be stored at the gateways or endpoints but only at the centralized devices.
-------------	--

---

# Centralized Call Control

The figure shows an environment where call control is handled by a single component in the network, a call agent.



Centralized call control applies when the voice-capable device does not support call control on its own but only through the use of a call agent. In this example, both voice gateways have the MGCP protocol enabled. With MGCP call control enabled, the gateways use the call agent to perform these functions:

- **Process dialed digits and route the call:** The figure shows an example of this sequence.
  1. After detecting a service request (the phone goes off hook), the first gateway (R1) informs its call agent.
  2. The call agent tells R1 to play a dial tone and receive the digits dialed by the user.
  3. R1 passes each received digit to the call agent (one by one).
  4. The call agent looks up the called number in its call routing table. According to the call routing table, the called number can be reached via the second gateway (R2). R2 is also controlled by this call agent and the call agent, therefore, knows about the phone numbers that can be reached by R2. As a consequence, the call agent knows which port at R2 the call has to be routed to.
  5. The call agent now sends a message to R2 requesting that the call be passed to a certain port (the port that connects to the destination phone number).

Both call routing decisions—which gateway to use after the call has been received at R1 and how to pass the call on at that next gateway (R2)—are made by the call agent. This is an example of the centralized call control model, where all call routing intelligence (required for the first stage of a call, the call setup) is at the call agent. The call agent then instructs the gateways how to handle the call. Therefore, only the call agent has a call routing table.

- **Supervise the call:** During the call, R1 and R2 both monitor the quality of the call. In the centralized call control model, if one of the gateways detects that the quality is no longer adequate, it will pass that information to the call agent. The call agent then terminates the call.

Supervision of the call occurs during the second stage of a call, call maintenance.

- **Terminate the call:** If the caller that is connected to R1 finishes the call, R1 informs the call agent. The call agent notifies both gateways that the VoIP call should be terminated and resources should be released.

In the centralized call control model, the call agent initiates the third stage of a call, call teardown.

As illustrated in the example, with centralized call control, the gateways do not make any local decisions. Instead, they inform the call agent about events (such as incoming or dropped calls). Only the call agent makes call routing decisions, and the gateways depend on the availability of their call agent. Availability of the call agent is critical, because the call agent is a single point of failure. However, only the call agent needs to have a call routing table. Therefore, administration of the centralized call control model is more scalable.

Centralized call control allows an external device (call agent) to handle signaling and call processing, leaving the gateway to translate audio signals into voice packets after call setup. After the call is set up, the voice path runs directly between the two gateways and does not involve the call agent. The difference between distributed and centralized call control applies only to signaling, never to the media exchange, which always goes on directly between the two gateways.

# Summary

This topic summarizes the key points that were discussed in this lesson.

## Summary

- Companies can benefit from a common infrastructure that serves voice and data. Advantages of such converged networks include lower costs, more efficient use of available bandwidth, and higher productivity.
- A packet telephony network consists of endpoints (such as IP phones, software phones, and video endpoints) and voice network devices (such as gateways, gatekeepers, conference bridges, call agents, and application servers).
- A voice gateway can use FXS, FXO, and E&M interfaces to connect to analog equipment, such as phones, PBXs, or the PSTN.
- A voice gateway can use BRI, T1, and E1 interfaces to connect to digital equipment, such as ISDN phones, PBXs, or the PSTN.
- A voice call consists of three stages: call setup, call maintenance, and call teardown.
- With distributed call control, each gateway has local intelligence to route calls, while with centralized call control, a call agent makes call routing decisions on behalf of all the gateways that are controlled by the call agent.

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—2-16



# Digitizing and Packetizing Voice

---

## Overview

Before voice can be sent over IP networks, it has to be digitized and encapsulated into IP packets. This lesson discusses in detail how analog signals are converted into digital format and describes various voice compression algorithms that can be used to reduce bandwidth requirements.

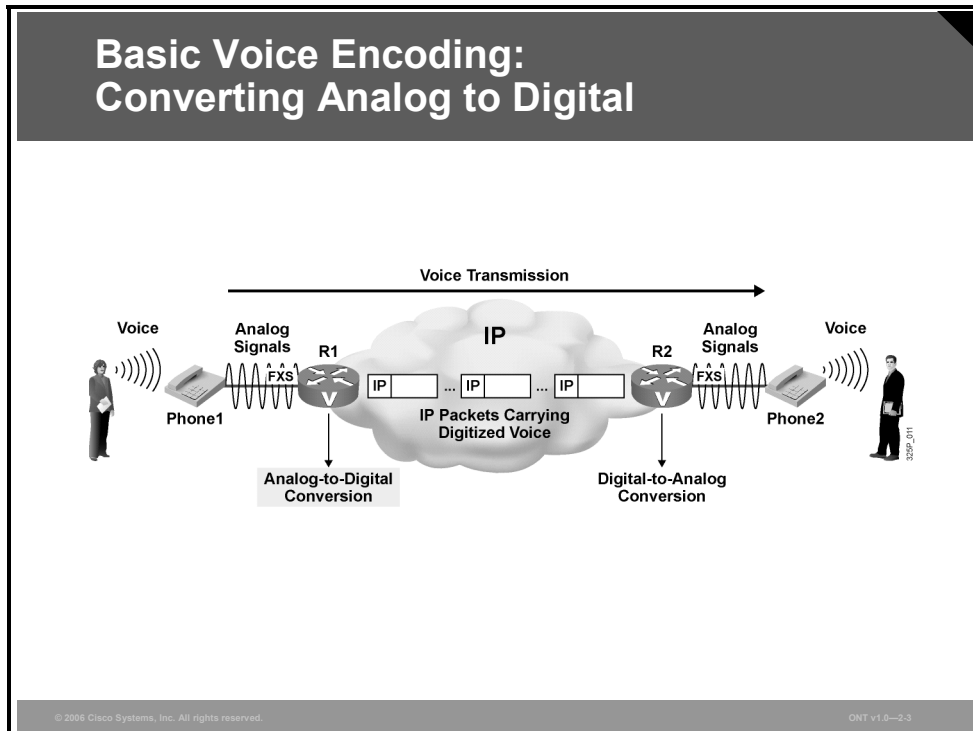
## Objectives

Upon completing this lesson, you will be able to describe the process by which voice is digitized and packetized for transport on a data network. This ability includes being able to meet these objectives:

- Identify the steps for converting analog signals to digital signals
- Identify the steps for converting digital signals to analog signals
- Explain why voice is sampled at 8000 bps for telephone calls
- Explain how a signal is quantized and processed according to the Nyquist theorem to yield a standard voice channel bit rate of 64 kbps
- List the common voice compression standards, noting their bandwidth requirements and voice quality measurement
- Describe the purpose of a DSP in a voice gateway

# Basic Voice Encoding: Converting Analog to Digital

This topic describes the process of converting analog signals to digital signals.



When voice is transported over an IP network, analog voice signals first have to be converted to digital format so that they can be encapsulated into IP packets.

The figure illustrates a call from an analog phone (Phone1) connected to a router (R1) and to an analog phone (Phone2) connected to another router (R2). The two routers are connected to an IP network. The user at Phone1 speaks into the microphone of the phone, and the phone sends these analog signals to the Foreign Exchange Station (FXS) port of router R1. Router R1 converts the received analog signal to digital and encapsulates the bits into IP packets. These IP packets are then sent to router R2 over the IP network.

# Analog-to-Digital Conversion Steps

When a router converts analog signals to digital signals, it performs several steps.

## Analog-to-Digital Conversion Steps

1. **Sample the analog signal.**
2. **Quantize the samples.**
3. **Encode the value into a binary expression.**
4. **(Optional) Compress the samples to reduce bandwidth.**

© 2006 Cisco Systems, Inc. All rights reserved. OMT v1.0—2-4

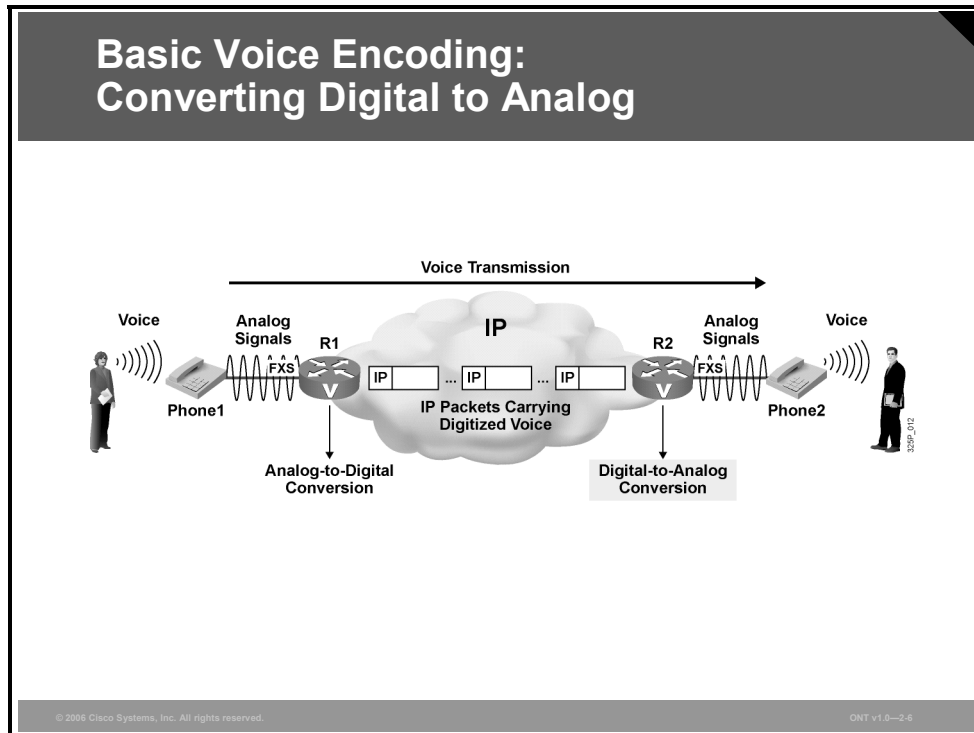
Analog to digital conversion steps include these:

- Step 1     Sampling:** The analog signal is sampled periodically. The output of the sampling is a pulse amplitude modulation (PAM) signal.
- Step 2     Quantization:** The PAM signal is matched to a segmented scale. This scale measures the amplitude (height) of the PAM signal.
- Step 3     Encoding:** The matched scale value is represented in binary format.
- Step 4     Compression:** Optionally, voice samples can be compressed to reduce bandwidth requirements.

Analog-to-digital conversion is done by digital signal processors (DSPs), which are located on the voice interface cards. The conversion is needed for calls received on analog lines, which are then sent out to a packet network or to a digital voice interface.

# Basic Voice Encoding: Converting Digital to Analog

This topic describes the process of converting digital signals back to analog signals.



When a router receives voice in digital format, it has to convert it back to analog signals before sending it out to analog voice interfaces.

The figure illustrates a call from an analog phone (Phone1) connected to a router (R1) and to an analog phone (Phone2) connected to another router (R2). The two routers are connected to an IP network. When router R2 receives the IP packets carrying digitized voice, it converts them back to analog signals. The analog signals are then sent to Phone2 where they are played by the speaker of the phone, so that the user at Phone2 can hear the original speech.

# Digital-to-Analog Conversion Steps

When a router converts digital signals to analog signals, it performs several steps.

## Digital-to-Analog Conversion Steps

1. **Decompress the samples, if compressed.**
2. **Decode the samples into voltage amplitudes, rebuilding the PAM signal.**
3. **Reconstruct the analog signal from PAM signals.**

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—2-7

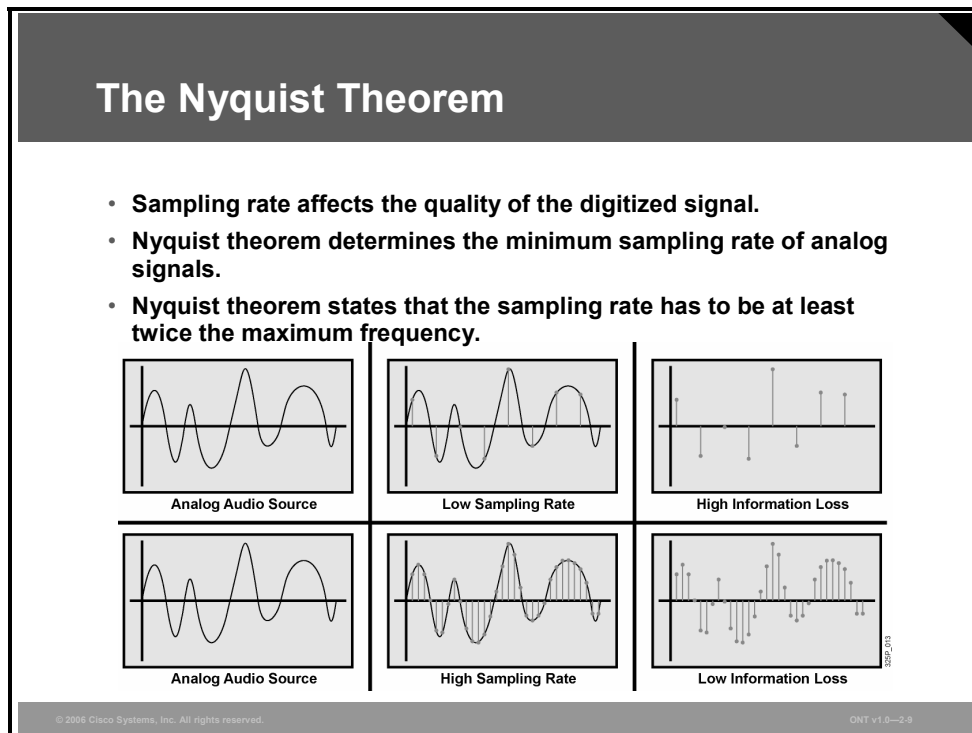
Digital-to-analog conversion steps include these:

- Step 1     Decompression:** If the voice signal was compressed by the sender, it is first decompressed.
- Step 2     Decoding:** The received, binary formatted voice samples are decoded to the amplitude value of the samples. This information is used to rebuild a PAM signal of the original amplitude.
- Step 3     Reconstruction of the analog signal:** The PAM signal is passed through a properly designed filter that reconstructs the original analog wave form from its digitally coded counterpart.

The whole process is simply the reverse of the analog-to-digital conversion. Like analog-to-digital conversion, digital-to-analog conversion is performed by DSPs, which are located on the voice interface cards. The conversion is needed for calls being received from a packet network or digital interfaces, which are then transmitted out an analog voice interface.

# The Nyquist Theorem

This topic describes the Nyquist theorem, which is the basis for digital signal technology.



When analog signals are converted to digital, the analog signal is first sampled. The sampling rate has impact on the quality of the digitized signal. If the sampling rate is too low, not enough information about the original analog signal is maintained, which results in degradation of quality.

Analog-to-digital conversion is based on the premise of the Nyquist theorem. The Nyquist theorem states that when a signal is instantaneously sampled at regular intervals and at a rate of at least twice the highest channel frequency, then the samples will contain sufficient information to allow an accurate reconstruction of the signal at the receiver.

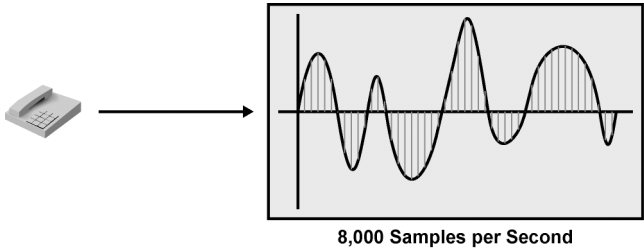
The figure illustrates two situations. In the first one, the analog signal is sampled at too low a sampling rate, resulting in imprecise information. The original waveform can hardly be derived from the obtained PAM signals. In the second situation, a higher sampling rate is used and the resulting PAM signals represent the original waveform very well.

## Example: Sampling of Voice

According to the Nyquist theorem, different sampling rates have to be used for digitizing analog signals of different frequency ranges.

### Example: Sampling of Voice

- Human speech uses 200–9,000 Hz.
- Human ear can sense 20–20,000 Hz.
- Traditional telephony systems were designed for 300–3,400 Hz.
- Sampling rate for digitizing voice was set to 8,000 samples per second, allowing frequencies up to 4,000 Hz.



8,000 Samples per Second

© 2006 Cisco Systems, Inc. All rights reserved. OHT v1.0-2.10

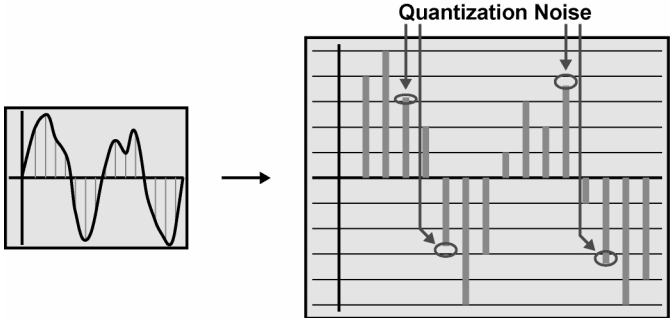
Although the human ear can sense sounds from 20 to 20,000 Hz, and speech encompasses sounds from about 200 to 9000 Hz, the telephone channel was designed to operate at about 300 to 3400 Hz. This economical range carries enough fidelity to allow callers to identify the party at the far end and sense their mood. To allow capturing of higher-frequency sounds that the telephone channel may deliver, the highest frequency for voice was set to 4000 Hz. Using the Nyquist theorem, the sampling rate results in 8000 samples per second, that is, one sample every 125 microseconds.

# Quantization

This topic explains quantization and its techniques.

## Quantization

- **Quantization is the representation of amplitudes by a certain value (step).**
- **Scale with 256 steps is used for quantization.**
- **Samples are rounded up or down to closer step.**
- **Rounding introduces inexactness (quantization noise).**



© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—2-12

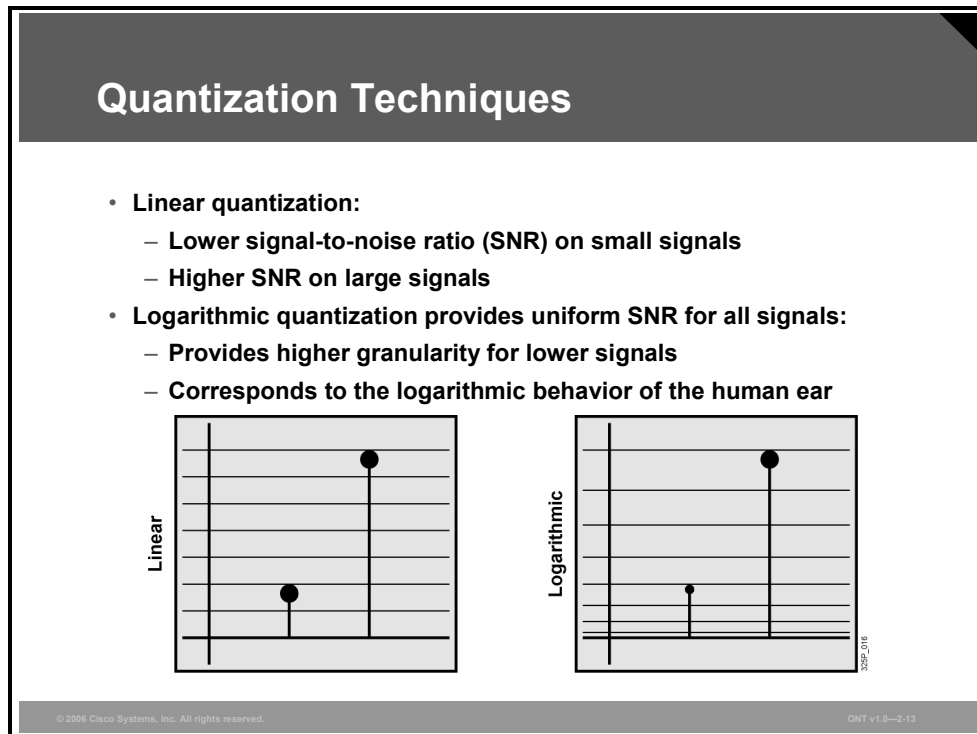
Quantization involves dividing the range of amplitude values that are present in an analog signal sample into a set of 256 discrete steps. The PAM values are rounded up or down to the step that is closest to the original analog signal. The difference between the original analog signal and the quantization level assigned is called *quantization error* or *quantization noise*, which is the source of distortion in digital transmission systems.

The figure depicts quantization. In this example, the x-axis is time and the y-axis is the voltage value (PAM). Quantization noise is indicated at all signals that do not exactly match one of the steps.



# Quantization Techniques

Quantization noise is a problem when a linear scale is used for quantization.



As illustrated in the figure, the signal-to-noise ratio (SNR) is very low on small signals and is higher on large signals, with a linear scale. Using a logarithmic scale provides better granularity for smaller signals, resulting in a uniform SNR for all signals.

The logarithmic segmentation also corresponds closely to the logarithmic behavior of the human ear.

## Example: Quantization of Voice

To avoid a low SNR, logarithmic quantization methods are common. This example shows methods used for voice signals in the telephony environment.

### Example: Quantization of Voice

- **There are two methods of quantization:**
  - **Mu-law**, used in Canada, U.S., and Japan
  - **A-law**, used in other countries
- **Both methods use a quasi-logarithmic scale:**
  - **Logarithmic segment sizes**
  - **Linear step sizes (within a segment)**
- **Both methods have eight positive and eight negative segments, with 16 steps per segment.**

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—2.14

The Bell system developed the mu-law method of quantization, which is widely used in North America and Japan. The ITU modified the mu-law method and created the a-law method, which is used in countries outside of North America and Japan.

Following the idea of allowing smaller step functions at lower amplitudes—rather than higher amplitudes—mu-law and a-law provide a quasi-logarithmic scale: The voltage range is divided into 16 segments (0 to 7 positive, and 0 to 7 negative). Each segment has 16 steps. Starting with segment 0, which is closest to zero amplitude, the segments get bigger toward the maximum amplitudes and the size of the steps increases. Within a segment, however, the size of the steps is linear.

The result of using mu-law and a-law is a more accurate value for smaller amplitudes and a uniform signal-to-noise quantization ratio (SQR) across the input range.

---

**Note** In the public switched telephone network (PSTN), for communication between a mu-law country and an a-law country, the mu-law country must change its signaling to accommodate the a-law country.

---

# Digital Voice Encoding

Digital voice samples are represented by 8 bits per sample.

## Digital Voice Encoding

- **Each sample is encoded using eight bits:**
  - One polarity bit
  - Three segment bits
  - Four step bits
- **Required bandwidth for one call is 64 kbps (8000 samples per second, 8 bits each).**
- **Circuit-based telephony networks use TDM to combine multiple 64-kbps channels (DS-0) to a single physical line.**

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—2-16

Each sample is encoded in the following way:

- **One polarity bit:** Indicates positive versus negative signals
- **Three segment bits:** Identify the logarithmically sized segment number (0–7)
- **Four step bits:** Identify the linear step within a segment

Because 8000 samples per second are taken for telephony, the bandwidth that is needed per call is 64 kbps. This is the reason why traditional, circuit-based telephony networks use time-division-multiplexed lines, combining multiple channels of 64 kbps each (digital signal level 0 [DS-0]) in a single physical interface.

# Compression Bandwidth Requirements

This topic lists the bandwidth requirements for various ITU compression standards.

Voice Codec Characteristics		
Standard, Codec	Bit Rate (kbps)	Voice Quality (MOS)
G.711, PCM	64	4.1
G.726, ADPCM	16, 24, 32	3.85 (with 32 kbps)
G.728, LDCELP	16	3.61
G.729, CS-ACELP	8	3.92
G.729A, CS-ACELP	8	3.9

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—2-17

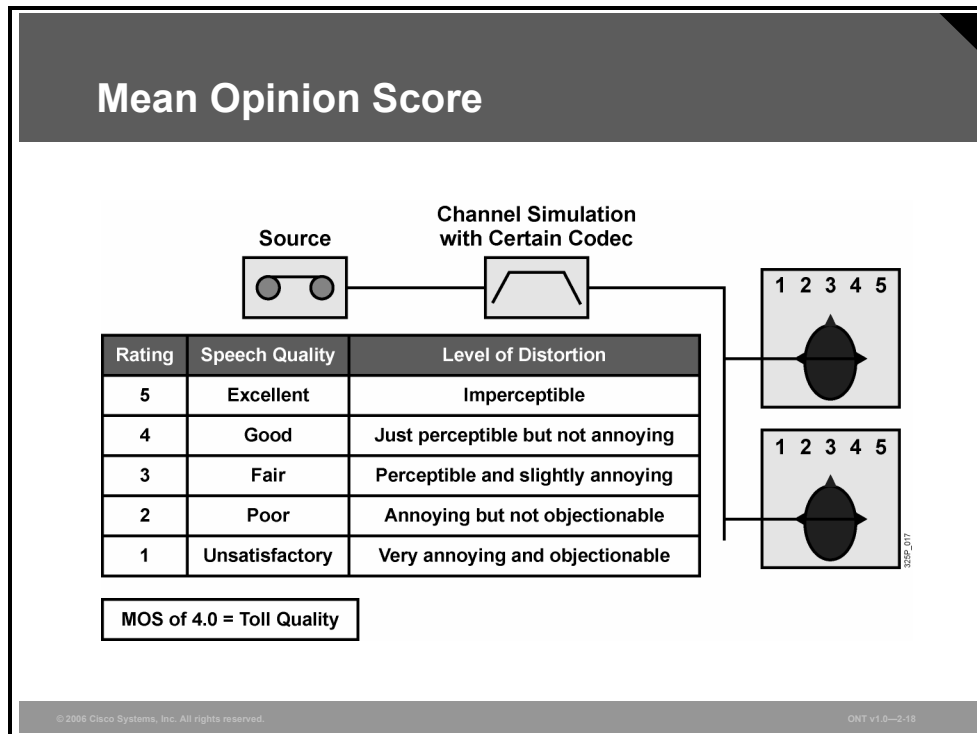
The most important characteristics of a codec are the required bandwidth and the quality degradation caused by the analog-to-digital conversion and compression.

The table shows several codecs, their bit rates, and the quality of the codec.

The quality of a codec can be measured by various means; the mean opinion score (MOS), used in this table, is a commonly used method.

# Mean Opinion Score

The MOS is a system of grading the voice quality of telephone connection using a codec.



The MOS is a statistical measurement of voice quality derived from the judgments of several subscribers. Graded by humans and, therefore, somewhat subjective, the range of the MOS is 1 to 5, where 5 is direct conversation. The table provides a description of the five ratings.

A newer, more objective measurement is quickly overtaking MOS scores as the industry quality measurement of choice for coding algorithms. Perceptual Speech Quality Measurement (PSQM), as per ITU standard P.861, provides a rating on a scale of 0 to 6.5, where 0 is best and 6.5 is worst.

PSQM is implemented in test equipment and monitoring systems that are available from vendors other than Cisco. Some PSQM test equipment converts the 0-to-6.5 scale to a 0-to-5 scale to correlate to MOS. PSQM works by comparing the transmitted speech to the original input and yielding a score. Test equipment from various vendors is now capable of providing a PSQM score for a test voice call over a particular packet network.

In 1998, British Telecom (BT) developed a predictive voice quality measurement algorithm called Perceptual Analysis Measurement System (PAMS). PAMS can predict the results of subjective speech quality measurement methods, such as MOS, when fidelity is affected by such things as waveform codecs, vocoders, and various speaker dependencies, such as language. PAMS, unlike PSQM, includes automatic normalization for levels.

ITU standard P.862 supersedes standard P.861 and describes a voice quality measurement technique that combines PSQM and PAMS. Originally developed by KPN Research (now TNO Telecom) and BT, Perceptual Evaluation of Speech Quality (PESQ) is an objective measuring tool that can predict the results of subjective measuring tests, such as MOS. PESQ can be found in test equipment from a variety of vendors.

# What Is a DSP?


This topic describes DSPs, their purpose in the voice gateway, and how they support the process of conferencing and transcoding.

## What Is a DSP?

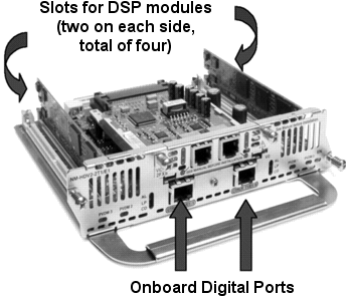
**A DSP is a specialized processor used for telephony applications:**

- **Voice termination:**
  - Converts analog voice into digital format (codec) and vice versa
  - Provides compression, echo cancellation, VAD, CNG, jitter removal, and so on
- **Conferencing:** Mixes incoming streams from multiple parties
- **Transcoding:** Translates between voice streams that use different, incompatible codecs

**DSP Module**



**Voice Network Module**



Slots for DSP modules  
(two on each side,  
total of four)

Onboard Digital Ports

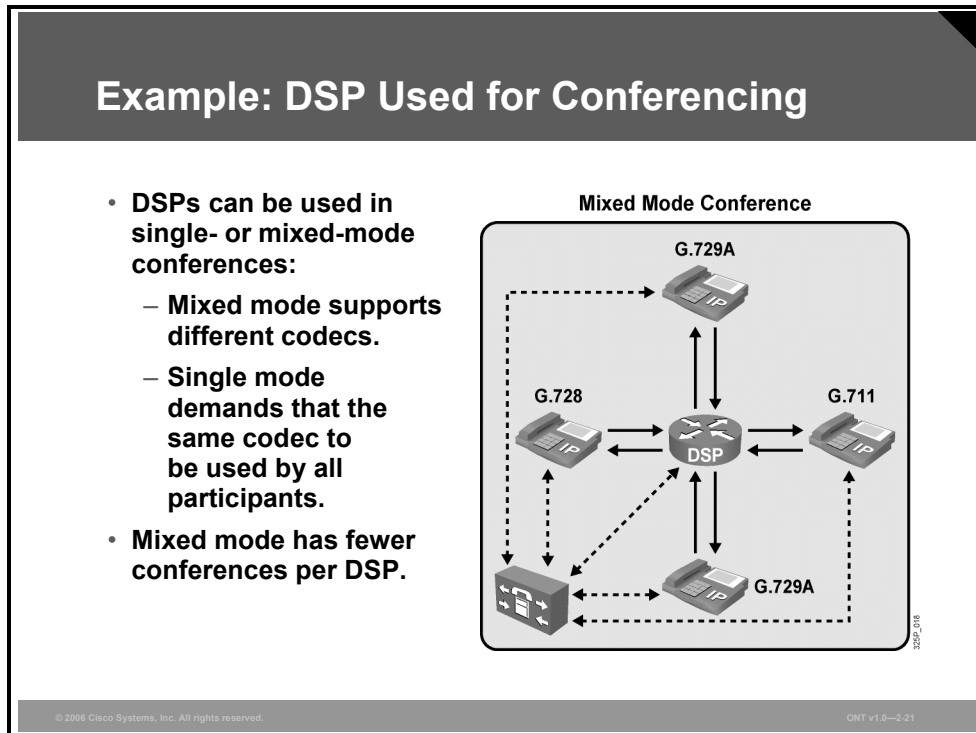
© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0-2.20

A digital signal processor (DSP) is a specialized processor used for telephony applications:

- **Voice termination:** DSPs are used to terminate calls to the gateway, received from or placed to traditional voice interfaces. These can be digital or analog interfaces. For example, when an analog phone places a call to the PSTN (over a digital trunk) or to a VoIP device, a DSP resource is used to accommodate this call. It converts the analog signal to digital (and vice versa for the reverse direction) and provides echo cancellation, compression, voice activity detection (VAD), comfort noise generation (CNG), jitter removal, and similar functions.
- **Conferencing:** In audio conferencing, DSPs are used to mix voice streams from multiple participants into a single conference call stream. Technically, all participants send their audio to the conference bridge (that is, the DSP), where the streams are mixed and then played back to all participants.
- **Transcoding:** DSP takes a voice stream of one codec type and converts it to another codec type. For example, transcoding takes a voice stream from a G.711 codec and transcodes it in real time to a G.729 codec stream.

## Example: DSP Used for Conferencing

When DSPs are used for conferencing, they can be used for two different types of conferences: mixed mode and single mode.

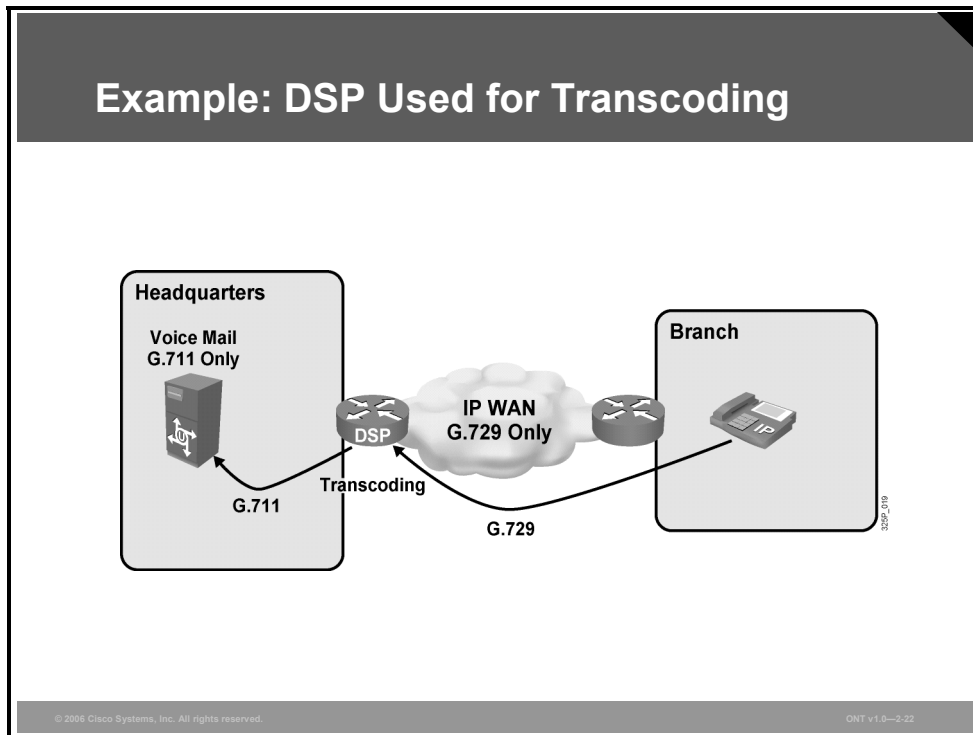


A DSP used for mixed-mode conferences allows the conference participants to use different codecs. In this case, the DSP not only mixes streams with the same codec type but can mix streams of different codecs. It also provides transcoding functions. Because of that additional functionality, mixed-mode conferences are more DSP intensive, and fewer conferences per DSP can be supported than in single mode.

A DSP used for single-mode conferences supports only one codec to be used by all conference participants. In this case, the DSP can mix streams with the same codec type only. If devices with different codecs would like to join the conference, separate DSPs have to be used for transcoding the codecs that are not compatible with the codec being used for the conference.

## Example: Transcoding Between Low-Bandwidth Codecs Used in the WAN and a Voice-Mail System Supporting Only G.711

Transcoding services are needed whenever two devices want to exchange voice information but use different codecs. As seen in the previous example, this can be the case if conference resources support only single-mode conferences but participants use different codecs.



In this example, a voice-mail system is located in the headquarters of a company. The voice-mail system is configured to use G.711 only. The company has a branch office that connects to the headquarters via an IP WAN. To conserve bandwidth, only G.729 is permitted over the WAN. If users from the branch access the voice-mail system, they can use only G.729 toward the headquarters, but the voice-mail system requires G.711. DSPs in the headquarters router are used to provide transcoding services to solve that problem. Calls from the branch to the voice-mail system set up a G.729 stream to the transcoding device (headquarters router), which transcodes the received G.729 stream into a G.711 stream toward the voice-mail system.



# Summary

This topic summarizes the key points that were discussed in this lesson.

## Summary

- **Whenever voice should be digitally transmitted, analog voice signals first have to be converted into digital. Conversion includes sampling, quantization, and encoding.**
- **Digitized voice has to be converted back to analog signals before being played out. Digital-to-analog conversion includes decoding and reconstruction of analog signals.**
- **The Nyquist theorem states the necessary sampling rate when converting analog signals to digital.**
- **Quantization is the process of representing the amplitude of a sampled signal by a binary number.**
- **Available codecs differ in their bandwidth requirements and voice quality.**
- **DSPs provide functions for call termination, conferences, and transcoding.**

© 2006 Cisco Systems, Inc. All rights reserved.

QNT v1.0—2.23



# Encapsulating Voice Packets for Transport

---

## Overview

Digitized voice has to be encapsulated into IP packets so that the VoIP packets can be sent across the IP network. This lesson explains how voice is encapsulated and discusses the purpose of the protocols that are used.

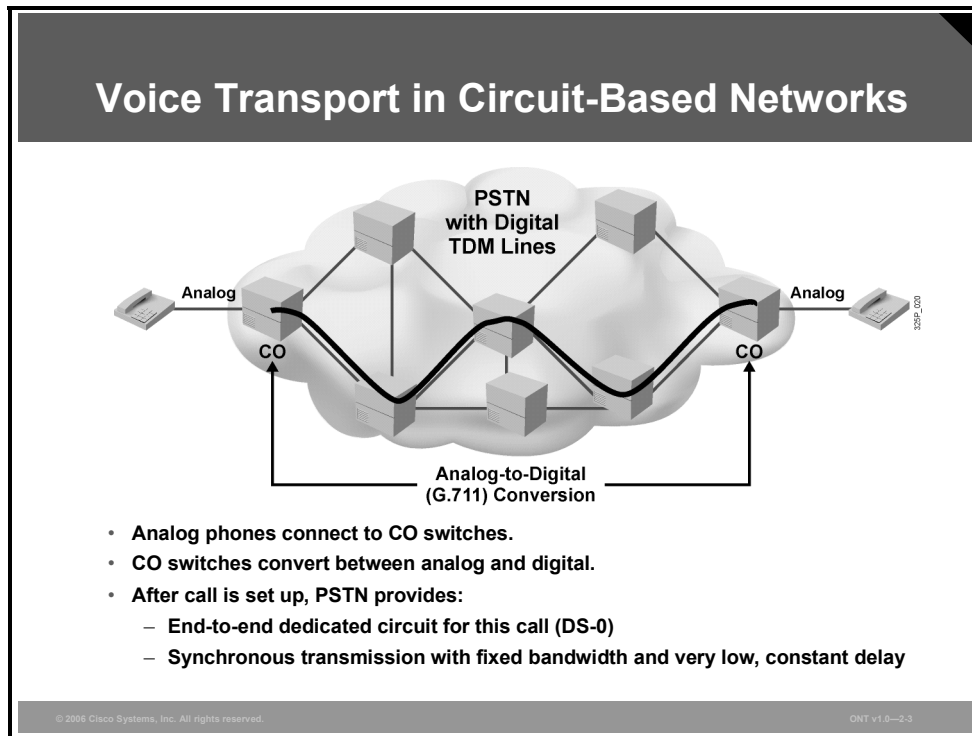
## Objectives

Upon completing this lesson, you will be able to explain encapsulation of voice into IP packets. This ability includes being able to meet these objectives:

- Explain how digitized voice packets are transported across a network in an RTP voice bearer stream
- Explain the purpose of RTP and UDP in packetizing and encapsulating voice for transport across a network
- Explain how and when to reduce header overhead with cRTP

# End-to-End Delivery of Voice Packets

This topic explains how digitized voice packets are transported across a network in a Real-Time Transport Protocol (RTP) voice bearer stream.



## Voice Transport in Circuit-Based Networks

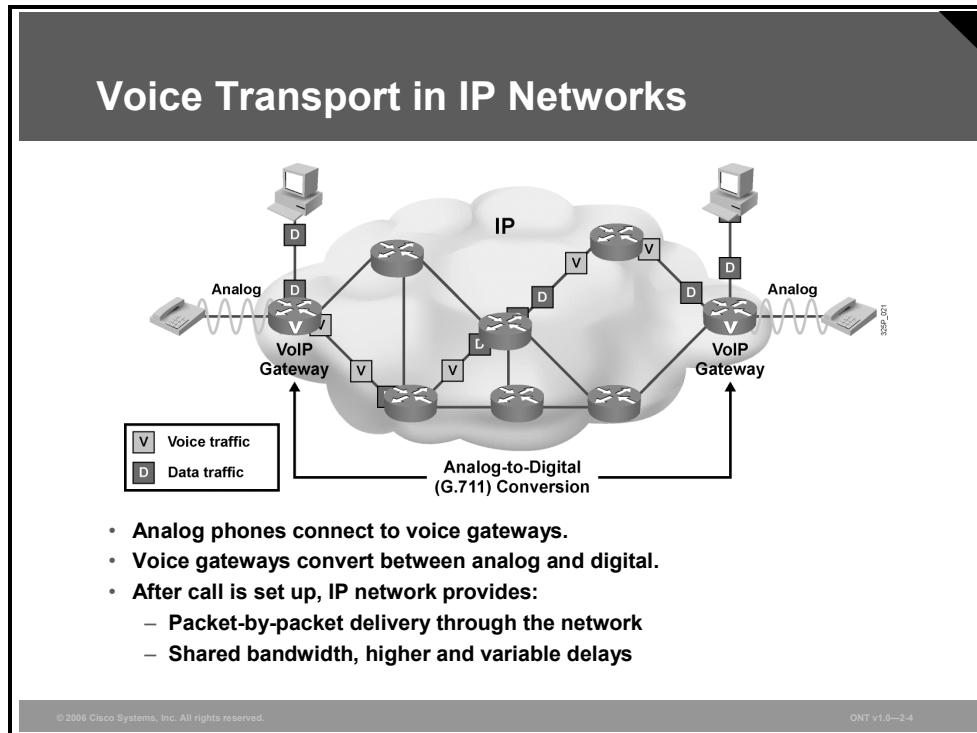
In traditional telephony networks, such as the public switched telephone network (PSTN), residential phones connect to central office (CO) switches via analog circuits. The core network is built of switches interconnected by digital trunks, as illustrated in the figure.

If a call is placed between two phones, the call setup stage occurs first. As a result of this process, an end-to-end dedicated circuit (digital signal level 0 [DS-0]) is created for the call. The CO switch then converts the received analog signals into digital format using the G.711 codec.

During the transmission stage, because of the synchronous transmission, the G.711 bits are sent at a fixed rate with very low and constant delay. The whole bandwidth of the circuit (64 kbps) is dedicated to the call, and because all bits follow the same path, all voice samples stay in order. When the call has finished, the individual DS-0 circuits are released and can be used for another call.

# Voice Transport in IP Networks

In IP packet telephony networks, analog phones connect to VoIP gateways through analog interfaces. The gateways are connected via an IP network, as illustrated in the figure.



When a call is placed between two phones, the call setup stage occurs first. As a result of this process, the call is logically set up, but no dedicated circuits (lines) are associated with the call. The gateway then converts the received analog signals into digital format using a codec, such as G.711 or G.729 if voice compression is being used.

During the transmission stage, voice is encapsulated into IP packets and sent, packet by packet, out to the network. The bandwidths of the links between the individual routers are not time-division multiplexed into separate circuits, but are single high-bandwidth circuits, carrying IP packets from several devices. As shown in the figure, data and voice packets share the same path and the same links. Although voice packets enter the network at a constant rate (which is lower than the physical line speed, leaving space for other packets), they may arrive at their destination at varying rates. Each packet encounters different delays on the route to the destination, and packets may even take different routes to the destination. The condition where packets arrive at varying, unpredictable rates is called *jitter*. For voice to be played back accurately, the destination router must both reinsert the correct time intervals and ensure that packets are in the correct order. After the call has finished, the call is logically torn down, and the gateway stops sending voice packets onto the network.

At any time when no voice packets are being sent, bandwidth is available for other applications, such as file transfers.

# Explaining Protocols Used in Voice Encapsulation

This topic explains the purpose of RTP and User Datagram Protocol (UDP) in packetizing and encapsulating voice for transport across a network.

Which Protocols to Use for VoIP?				
Feature	Voice Needs	TCP	UDP	RTP
Reliability	No	Yes	No	No
Reordering	Yes	Yes	No	Yes
Time-stamping	Yes	No	No	Yes
Multiplexing	Yes	Yes	Yes	No

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—2.6

When a VoIP device, such as a gateway, sends voice over an IP network, the digitized voice has to be encapsulated into an IP packet. Voice transmission requires features not provided by the IP protocol header; therefore, additional transport protocols have to be used. Transport protocols that include features required for voice transmission are TCP, UDP, and RTP. VoIP utilizes a combination of UDP and RTP. For a better understanding why these two protocols are used, it is important to understand the features provided by each protocol, as well as the relevance of those features to voice transmission:

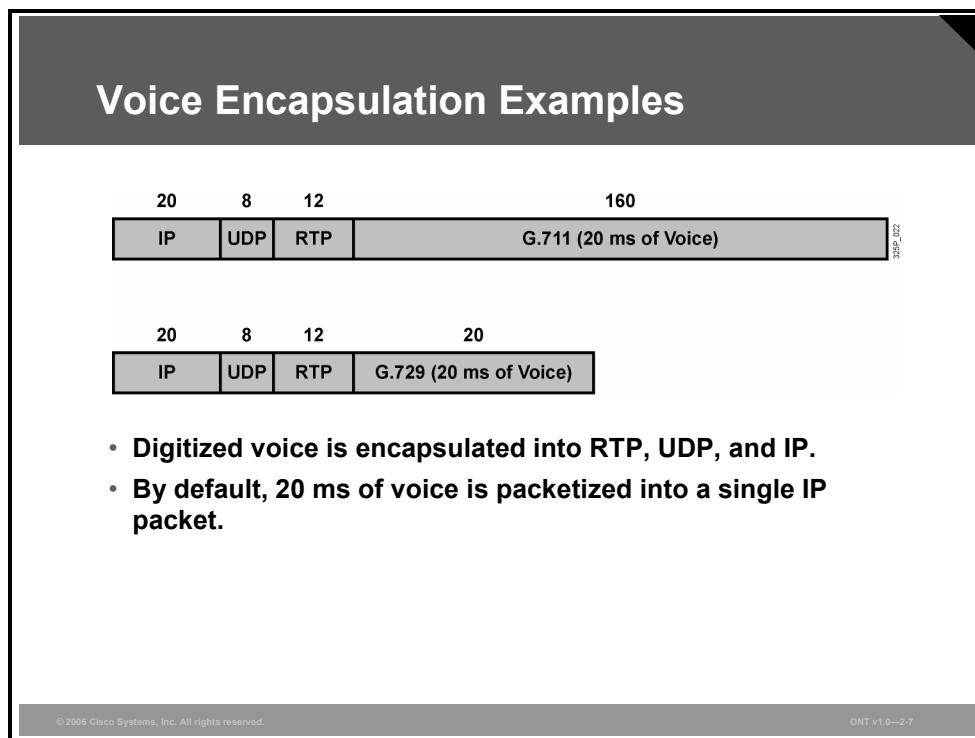
- **Reliability:** TCP offers both connection-oriented and reliable transmission. Both capabilities require some overhead. To establish a TCP connection, a TCP handshake has to occur before the actual transmission can start. Although this process might be useful for other protocols, it is not needed for voice transmission because the two parties agree on the exchange of media packets during the call setup (signaling) phase. From this perspective, both devices are aware that RTP streams will be exchanged between them, and there is no need for a session handshake. Neither RTP nor UDP is a connection-oriented protocol. Reliable transmission in TCP is an important feature for applications such as file transfers so that lost or damaged packets can be automatically re-sent. For real-time applications, such as telephony, it is totally useless. If a voice packet is lost, a TCP retransmission (which is triggered by the expiration of a retransmission timer) would arrive far too late. In such a situation it is better to lose a few packets (which will degrade quality for a short moment) rather than to re-send the packet seconds later. The TCP overhead needed to provide reliable transport is considerable (received packets are acknowledged, sent packets are kept in a retransmission buffer until they are acknowledged, and so on), but the functionality is not needed for voice transmission. Therefore, TCP is not used for the actual

transmission of voice but is used in the call setup phase. UDP and RTP do not provide reliable transport and, hence, do not introduce such unnecessary overhead.

- **Reordering:** As mentioned, in an IP network, packets can arrive in a different order than they were transmitted. Before the packet payload is passed on to the application, the correct order of packets must be ensured. TCP provides this functionality but is not an option because of its high overhead. UDP does not provide reordering, but RTP does. Using RTP, the voice packets will be delivered to the application in the correct order.
- **Time-stamping:** Real-time applications must know the relative time that a packet was transmitted. RTP time-stamps packets, which provides these benefits:
  - The packets can be correctly reordered.
  - The packets can have appropriate delays inserted between packets.
- **Multiplexing:** A VoIP device can have multiple calls active, and it must track which packets belong to each call. RTP does not provide identification of the call to which the packet belongs and therefore cannot be used to track packets to specific calls. Instead, UDP port numbers are used to identify the call to which the packet belongs. During call setup, UDP port numbers are negotiated for each call, and the VoIP device ensures that the port numbers are unique for all currently active calls. The UDP port numbers used for RTP are in the range from 16,384 to 32,767.
- **Others:** TCP has a substantially larger header (20 bytes) than UDP (8 bytes). A smaller header means that less bandwidth is needed because of lower overhead.

## Voice Encapsulation Examples

As just explained, voice is encapsulated into RTP and UDP before the IP header is added. The size of the whole VoIP packet depends on the codec and the amount of voice that is packetized. These examples assume a default of 20 ms of voice per packet.



### **Example: Encapsulation of G.711-Coded Voice**

When analog signals are digitized using the G.711 codec, 20 ms of voice consists of 160 samples, 8 bits each. The result is 160 bytes of voice information. These G.711 samples (160 bytes) are encapsulated into an RTP header (12 bytes), a UDP header (8 bytes), and an IP header (20 bytes). Therefore, the whole IP packet carrying UDP, RTP, and the voice payload has a size of 200 bytes.

### **Example: Encapsulation of G.729-Coded Voice**

In contrast, 20 ms of voice encoded with the G.729 codec consists of 160 samples, where groups of 10 samples are represented by a 10-bit codeword. The result is 160 bits (20 bytes) of voice information. These G.729 codewords (20 bytes) are encapsulated into an RTP header (12 bytes), a UDP header (8 bytes), and an IP header (20 bytes). Therefore, the whole IP packet carrying UDP, RTP, and the voice payload has a size of 60 bytes.



# Reducing Header Overhead

This topic describes how IP voice headers are compressed using compressed RTP (cRTP) and when to use it in a network.

## Voice Encapsulation Overhead

20	8	12	160
IP	UDP	RTP	G.711

20	8	12	20
IP	UDP	RTP	G.729

- Voice is sent in small packets at high packet rates.
- IP, UDP, and RTP header overheads are enormous:
  - For G.729, the headers are twice the size of the payload.
  - For G.711, the headers are one-fourth the size of the payload.
- Bandwidth is 24 kbps for G.729 and 80 kbps for G.711, ignoring Layer 2 overhead.

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—2-9

## Voice Encapsulation Overhead

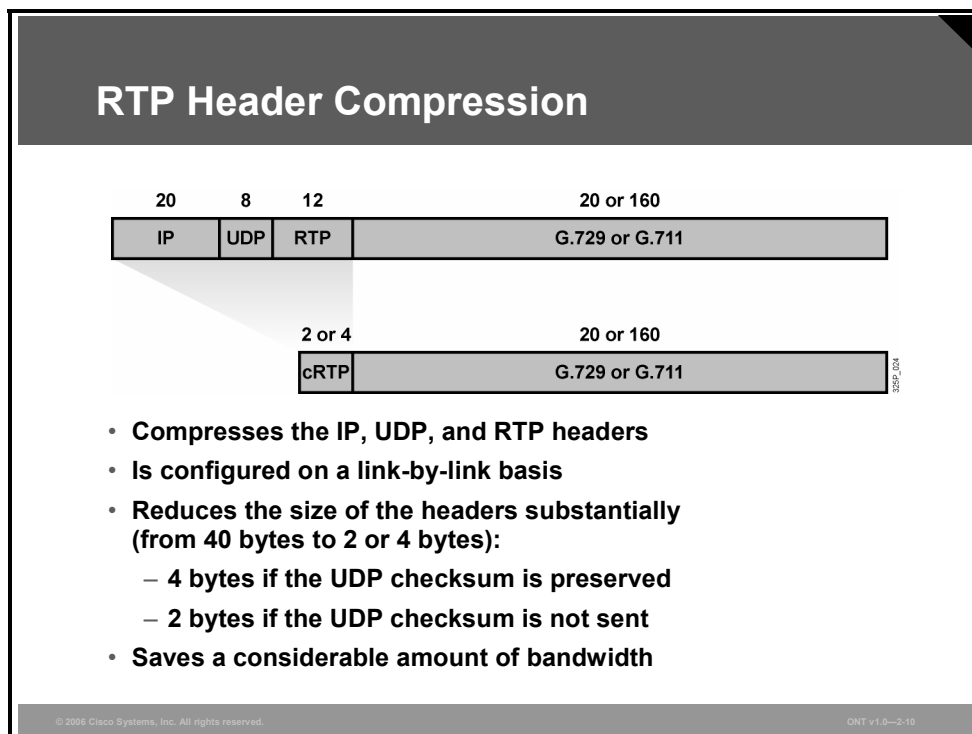
The combined overhead of IP, UDP, and RTP headers is enormously high, especially because voice is sent in relatively small packets and at high packet rates.

When G.729 is used, the headers are twice the size of the voice payload. The pure voice bandwidth of the G.729 codec (8 kbps) has to be tripled for the whole IP packet. This total, however, is still not the final bandwidth requirement, because Layer 2 overhead must also be included. Without the Layer 2 overhead, a G.729 call requires 24 kbps.

When G.711 is being used, the ratio of header to payload is smaller because of the larger voice payload. Forty bytes of headers are added to 160 bytes of payload, so one-fourth of the G.711 codec bandwidth (64 kbps) has to be added. Without Layer 2 overhead, a G.711 call requires 80 kbps.

# RTP Header Compression

To reduce the huge bandwidth overhead caused by the IP, UDP, and RTP headers, RTP header compression (cRTP) can be used. The name is a bit misleading because cRTP not only compresses the RTP header, but it also compresses the IP and UDP headers.



cRTP is configured on a link-by-link basis. There is no problem in using cRTP on just some links within your IP network. In any case—even if cRTP is configured on all links in the path—a router that receives cRTP packets on one interface and routes them out another interface that is also configured for cRTP has to decompress the packet at the first interface and then compress it again at the second interface.

cRTP compresses the IP, UDP, and RTP headers from 40 bytes to 2 bytes if the UDP checksum is not conserved (which is the default on Cisco devices) and to 4 bytes if the UDP checksum is also transmitted. cRTP is especially beneficial when the RTP payload size is small; for example, with compressed audio payloads between 20 and 50 bytes.

## cRTP Operation

cRTP works on the premise that most of the fields in the IP, UDP, and RTP headers do not change or that the change is predictable. Static fields include source and destination IP address, source and destination UDP port numbers, and many other fields in all three headers. For the fields where the change is predictable, the cRTP process is illustrated in the table.

## RTP Header Compression Process

Condition	Action
The change is predictable.	The sending side tracks the predicted change.
The predicted change is tracked.	The sending side sends a hash of the header.
The receiving side predicts what the constant change is.	The receiving side substitutes the original stored header and calculates the changed fields.
There is an unexpected change.	The sending side sends the entire header without compression.

The impact of header compression under various conditions is illustrated in these examples.

### Example: cRTP with G.729, without UDP Checksum

When RTP is used for G.729 voice streams without preserving the UDP checksum, there will be 20 bytes of voice, encapsulated into 2 bytes of cRTP. The overhead in this case is 10 percent; uncompressed encapsulation would add 200 percent of overhead.

### Example: cRTP with G.711, with UDP Checksum

When cRTP is used for G.711 voice streams, preserving the UDP checksum, there will be 160 bytes of voice, encapsulated into 4 bytes of cRTP. The overhead in this case is 2.5 percent; uncompressed encapsulation would add 25 percent of overhead.

# When to Use RTP Header Compression

Despite its advantages, there are some factors that you should consider before enabling cRTP.

## When to Use RTP Header Compression

- **Use cRTP:**
  - Only on slow links (less than 2 Mbps)
  - If bandwidth needs to be conserved
- **Consider the disadvantages of cRTP:**
  - Adds to processing overhead
  - Introduces additional delays
- **Tune cRTP—set the number of sessions to be compressed (default is 16)**

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—2-11

cRTP reduces overhead for multimedia RTP traffic. The reduction in overhead for multimedia RTP traffic results in a corresponding reduction in delay; cRTP is especially beneficial when the RTP payload size is small, such as audio payloads. Use RTP header compression on any WAN interface where you are concerned about bandwidth and where there is a high proportion of RTP traffic.

However, consider the following factors before enabling cRTP:

- Use cRTP when you need to conserve bandwidth on your WAN links, but enable cRTP only on slow links (less than 2 Mbps).
- Consider the disadvantages of cRTP:
  - cRTP adds to processing overhead, so make sure to check the available resources on your routers before turning on cRTP.
  - cRTP introduces additional delays because of the time it takes to perform compression and decompression.
- Tune cRTP by limiting the number of sessions to be compressed on the interface. The default is 16 sessions. If you see that the router CPU cannot manage that many sessions, lower the number of cRTP sessions. If the router has enough CPU power and you want to compress more than 16 sessions on a link, set the parameter to a higher value.

# Summary

This topic summarizes the key points that were discussed in this lesson.

## Summary

- In packet telephony, digitized voice is carried in IP packets, which are routed one by one across the IP network.
- Voice is encapsulated using RTP, UDP, and IP protocol headers.
- IP, UDP, and RTP headers can be compressed using cRTP to substantially reduce the encapsulation overhead.



# Calculating Bandwidth Requirements

---

## Overview

Knowing the exact bandwidth that is used per call is important for network capacity planning. This lesson describes sources of overhead and how to calculate the total bandwidth, taking overhead and voice packetization parameters into consideration.

## Objectives

Upon completing this lesson, you will be able to list the bandwidth requirements for various codecs and data links, and, given the formula to calculate total bandwidth for a VoIP call, list the methods to reduce bandwidth consumption. This ability includes being able to meet these objectives:

- Describe how the number of voice samples that are encapsulated affects bandwidth requirements
- List the overhead for various Layer 2 protocols
- Describe how IPsec and GRE or LT2P tunneling affect bandwidth overhead
- Use a formula to calculate the total bandwidth that is required for a VoIP call
- Describe the operation of VAD and bandwidth savings associated with the use of VAD

# Impact of Voice Samples and Packet Size on Bandwidth

This topic illustrates the effect of voice sample size on bandwidth.

Factors Influencing Bandwidth	
Factor	Description
Packet rate	<ul style="list-style-type: none"><li>Derived from packetization period (the period over which encoded voice bits are collected for encapsulation)</li></ul>
Packetization size (payload size)	<ul style="list-style-type: none"><li>Depends on packetization period</li><li>Depends on codec bandwidth (bits per sample)</li></ul>
IP overhead (including UDP and RTP)	<ul style="list-style-type: none"><li>Depends on the use of cRTP</li></ul>
Data-link overhead	<ul style="list-style-type: none"><li>Depends on protocol (different per link)</li></ul>
Tunneling overhead (if used)	<ul style="list-style-type: none"><li>Depends on protocol (IPsec, GRE, or MPLS)</li></ul>

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0-2.3

When voice is sent over packet networks, digitized voice information is encapsulated into IP packets. This encapsulation overhead causes extra bandwidth needs. The total bandwidth that will be required is determined by these elements:

- **Packet rate:** Packet rate specifies the number of packets sent in a certain time interval. The packet rate is usually specified in packets per second (pps). Packet rate is the multiplicative inverse of the packetization period. The packetization period is the amount of voice (time) that will be encapsulated per packet, and is usually specified in milliseconds.
- **Packetization size:** Packetization size specifies the number of bytes that are needed to represent the voice information that will be encapsulated per packet. Packetization size depends on the packetization period and the bandwidth of the codec used.
- **IP overhead:** IP overhead specifies the number of bytes added to the voice information during IP encapsulation. When voice is encapsulated into Real-Time Transport Protocol (RTP), User Datagram Protocol (UDP), and IP, the IP overhead is the sum of all these headers.
- **Data link overhead:** Data-link overhead specifies the number of bytes added during data-link encapsulation. The data-link overhead depends on the used data-link protocol, which can be different per link.
- **Tunneling overhead:** Tunneling overhead specifies the number of bytes added by any security or tunneling protocol, such as 802.1Q tunneling, IPsec, Generic Route Encapsulation (GRE), or Multiprotocol Label Switching (MPLS). This overhead must be considered on all links between the tunnel source and the tunnel destination.



Considering that some of these elements stem from the same sources, the information units that are needed for bandwidth calculation are packetization period or packetization size, codec bandwidth, IP overhead, data-link overhead, and tunneling or security overhead.

## Bandwidth Implications of Codecs

A codec transforms analog signals into digital format.

Bandwidth Implications of Codecs															
<ul style="list-style-type: none"><li>• <b>Codec bandwidth is for voice information only</b></li><li>• <b>No packetization overhead included</b></li></ul>	<table><tr><th>Codec</th><th>Bandwidth</th></tr><tr><td><b>G.711</b></td><td><b>64 kbps</b></td></tr><tr><td><b>G.726 r32</b></td><td><b>32 kbps</b></td></tr><tr><td><b>G.726 r24</b></td><td><b>24 kbps</b></td></tr><tr><td><b>G.726 r16</b></td><td><b>16 kbps</b></td></tr><tr><td><b>G.728</b></td><td><b>16 kbps</b></td></tr><tr><td><b>G.729</b></td><td><b>8 kbps</b></td></tr></table>	Codec	Bandwidth	<b>G.711</b>	<b>64 kbps</b>	<b>G.726 r32</b>	<b>32 kbps</b>	<b>G.726 r24</b>	<b>24 kbps</b>	<b>G.726 r16</b>	<b>16 kbps</b>	<b>G.728</b>	<b>16 kbps</b>	<b>G.729</b>	<b>8 kbps</b>
Codec	Bandwidth														
<b>G.711</b>	<b>64 kbps</b>														
<b>G.726 r32</b>	<b>32 kbps</b>														
<b>G.726 r24</b>	<b>24 kbps</b>														
<b>G.726 r16</b>	<b>16 kbps</b>														
<b>G.728</b>	<b>16 kbps</b>														
<b>G.729</b>	<b>8 kbps</b>														
<small>© 2006 Cisco Systems, Inc. All rights reserved. DNT v1.0—2-4</small>															

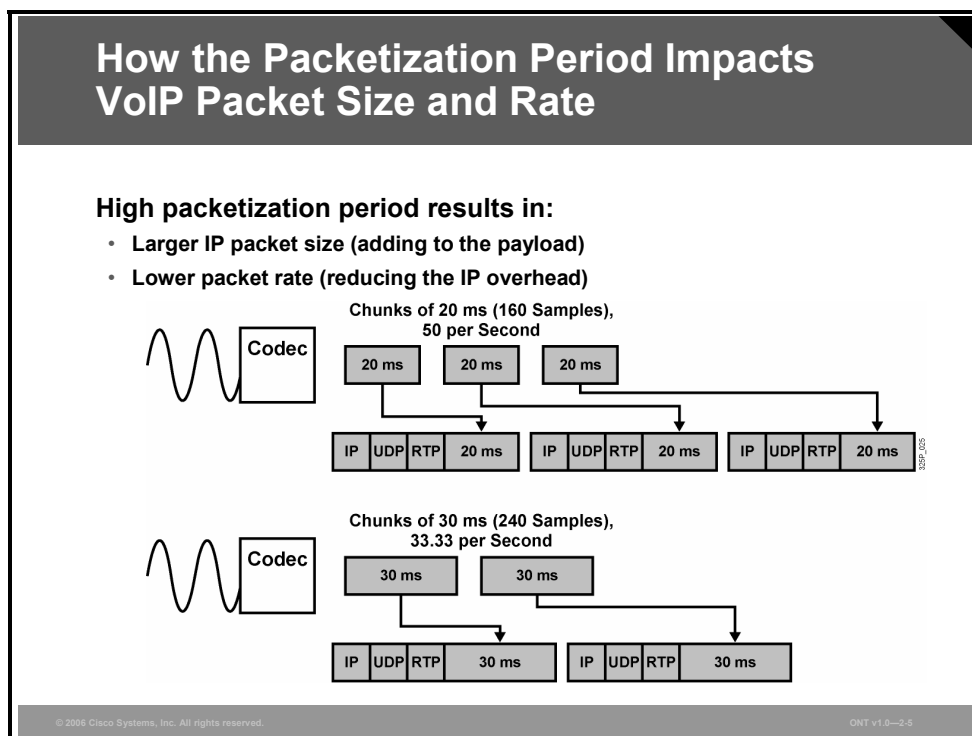
Different codecs have different bandwidth requirements:

- **G.711:** The G.711 codec uses the most bandwidth. It encodes each of the 8000 samples taken each second into 8 bits, resulting in a 64-kbps codec bandwidth.
- **G.722:** The G.722 wideband codec splits the input signal into two sub-bands and uses a modified version of adaptive differential pulse code modulation (ADPCM) (including adaptive prediction) for each band. The bandwidth of G.722 is 64, 56, or 48 kbps.
- **G.726:** The G.726 ADPCM coding schemes use somewhat less bandwidth. They encode each of the 8000 samples taken each second using 4, 3, or 2 bits, resulting in bandwidths of 32, 24, or 16 kbps.
- **G.728:** The G.728 low-delay code excited linear prediction (LDCELP) coding scheme compresses pulse code modulation (PCM) samples using a codebook. Waveshapes of five samples are represented by a 10-bit codeword, which identifies the best matching pattern of the codebook. Because of the compression of five samples (worth 40 bits in PCM) to 10 bits, the bandwidth of LDCELP is 16 kbps.
- **G.729:** The G.729 conjugate structure algebraic code excited linear prediction (CS-ACELP) coding scheme also offers codebook-based compression. Waveshapes of 10 bits are represented by a 10-bit codeword, reducing the bandwidth to 8 kbps.

The bandwidth codec, however, indicates only the bandwidth required for the digitized voice itself. It does not include any packetization overhead.

# How the Packetization Period Affects VoIP Packet Size and Rate

The packetization overhead that must be added to the codec bandwidth depends on the size of the added headers and the packet rate. The more packets sent, the more often IP, UDP, and RTP headers have to be added to the voice payload. The overall VoIP bandwidth is composed of the size of the whole VoIP packet (headers and voice payload) and the rate at which the VoIP packets are sent.



On VoIP devices, in addition to the codec, you can specify the amount of voice that is encapsulated per packet. Usually, this value is configured by the packetization period (in milliseconds). A higher packetization period results in a larger IP packet size, because of the larger payload (the digitized voice samples). However, a higher packetization period results in a lower packet rate, reducing the IP overhead because of the smaller number of packets that must be generated.

The figure contrasts two scenarios with different packetization periods. In the first scenario, chunks of 20 ms of voice (160 PCM samples) are packetized. The packet rate is the reciprocal of the packetization period. If packetization is done every 20 ms, 50 packets are generated per second. For a total of 60 ms of voice (as shown in the figure), three packets, each carrying 20 ms of voice, will be needed. Therefore, packetization of this 60 ms of voice introduces an overhead of three IP, UDP, and RTP headers.

In the second scenario, chunks of 30 ms of voice (240 PCM samples) are packetized. This results in a lower packet rate of 33.3 pps. For the 60 ms of voice shown in the figure, only two packets (carrying 30 ms of voice each) are generated, so the IP overhead is reduced by one third, compared to the first scenario.

The default value for the packetization period on most Cisco VoIP devices is 20 ms. This default is the optimal value for most scenarios. When you consider increasing this value to benefit from lower IP encapsulation overhead, you also have to consider that a higher packetization period causes a higher delay. The extra delay is introduced during packetization because more voice information has to be collected before a packet can be generated and sent off.

---

<b>Caution</b>	You should only increase the default packetization period if you are sure that the additional delay can be accepted (for instance, if other sources of delay, such as buffering, are rather small) and if you cannot solve bandwidth issues by any other means (for example, using compressed RTP [cRTP] or adding bandwidth). On the other hand, because of the additional overhead, you should avoid trying to reduce delay by decreasing the default packetization period but use other methods (such as quality of service [QoS] or adding bandwidth).
----------------	--

---

## VoIP Packet Size and Packet Rate Examples

The table shows examples of VoIP encapsulation with varying codecs and packetization periods.

VoIP Packet Size and Packet Rate Examples				
Codec and Packetization Period	G.711 20 ms	G.711 30 ms	G.729 20 ms	G.729 40 ms
Codec bandwidth (kbps)	64	64	8	8
Packetization size (bytes)	160	240	20	40
IP overhead (bytes)	40	40	40	40
VoIP packet size (bytes)	200	280	60	80
Packet rate (pps)	50	33.33	50	25

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0-3.6

You can see that an increased packetization period increases the size of the IP packet while it reduces the packet rate.

In the table, the IP overhead was assumed to be 40 bytes. This is the normal value for VoIP packets composed of 20 bytes of IP header, 8 bytes of UDP header, and 12 bytes of RTP header. If RTP header compression (cRTP) is used, the IP overhead would be smaller.

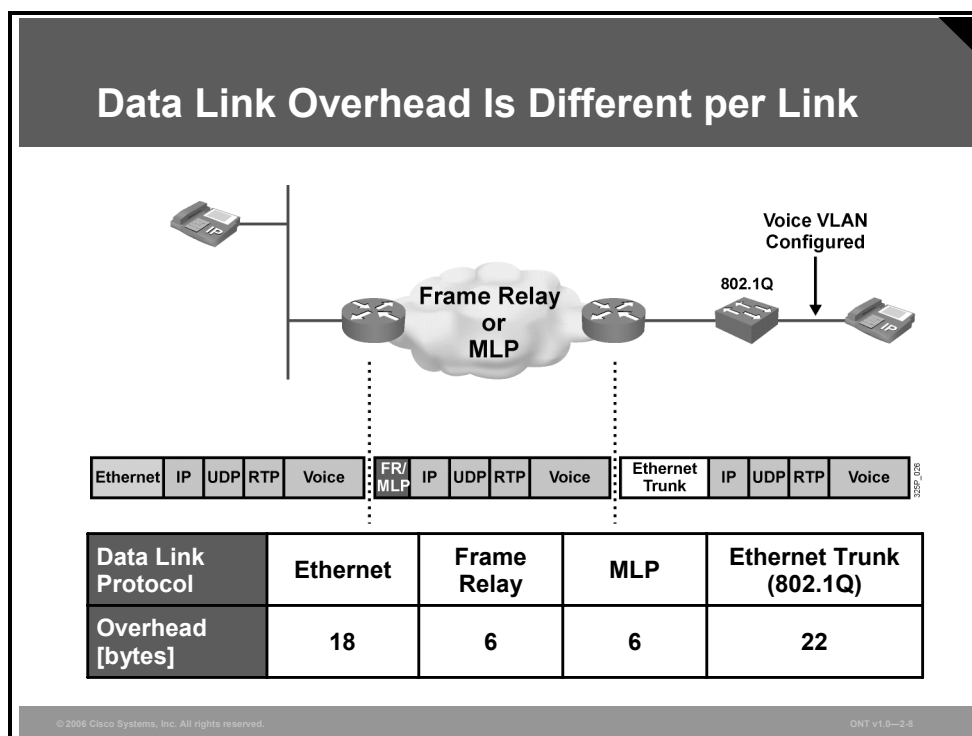
---

**Note** This table shows the IP packet size only and does not consider the overhead caused by data-link encapsulation.

---

# Data-Link Overhead

This topic lists overhead sizes for various Layer 2 protocols.



When IP packets are sent over a link within an IP network, they have to be encapsulated using the data-link protocol for that link. Each link can use a different data-link protocol.

The figure illustrates an IP packet being sent from one IP phone to another. The two IP phones are located in different LANs, separated by a Frame Relay network. Before the sending phone transmits the VoIP packet onto the LAN, it has to encapsulate it into an Ethernet frame. The router that receives the frame removes the Ethernet header and encapsulates the VoIP packet into Frame Relay before sending it out to the WAN. The router receiving the VoIP packet from the Frame Relay network removes the Frame Relay header and encapsulates the VoIP packet into an Ethernet frame again before passing it on to the receiving IP phone. As illustrated in the figure, the Ethernet header and the Frame Relay header differ in size.

The overhead of data-link protocols commonly used for VoIP is 18 bytes for Ethernet, 22 bytes for 802.1Q-tagged Ethernet frames, and 6 bytes for Frame Relay or multilink PPP (MLP).

When you are calculating the bandwidth of a VoIP call for a certain link, the appropriate overhead of the data-link protocol has to be considered.

# Security and Tunneling Overhead

This topic describes overhead associated with various security and tunneling protocols.

## Security and Tunneling Overhead

- **IP packets can be secured by IPsec.**
- **Additionally, IP packets or data-link frames can be tunneled over a variety of protocols.**
- **Characteristics of IPsec and tunneling protocols are:**
  - **The original frame or packet is encapsulated into another protocol.**
  - **The added headers result in larger packets and higher bandwidth requirements.**
  - **The extra bandwidth can be extremely critical for voice packets because of the transmission of small packets at a high rate.**

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0–2-10

IP packets, consequently also VoIP packets, can be secured using IPsec. There are two IPsec modes: transport mode and tunnel mode. In either mode, the packets can be protected by the Authentication Header (AH) or the Encapsulating Security Payload (ESP) header or both. In tunnel mode, an additional IP header is generated, allowing the use of virtual private networks (VPNs).

Additionally, IP packets or data-link frames can be tunneled over a variety of protocols. Examples for such tunnel protocols include these:

- GRE, which can transport network layer packets or data-link frames over IP packets
- Layer 2 Forwarding (L2F) and Layer 2 Tunneling Protocol (L2TP), which tunnel PPP frames over IP networks
- PPP over Ethernet (PPPoE), which allows PPP to be used over Ethernet
- 802.1Q tunneling, which transports 802.1Q frames inside another VLAN

Tunneling protocols and IPsec have some common characteristics. They all encapsulate the original packet or frame into another protocol. Adding the tunneling protocol header increases the size of the original packet, resulting in higher bandwidth needs. The extra bandwidth can be critical, especially for voice packets, because of their high transmission rate and small packet size. The larger the size of the additional headers, the greater the need for extra bandwidth for VoIP packets.

# Extra Headers in Security and Tunneling Protocols

IPsec and tunneling protocols add headers of different sizes.

Extra Headers in Security and Tunneling Protocols	
Protocol	Header Size (bytes)
IPsec transport mode	30–53
IPsec tunnel mode	50–73
L2TP/GRE	24
MPLS	4
PPPoE	8

© 2006 Cisco Systems, Inc. All rights reserved. ORT v1.0—2.11

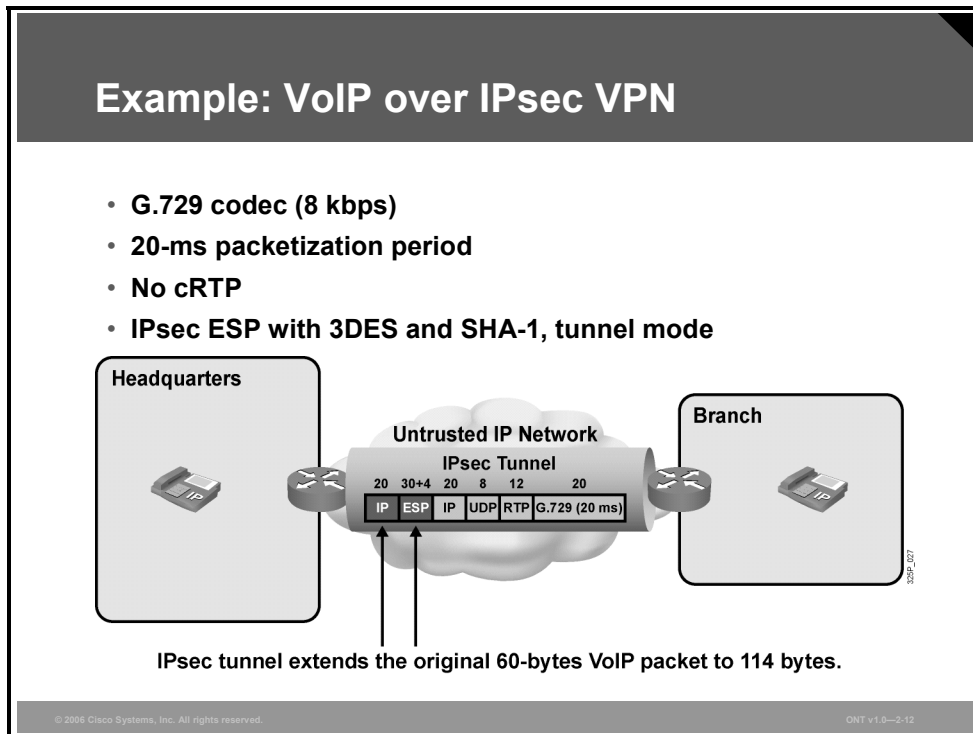
IPsec overhead depends on the use of the available headers (AH and ESP), the encryption or authentication algorithms used in these headers, and the mode (transport or tunnel mode). Because AH supports only authentication while ESP supports authentication and encryption, ESP is used more often. With Data Encryption Standard (DES) or Triple DES (3DES) used for encryption and Message Digest 5 (MD5) or Secure Hash Algorithm 1 (SHA-1) used for authentication, the ESP header adds 30 to 37 bytes in transport mode. When Advanced Encryption Standard (AES) is used as the encryption algorithm and AES-extended cipher block chaining (AES-XCBC) for authentication, 38 to 53 bytes are added in transport mode. ESP DES and 3DES require the payload to be rounded up to multiples of 8 bytes (resulting in 0–7 bytes of padding), while the ESP AES payload is rounded up to multiples of 16 bytes (resulting in 0–15 bytes of padding).

In tunnel mode, an extra 20 bytes must be added for the additional IP header.

L2TP or GRE adds 24 bytes to the original PPP frame, MPLS adds 4 bytes to the original IP packet, and PPPoE puts an extra 8-byte PPPoE header between the Ethernet frame and the IP packet.

## Example: VoIP over IPsec VPN

The example shows a company with two sites. The headquarters site is separated from the branch site by an untrusted network. IPsec is used by the routers connecting the sites over the untrusted network. IPsec ESP tunnel mode is used with 3DES encryption and SHA-1 authentication. IP phones located at each site use the G.729 codec, with a default packetization period of 20 ms. RTP header compression is not enabled.

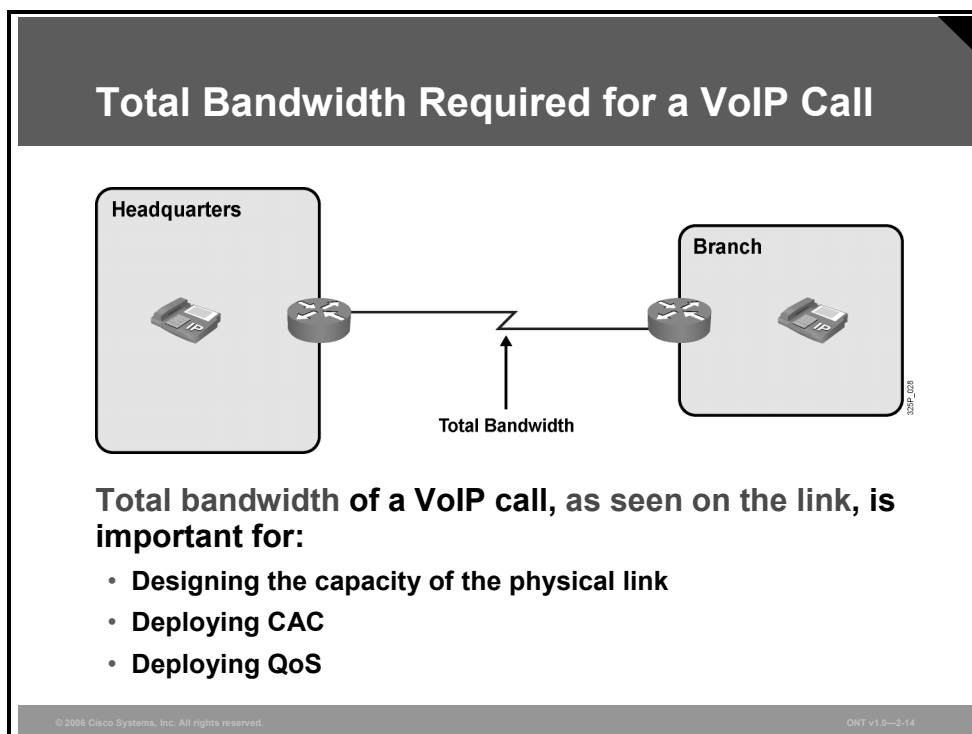


During a voice call between the headquarters and the branch site, every 20 ms, the IP phones encapsulate 20 bytes of digitized voice into RTP, UDP, and IP, resulting in IP packets of 60 bytes. When these VoIP packets are sent out to the untrusted network by the routers, the routers encapsulate each packet into another IP header and protect it by an ESP header. This process adds an additional 54 bytes (20 for the extra IP header, 4 bytes of padding to get to a payload size of 64 bytes, and 30 bytes for the ESP header) to the original VoIP packets. The IPsec packet, which is now transporting the VoIP packet, has a size of 114 bytes, which is almost twice the size of the original VoIP packet.



# Calculating the Total Bandwidth for a VoIP Call

This topic calculates the total bandwidth required for a VoIP call using codec, data link, and sample size.



When you are designing networks for VoIP, it is crucial to know the total bandwidth of a VoIP call as it is seen on the link. This information is needed for designing the capacity of physical links, as well as for deploying Call Admission Control (CAC) and QoS. CAC limits the number of concurrent voice calls, avoiding oversubscription of the link, which would cause quality degradation. QoS gives priority to voice packets, avoiding too-high delays caused by queuing, which again would affect voice quality.

# Total Bandwidth Calculation Procedure

This section illustrates the procedure of calculating the total bandwidth.

## Total Bandwidth Calculation Procedure

1. **Gather required packetization information:**
  - Packetization period (default is 20 ms) or size
  - Codec bandwidth
2. **Gather required information about the link:**
  - cRTP enabled
  - Type of data-link protocol
  - IPsec or any tunneling protocols used
3. **Calculate the packetization size or period.**
4. **Sum up packetization size and all headers and trailers.**
5. **Calculate the packet rate.**
6. **Calculate the total bandwidth.**

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0–2-15

To calculate the total bandwidth of a VoIP call, perform these steps:

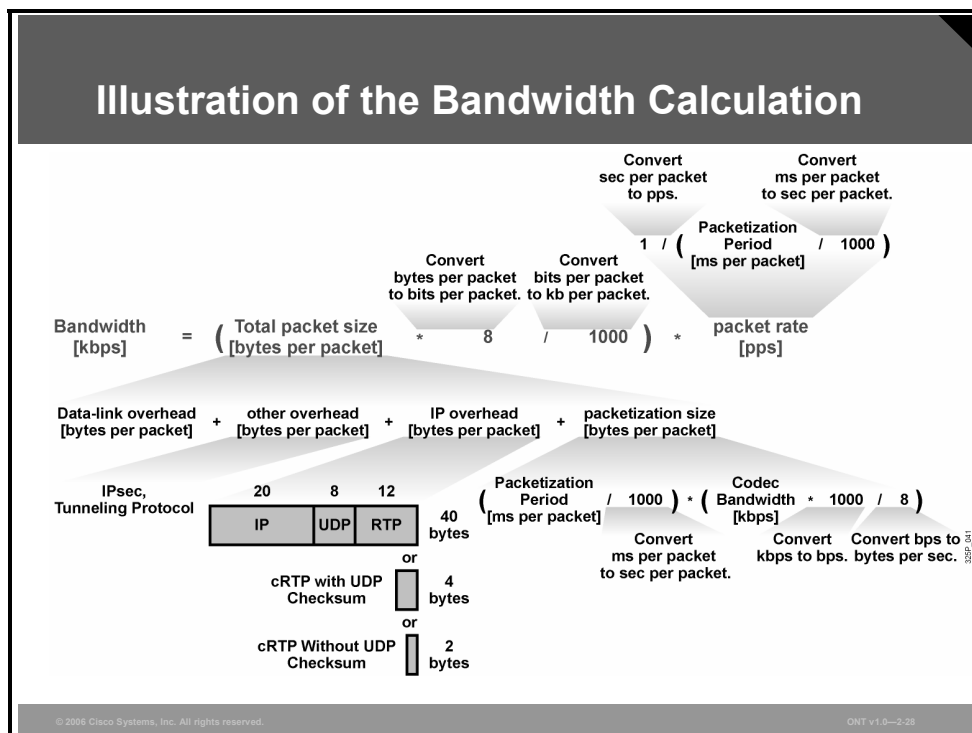
- Step 1 Gather required packetization information:** First, you must determine the bandwidth of the codec that is used to digitize the analog signals. The codec bandwidth is specified in kilobits per second and is usually in the range of approximately 8–64 kbps. In addition, the packetization period (specified in milliseconds) or the packetization size (specified in bytes) is required. Given the codec bandwidth and one of these two values, the other value can be calculated.
- Step 2 Gather required information about the link:** The amount of overhead that will be added per packet on each link is the next piece of information needed. Will cRTP be used? Which data-link protocol is in use, and what is its overhead per packet? IP, UDP, and RTP overhead is 40 bytes unless cRTP is used, then it is 2 (the default) or 4 bytes. Make sure to include the overhead (in bytes) of the data-link protocol that is used. Finally, you must observe whether any other features that cause additional overhead are being used. These could be security features, such as VLANs, IPsec, or any special tunneling applications.
- Step 3 Calculate the packetization size or period:** Depending on the voice device, you might know either the packetization period or the packetization size (determined in Step 1). You have to calculate the missing information based on the known value plus the codec bandwidth, also noted in Step 1. The packetization size is expressed in bytes, the packetization period in milliseconds.
- Step 4 Sum up the packetization size and all headers and trailers:** Add the overhead of IP, UDP, and RTP (or cRTP), the data-link protocol, and any other protocols that you noted in Step 2 to the voice payload (packetization size), which you determined either in Step 1 or Step 3. All values must be in bytes.

- Step 5     Calculate the packet rate:** Calculate how many packets will be sent per second by using the multiplicative inverse of the packetization period. Because the packet rate is specified in packets per second, make sure to convert the milliseconds value of the packetization period to seconds.
- Step 6     Calculate the total bandwidth:** Multiply the total size of the packet or frame by the packet rate to get the total bandwidth. Because the packet size is specified in bytes and the bandwidth is specified in kilobits per second, you need to convert bytes to kilobits.

Based on this procedure, you can calculate the bandwidth used by a VoIP call on a specific link. For planning the capacity of physical links, you will have to consider the maximum number of calls that should be supported and the bandwidth that is needed for applications other than VoIP. In addition, you have to ensure that enough bandwidth is available for call setup and call teardown signaling. Although signaling messages need relatively little bandwidth, you should not forget to provision the bandwidth for signaling protocols (especially in your QoS configuration).

# Illustration of the Bandwidth Calculation

This section represents the bandwidth calculation in a form of a formula.



Following the steps of the bandwidth calculation procedure, the following formula can be generated:

$$\text{Bandwidth [kbps]} = (\text{Total packet size [bytes per packet]} \times 8 / 1000) \times \text{packet rate [pps]}$$

Because the total packet size is specified in bytes while the bandwidth is specified in kilobits per second, the total packet size has to be multiplied by 8 and divided by 1000 to convert bytes to kilobits per second.

## Calculation of the Total Packet Size

The total packet size can be calculated by using this formula:

$$\text{Total packet size [bytes per packet]} = \text{Data-link overhead [bytes per packet]} + \text{other overhead [bytes per packet]} + \text{IP overhead [bytes per packet]} + \text{packetization size [bytes per packet]}$$

The data link overhead is the overhead caused by the data-link protocol during Layer 2 encapsulation.

The other overhead is any additional overhead, for example, the overhead caused by using IPsec or any of the tunneling protocols.

If cRTP is not used, the IP overhead is 40 bytes for the IP, UDP, and RTP headers. If cRTP is used, the IP overhead is 2 bytes, when the UDP checksum is not transmitted, or 4 bytes, when the UDP checksum is sent.

The packetization size is the size of the voice payload that is encapsulated per packet. It is either known (instead of the packetization period) or calculated using this formula:

$$\text{Packetization size [bytes per packet]} = (\text{Packetization period [ms per packet]} / 1000) \times \text{codec bandwidth [kbps]} \times 1000 / 8$$

Because the packetization size is specified in bytes and the codec bandwidth is specified in kilobits per second, the codec bandwidth has to be converted by dividing it by 8 and multiplying it by 1000. In addition, because of the mismatch of the units in the codec bandwidth (kilobits per second) and packetization period (milliseconds per packet), the packetization period has to be converted by dividing it by 1000. Balancing the conversions, the formula can be simplified to this:

$$\text{Packetization size [bytes per packet]} = \text{Packetization period [ms per packet]} * \text{codec bandwidth [kbps]} / 8$$

## Calculation of the Packet Rate

The packet rate, specified in packets per second, is the multiplicative inverse of the packetization period, which is specified in milliseconds per packet. Therefore, you have to convert the packetization period from milliseconds to seconds when building the reciprocal value:

$$\text{Packet rate [pps]} = 1 / (\text{packetization period [ms per packet]} / 1000)$$

Sometimes, you will not know the packetization period (milliseconds per packet), but you will know the packetization size (bytes per packet). This is because on some devices, the packetization size is configured instead of the packetization period. In that case, you have to calculate the packetization period first, using this formula:

$$\text{Packetization period [ms per packet]} = (\text{Packetization size [bytes per packet]} * 8 / 1000) / (\text{codec bandwidth [kbps]} / 1000)$$

Because of the different units used for the packetization size (bytes per packet) and the codec bandwidth (kilobits per second), you have to multiply the packetization size by 8 and then divide it by 1000. In addition, you must convert the codec bandwidth value (because the packetization period uses milliseconds instead of seconds) by dividing the codec bandwidth by 1000. Balancing the two conversions, the formula can be simplified to this:

$$\text{Packetization period [ms per packet]} = \text{Packetization size [bytes per packet]} * 8 / \text{codec bandwidth [kbps]}$$

## Summary

Assuming that you know the packetization period (in milliseconds per packet), the formulas can be aggregated and then simplified to this:

$$\text{Bandwidth [kbps]} = (8 * (\text{data-link overhead [bytes per packet]} + \text{other overhead [bytes per packet]} + \text{IP overhead [bytes per packet]}) + \text{packetization period [ms per packet]} * \text{codec bandwidth [kbps]}) / \text{packetization period [ms per packet]}$$

If the packetization size (in bytes per packet) is known instead of the packetization period (milliseconds per packet), the simplest way to calculate the total bandwidth is the following:

$$\text{Bandwidth [kbps]} = (\text{Codec bandwidth [kbps]} / \text{packetization size [bytes per packet]}) * (\text{packetization size [bytes per packet]} + \text{data-link overhead [bytes per packet]} + \text{other overhead [bytes per packet]} + \text{IP overhead [bytes per packet]})$$

## Quick Bandwidth Calculation

The quick way to calculate the total bandwidth when the packetization size is given is illustrated in the figure.

### Quick Bandwidth Calculation

$$\frac{\text{Total packet size}}{\text{Payload size}} = \frac{\text{Total bandwidth requirement}}{\text{Nominal bandwidth requirement}}$$

Total packet size = All headers + payload

Parameter	Value
Layer 2 header	6 to 18 bytes
IP + UDP + RTP headers	40 bytes
Payload size (20-ms sample interval)	20 bytes for G.729, 160 bytes for G.711
Nominal bandwidth	8 kbps for G.729, 64 kbps for G.711

Example: G.729 with Frame Relay:

$$\text{Total bandwidth requirement} = \frac{\text{Total packet size} * \text{nominal bandwidth requirement}}{\text{payload size}} = \frac{(6 + 40 + 20 \text{ bytes}) * 8 \text{ kbps}}{20 \text{ bytes}} = 26.4 \text{ kbps}$$

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—5.29

A quick way to calculate total bandwidth requirement for a voice packet is to remember that the ratio of the total bandwidth requirement to the nominal bandwidth for the payload is the same as the ratio of total packet size to payload size.

The payload size is dependent on the sample interval and the codec used and is usually 20 bytes for G.729 and 160 bytes for G.711, assuming a 20-ms sample interval.

The headers are always 40 bytes for the IP, UDP, and RTP headers, plus the Layer 2 header size. This size might be 6 bytes for Frame Relay or PPP or 18 bytes for Ethernet, for example.

To calculate the total bandwidth, find the total packet size, including all the headers plus payload and divide by the payload size. Multiply the result by the nominal bandwidth for the codec, and the result will be the total bandwidth requirement.

---

**Note** At <http://tools.cisco.com/Support/VBC/do/CodecCalc1.do>, you can try to calculate the bandwidth using the Voice Codec Bandwidth Calculator. Access to the URL requires a valid Cisco.com account.

---

# Effects of VAD on Bandwidth

This topic describes the effect of voice activity detection (VAD) on total bandwidth.

## VAD Characteristics

- **Detects silence (speech pauses)**
- **Suppresses transmission of “silence patterns”**
- **Depends on multiple factors:**
  - **Type of audio (for example, speech or MoH)**
  - **Level of background noise**
  - **Others (for example, language, character of speaker, or type of call)**
- **Can save up to 35 percent of bandwidth**

© 2006 Cisco Systems, Inc. All rights reserved.

QNT v1.0—2-31

## VAD Characteristics

In a circuit-switched telephony network, because of the nature of the network, the bandwidth of a call is permanently available and dedicated to that call. There is no way to take advantage of speech pauses, one-way audio transmission, or similar instances when a link is not being utilized. In a packet network, however, VAD can take advantage of the fact that one-third of the average voice call consists of silence.

VAD detects silence, for instance, caused by speech pauses or by one-way audio transmission while a caller is listening to music on hold (MoH) when being transferred. VAD suppresses the transmission of silence and, therefore, saves bandwidth.

The amount of bandwidth that can be saved by VAD depends on several factors:

- **Type of audio:** During a human conversation, the two parties do not generally talk at the same time. When MoH is played, the call usually turns into a one-way call. Because of the constantly playing music, no bandwidth can be saved in this direction of the call. However, the caller listening to the music does not send any audio and no packets have to be transmitted while the call is on hold.
- **Level of background noise:** VAD needs to detect silence to be able to perform silence suppression. If the background noise is too high, VAD cannot detect silence and continues the transmission.
- **Others:** Differences in the language and character of speakers have an impact to the amount of silence in a call. Some calls, such as conferences or broadcasts where only one or a few participants are speaking and most of the participants are listening, allow higher bandwidth savings than other calls.

On average, the use of VAD can save about 35 percent of bandwidth. Because of the factors mentioned, there is considerable deviation per individual call. Therefore, the average of 35 percent assumes a certain statistical distribution of call types, which is usually achieved only if a link carries at least 24 calls. If you are calculating bandwidth for fewer calls, you should not take VAD into account.

## VAD Bandwidth Reduction Examples

The table shows the reduced bandwidth needs of various calls, assuming that VAD can save 35 percent of bandwidth.

VAD Bandwidth-Reduction Examples				
<b>Data-Link Overhead</b>	<b>Ethernet 18 bytes</b>	<b>Frame Relay 6 bytes</b>	<b>Frame Relay 6 bytes</b>	<b>MLPP 6 bytes</b>
<b>IP overhead</b>	<b>no cRTP 40 bytes</b>	<b>cRTP 4 bytes</b>	<b>no cRTP 40 bytes</b>	<b>cRTP 2 bytes</b>
<b>Codec</b>	<b>G.711 64 kbps</b>	<b>G.711 64 kbps</b>	<b>G.729 8 kbps</b>	<b>G.729 8 kbps</b>
<b>Packetization</b>	<b>20 ms 160 bytes</b>	<b>30 ms 240 bytes</b>	<b>20 ms 20 bytes</b>	<b>40 ms 40 bytes</b>
<b>Bandwidth without VAD</b>	<b>87.2 kbps</b>	<b>66.67 kbps</b>	<b>26.4 kbps</b>	<b>9.6 kbps</b>
<b>Bandwidth with VAD (35% reduction)</b>	<b>56.68 kbps</b>	<b>43.33 kbps</b>	<b>17.16 kbps</b>	<b>6.24 kbps</b>

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—2-32



# Summary

This topic summarizes the key points that were discussed in this lesson.

## Summary

- The amount of voice that is encapsulated per packet affects the packet size and the packet rate. More packets result in higher overhead caused by added IP headers.
- Different data link protocols add different amounts of overhead during encapsulation.
- IPsec and tunneling protocols add to the packet size resulting in higher bandwidth needs.
- The total bandwidth is calculated in several steps, including the determination of packet size and packet rate and the multiplication of these two values.
- VAD can save up to 35 percent bandwidth.



# Implementing Voice Support in an Enterprise Network

---

## Overview

Implementing voice support in an enterprise network requires component and feature selection and determination of the best IP telephony deployment model. This lesson provides an overview of the available components, features, and deployment models and lists some gateway configuration features.

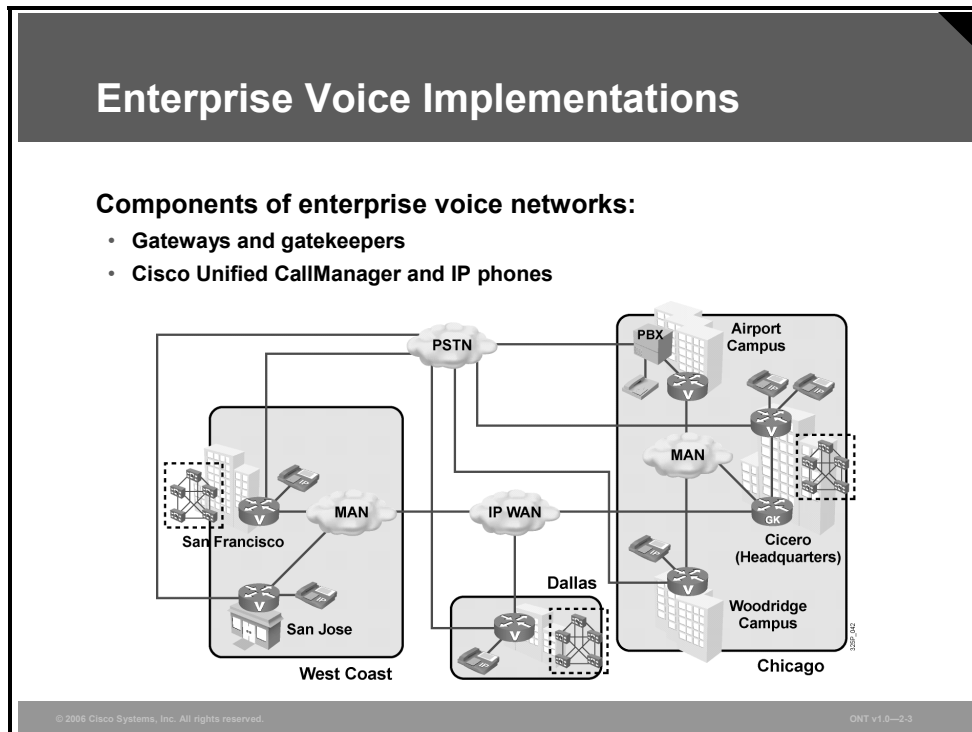
## Objectives

Upon completing this lesson, you will be able to describe various aspects of voice network implementation. This ability includes being able to meet these objectives:

- Given an enterprise network topology diagram, identify the components that are necessary for VoIP support
- Describe the voice capabilities available on Cisco ISRs
- Explain the role of a call agent, such as Cisco Unified CallManager, in a VoIP implementation
- Describe the main IP telephony deployment models that may be used in an enterprise
- Given a **show running-config** output from a Cisco router configured as a voice gateway, identify the sections of the configuration that are related to the voice implementation on the router
- Explain how CAC prevents calls from crossing overly busy links and how such calls can be rerouted by mechanisms such as AAR instead of simply being blocked

# Enterprise Voice Implementations

This topic describes a VoIP implementation in an enterprise.



Enterprise voice implementations use components such as gateways, gatekeepers, Cisco Unified CallManager, and IP phones. Cisco Unified CallManager offers PBX-like features to IP phones. Gateways interconnect traditional telephony systems, such as analog or digital phones, PBXs, or the public switched telephone network (PSTN) to the IP telephony solution. Gatekeepers can be used for scalability of dial plans and for bandwidth management when using the H.323 protocol.

## Example of an Enterprise Voice Implementation

The figure shows a company with a headquarters in Chicago, two offices on the west coast in the San Francisco area, and one smaller office in Dallas. In Chicago, three headquarters locations are connected via a metropolitan area network (MAN). The main west coast office in San Francisco is connected with the San Jose office via a MAN. Dallas has a single site. The three main locations of the company (Chicago, San Francisco bay area, and Dallas) are interconnected via an IP WAN.

The Chicago, San Francisco, and Dallas locations each have a Cisco Unified CallManager cluster serving local IP phones and IP phones located in the MAN-connected sites. At the airport campus in Chicago, IP phones are not used because the company is using a managed telephony service offered by the owner of the building. However, a voice gateway connects to the managed PBX, allowing VoIP calls to and from the airport office phones through the gateway.

The Chicago headquarters sites use the IP WAN router as a gatekeeper that provides Call Admission Control (CAC) and bandwidth management for H.323 calls. In addition, each main site has a voice gateway that connects to the PSTN, allowing off-net calls. These gateway routers are equipped with digital signal processors (DSPs) that provide conferencing and transcoding resources. Within each area, the G.711 codec is used, while calls between the three areas use the G.729 codec. All calls within the enterprise should use the IP WAN. Should the IP WAN fail, or when calls are denied by CAC, the calls are rerouted through the PSTN.

# Voice Gateway Functions on a Cisco Router

This topic describes the voice capabilities available on Cisco integrated services routers (ISRs).

## Voice Gateway Functions on a Cisco Router

- **Connect traditional telephony devices to VoIP**
- **Convert analog signals to digital format**
- **Encapsulate voice into IP packets**
- **Perform voice compression**
- **Provide DSP resources for conferencing and transcoding**
- **Support fallback scenarios for IP phones (Cisco SRST)**
- **Act as a call agent for IP phones (Cisco Unified CallManager Express)**
- **Provide DTMF relay and fax and modem support**



© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—2.5

Cisco routers, especially the Cisco ISR, such as the Cisco 3800 Series Integrated Services Router, are voice capable. These routers can be equipped with traditional telephony interfaces to act as gateways for analog and digital devices, such as phones, faxes, PBXs, and the PSTN, allowing those devices to interact with the VoIP world. Numerous analog interfaces, digital interfaces, and signaling protocols are supported, including these:

- Foreign Exchange Station (FXS)
- Foreign Exchange Office (FXO)
- Ear and mouth (E&M, also called earth and magneto)
- T1 or E1 channel associated signaling (CAS) and T1 or E1 common channel signaling (CCS) using ISDN
- Q Signaling (Q.SIG) protocols

Gateways with analog interfaces must convert analog signals into digital format before voice is encapsulated into IP packets. They can compress digitized voice before the encapsulation to reduce the bandwidth needed per call.

Cisco IOS routers support H.323, session initiation protocol (SIP), and Media Gateway Control Protocol (MGCP) for VoIP signaling. In addition, gateways can be equipped with DSPs, which provide conferencing and transcoding resources.

In IP telephony environments, gateways support fallback scenarios for IP phones that have lost IP connectivity to their call agent (that is, Cisco Unified CallManager). This feature, called Cisco Survivable Remote Site Telephony (SRST), enables the gateway to take the role of the call agent during WAN failure. Local calls can then proceed even if IP connectivity to Cisco

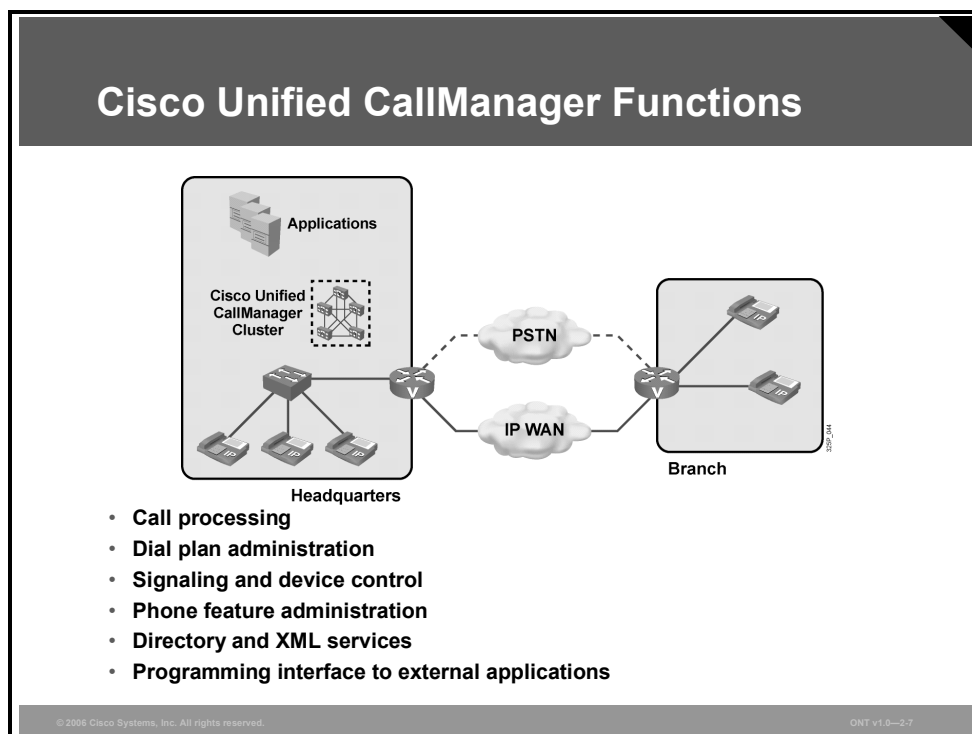
Unified CallManager is broken. In addition, Cisco SRST can route calls out to the PSTN and, thus, use the PSTN as the backup route for calls toward any site that is not reachable via IP.

Further, Cisco IOS routers can permanently act as a call agent for IP phones. The feature that provides this functionality is Cisco Unified CallManager Express. With Cisco Unified CallManager Express, Cisco Unified CallManager functionality is provided by the router. If the router is also a voice gateway, it combines IP telephony and VoIP gateway functionality in a single box.

Cisco IOS gateways also support other features, such as call preservation (Real-Time Transport Protocol [RTP] stream) in case of a lost signaling channel, dual tone multifrequency (DTMF) relay capabilities, supplementary services support (for user functions, such as hold, transfer, and conferencing), and fax and modem support.

# Cisco Unified CallManager Functions

This topic explains the role of a call agent, such as Cisco Unified CallManager, in a VoIP implementation.



Cisco Unified CallManager is the IP-based PBX in an IP telephony solution. It acts as a call agent for IP phones and MGCP gateways and can also interact with H.323 or SIP devices using their protocols. For redundancy and load sharing, multiple Cisco Unified CallManager servers operate in a cluster. From an administration perspective, the whole cluster is a single logical instance. The main functions performed by Cisco Unified CallManager are these:

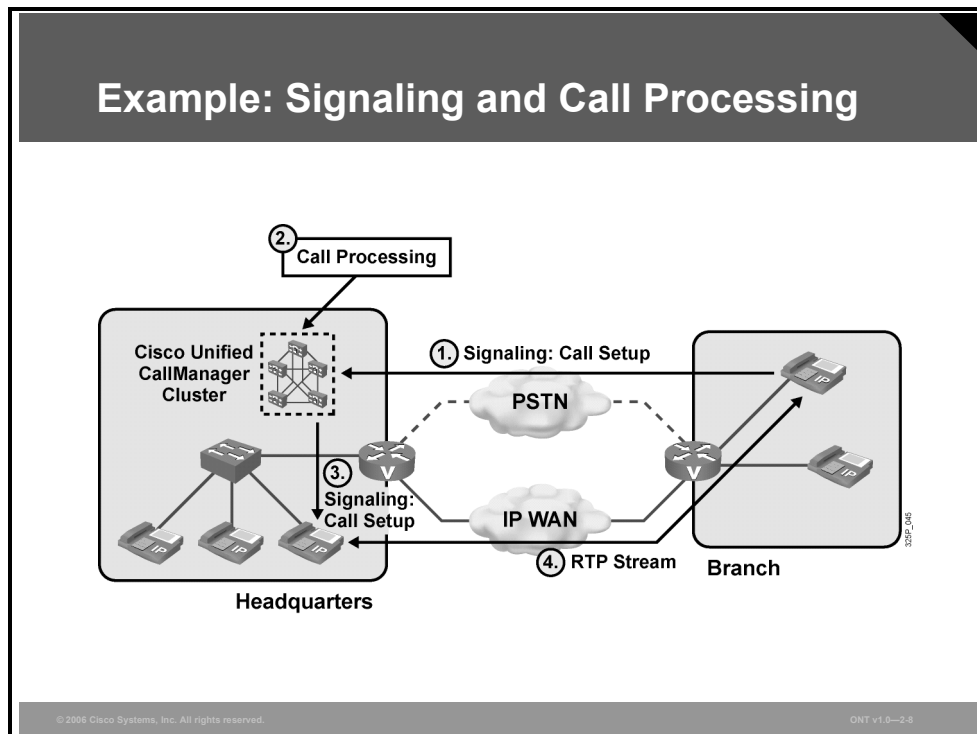
- **Call processing:** Cisco Unified CallManager processes calls between end devices and gateways. Call processing includes call routing decisions, signaling between the affected devices, and accounting of calls. In addition, class of service (CoS) and bandwidth management can be configured to influence call processing.
- **Dial plan administration:** Cisco Unified CallManager acts as a call agent for IP phones and MGCP gateways and thus eliminates the need for local call routing tables in these devices. Only the call agent (that is, Cisco Unified CallManager) needs to know the dial plan and, therefore, all dial plan administration is performed on Cisco Unified CallManager. H.323 and SIP devices follow the distributed call-processing model, and require locally available and administrated dial plans.
- **Signaling and device control:** In its role as a call agent, Cisco Unified CallManager controls IP phones and MGCP gateways by telling these devices what to do in certain events. For instance, when an IP phone informs Cisco Unified CallManager that the user went off hook, Cisco Unified CallManager tells the IP phone to update the screen and play a dial tone.
- **Phone feature administration:** The entire IP phone configuration is entered and stored at Cisco Unified CallManager. The IP phones load their configuration file during boot after a device reset. IP phone administration is fully centralized.

- **Directory and Extensible Markup Language (XML) services:** Cisco Unified CallManager provides access to directories. IP phones can be used to perform lookups in the available directories. Further, IP phones can use XML-based applications, accessible and configured as IP phone services.
- **Programming interface to external applications:** Through a programming interface, external application can be integrated with the Cisco Unified CallManager IP telephony solution. Examples of such applications are Cisco IP Communicator, Cisco IP Interactive Voice Response (IVR), Cisco Personal Assistant, and Cisco Unified CallManager Attendant Console. A variety of third-party products use Cisco Unified CallManager programming interfaces.



## Example of Cisco Unified CallManager Functions

The figure shows a company using Cisco Unified CallManager. The company has two sites: the headquarters and a branch. A Cisco Unified CallManager cluster is located in the headquarters. Each site has a voice gateway for PSTN access.



In the example, a user in the branch office wants to place a call to a user located at headquarters:

- Step 1** When the branch user dials the number, the IP phone sends signaling messages to a member of the Cisco Unified CallManager cluster.
- Step 2** The Cisco Unified CallManager server processes the call by looking up the called number in its call routing table.
- Step 3** When the Cisco Unified CallManager server determines the IP address of the destination phone, it sends a signaling message to the destination phone. The destination phone starts ringing, and the called user can accept the call.
- Step 4** After the call is accepted, the phones start sending and receiving RTP packets carrying audio signals.

# Enterprise IP Telephony Deployment Models

This topic describes the main IP telephony deployment models that may be used in an enterprise, including single site, multisite centralized, multisite distributed, and clustering over WAN.

Enterprise IP Telephony Deployment Models	
Deployment Model	Characteristics
Single site	<ul style="list-style-type: none"><li>• Cisco Unified CallManager cluster at the single site</li><li>• Local IP phones only</li></ul>
Multisite with centralized call processing	<ul style="list-style-type: none"><li>• Cisco Unified CallManager cluster only at a single site</li><li>• Local and remote IP phones</li></ul>
Multisite with distributed call processing	<ul style="list-style-type: none"><li>• Cisco Unified CallManager clusters at multiple sites</li><li>• Local IP phones only</li></ul>
Clustering over WAN	<ul style="list-style-type: none"><li>• Single Cisco Unified CallManager cluster, distributed over multiple sites</li><li>• Usually local IP phones only</li><li>• Round-trip delay between any pair of servers not to exceed 40 ms</li></ul>

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—2-10

Cisco Unified CallManager can be deployed using any of the following models:

- **Single site:** The IP telephony solution is deployed at a single site only. There is a local Cisco Unified CallManager cluster, which serves local phones only.
- **Multisite with centralized call processing:** The IP telephony solution is deployed at multiple sites. In the centralized call-processing deployment, there is a single Cisco Unified CallManager cluster, located at one of the sites. The Cisco Unified CallManager cluster serves local and remote IP phones.
- **Multisite with distributed call processing:** The IP telephony solution is deployed at multiple sites. In the distributed call-processing deployment, there is a Cisco Unified CallManager cluster at each site. Each Cisco Unified CallManager cluster serves local IP phones.
- **Clustering over WAN:** The IP telephony solution is deployed at multiple sites. Cisco Unified CallManager servers are located at more than one site. However, all of the Cisco Unified CallManager servers belong to a single cluster. The members of the cluster are separated by an IP WAN. IP phones usually use the local servers as their call agents.

---

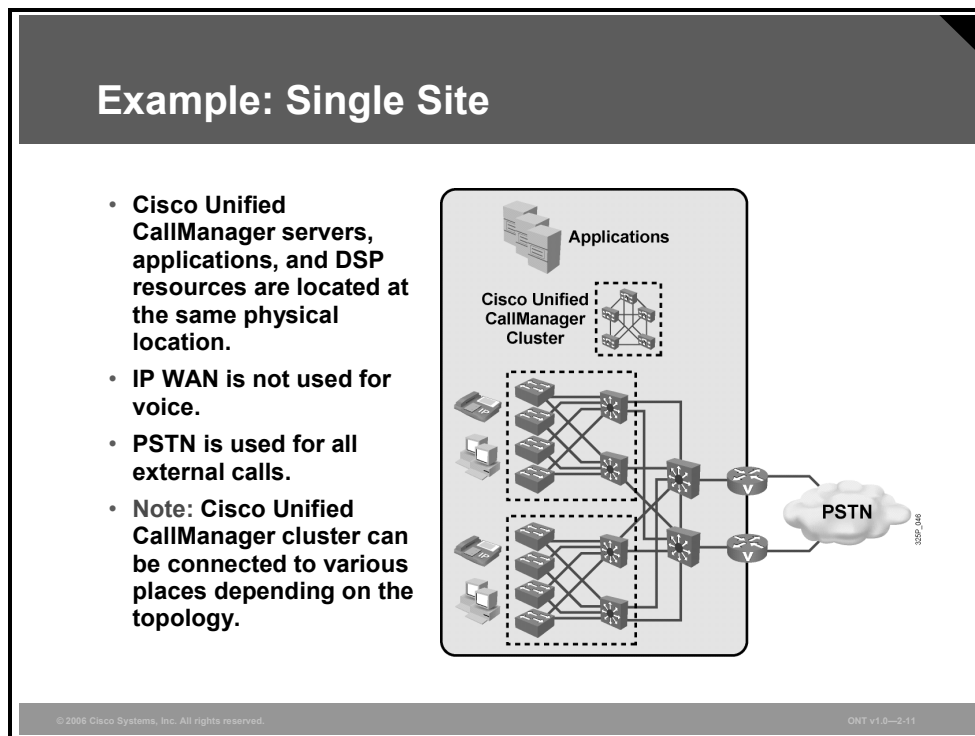
**Caution** The clustering-over-WAN deployment model requires the round-trip delay between any pair of servers to be less than 40 ms.

---

Hybrid deployments are also possible. For example, Cisco Unified CallManager clusters may reside at multiple sites, but there can also be sites that do not have a Cisco Unified CallManager cluster and are served by a cluster at a remote site.

## Example: Single Site

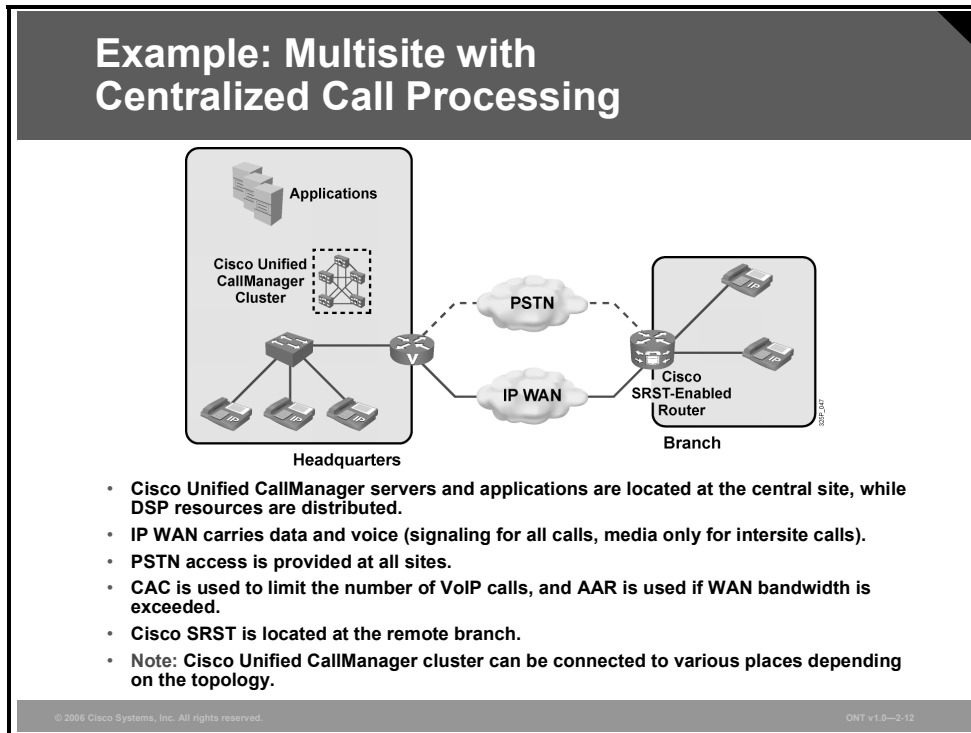
The figure shows an example of a single-site deployment of Cisco Unified CallManager.



In the example, IP telephony is deployed at a single building with a local Cisco Unified CallManager cluster and application servers. Local voice gateways offer PSTN access and provide DSP resources. The IP WAN is not used for voice; the PSTN is used for all external calls.

## Example: Multisite with Centralized Call Processing

This example shows a company with two sites using a centralized call-processing model.



In the example, there is one Cisco Unified CallManager cluster, and all its servers are located at the headquarters site. Application servers are also centralized, but each site has local DSP resources. Because there are local DSPs at each site, conference calls using the DSP resources do not have to cross the IP WAN if all participants of the conference are at the same site.

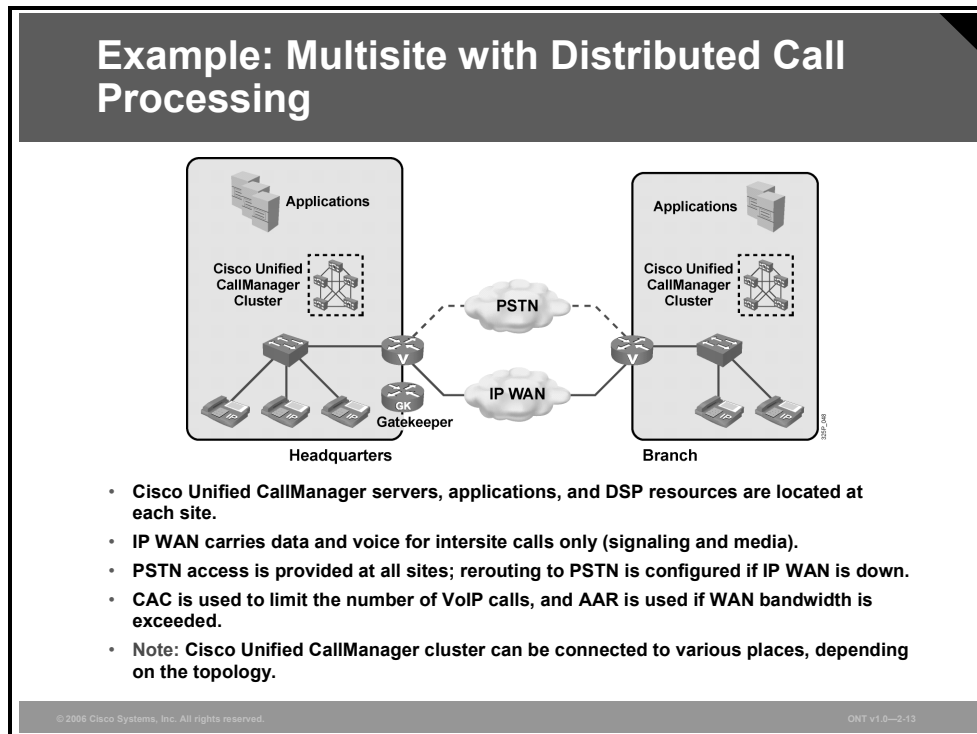
The IP WAN carries data and voice. All signaling messages from branch IP phones have to cross the IP WAN. Media streams cross the IP WAN only for intersite calls. Each site has local PSTN access.

CAC is used to limit the number of VoIP calls between the two sites. If calls are denied by CAC, they are rerouted through the PSTN using the Automated Alternate Routing (AAR) feature of Cisco Unified CallManager.

If the IP WAN goes down, the branch router uses Cisco SRST to maintain IP telephony operation for calls within the branch. All other calls are routed through the PSTN.

## Example: Multisite with Distributed Call Processing

In this example, the distributed call-processing deployment has been used for two sites.

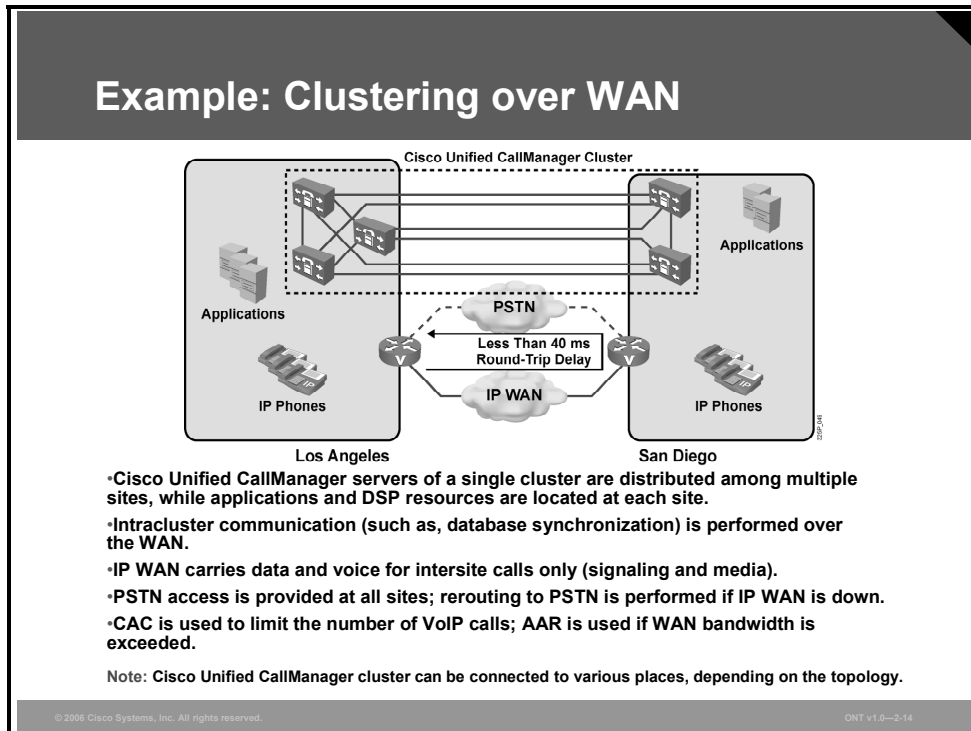


In the example, each site has its own Cisco Unified CallManager cluster with local servers. Local application servers and DSP resources are available at each site.

The IP WAN carries data and voice. Only intersite calls use the IP WAN. Both the signaling messages and media streams of intrasite calls remain local to each site, because of the presence of a call agent in each site. Both sites have local PSTN access. CAC is used to limit the number of VoIP calls between the two sites. If calls are denied by CAC, they are rerouted through the PSTN using AAR. If the IP WAN goes down, the local Cisco Unified CallManager clusters reroute intersite calls over the PSTN via their local gateways.

## Example: Clustering over WAN

This example shows clustering over an IP WAN.



In the example, each site has local Cisco Unified CallManager servers, but they all belong to a single cluster. Application servers and DSP resources are locally available at each site.

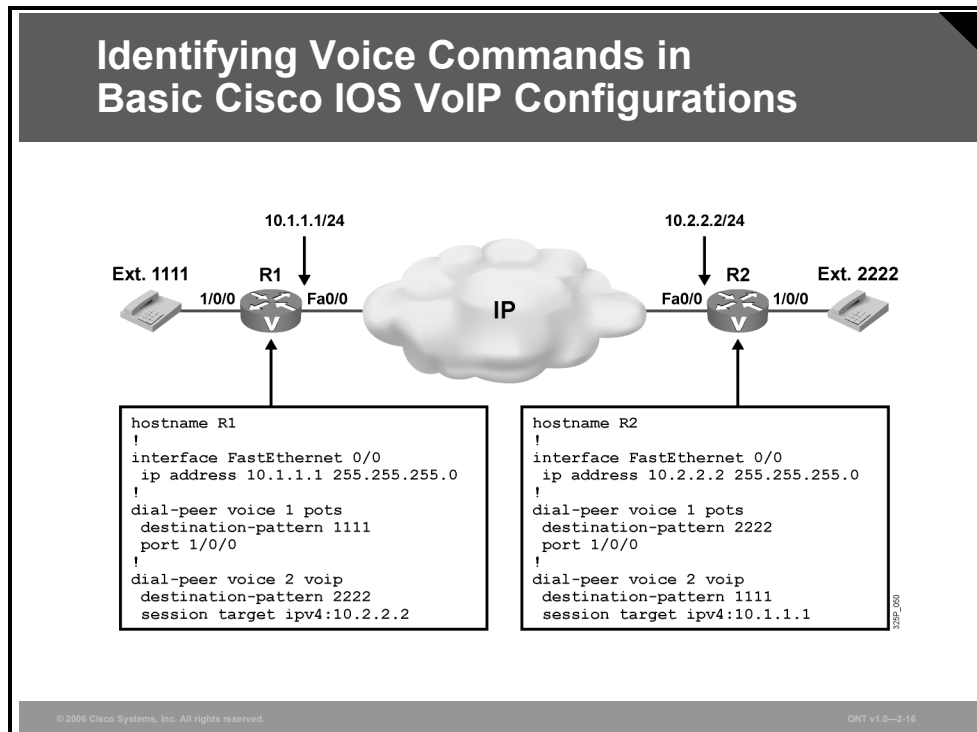
IP phones at the Los Angeles office use local Cisco Unified CallManager servers as their primary call agents, while the servers in San Diego are used as backup call agents. IP phones in San Diego are configured in a similar way, using the local Cisco Unified CallManager servers as their primary call agents and the servers in Los Angeles as backup.

The IP WAN carries data and voice. Only intersite calls use the IP WAN. Both the signaling messages and media streams of intrasite calls remain local to each site, because of the presence of a call agent at each site. Both sites have local PSTN access. CAC is used to limit the number of VoIP calls between the two sites. If calls are denied by CAC, they are rerouted through the PSTN by AAR. If the IP WAN goes down, the local Cisco Unified CallManager clusters reroute intersite calls over the PSTN using the local gateways.

All intracluster communication between Cisco Unified CallManager servers (for example, database synchronization or exchange of account data) goes over the IP WAN. The company must guarantee a required round-trip time of less than 40 ms for the intracluster traffic between the two sites.

# Identifying Voice Commands in Cisco IOS Configurations

This topic identifies the sections of the output of the **show running-config** command from a Cisco router configured as a voice gateway that are related to the voice implementation on the router.



Cisco IOS routers can be used as VoIP gateways. For a basic VoIP configuration, two gateways are needed. Both need a connection to a traditional telephony device, such as an analog telephone. The gateways themselves must have IP connectivity.

In the example, the first router is configured with these settings:

- **Name:** R1
- **IP address:** 10.1.1.1/24
- **IP interface:** FastEthernet 0/0
- **Voice port:** 1/0/0
- **Extension of the phone, connected to the voice port:** 1111

The second router is configured with similar settings:

- **Name:** R2
- **IP address:** 10.2.2.2/24
- **IP interface:** FastEthernet 0/0
- **Voice port:** 1/0/0
- **Extension of the phone, connected to the voice port:** 2222

Based on this information, this configuration is applied to the first router:

```
hostname R1
interface FastEthernet 0/0
 ip address 10.1.1.1 255.255.255.0
!
dial-peer voice 1 pots
 destination-pattern 1111
 port 1/0/0
!
dial-peer voice 2 voip
 destination-pattern 2222
 session target ipv4:10.2.2.2
!
```

The second router is configured with these commands:

```
hostname R2
interface FastEthernet 0/0
 ip address 10.2.2.2 255.255.255.0
!
dial-peer voice 1 pots
 destination-pattern 2222
 port 1/0/0
!
dial-peer voice 2 voip
 destination-pattern 1111
 session target ipv4:10.1.1.1
!
```

The voice-specific commands in the configurations (two dial peers in each configuration) are highlighted in gray. A dial peer describes where to find a telephone number; the total of all dial peers makes up the call routing table of a voice gateway. Two types of dial peers are shown in this example: plain old telephone service (POTS) dial peers and VoIP dial peers. POTS dial peers indicate that the telephone number specified in the dial peer is found at a physical port. A VoIP dial peer refers to the IP address of a VoIP device. The table lists the commands that are used.

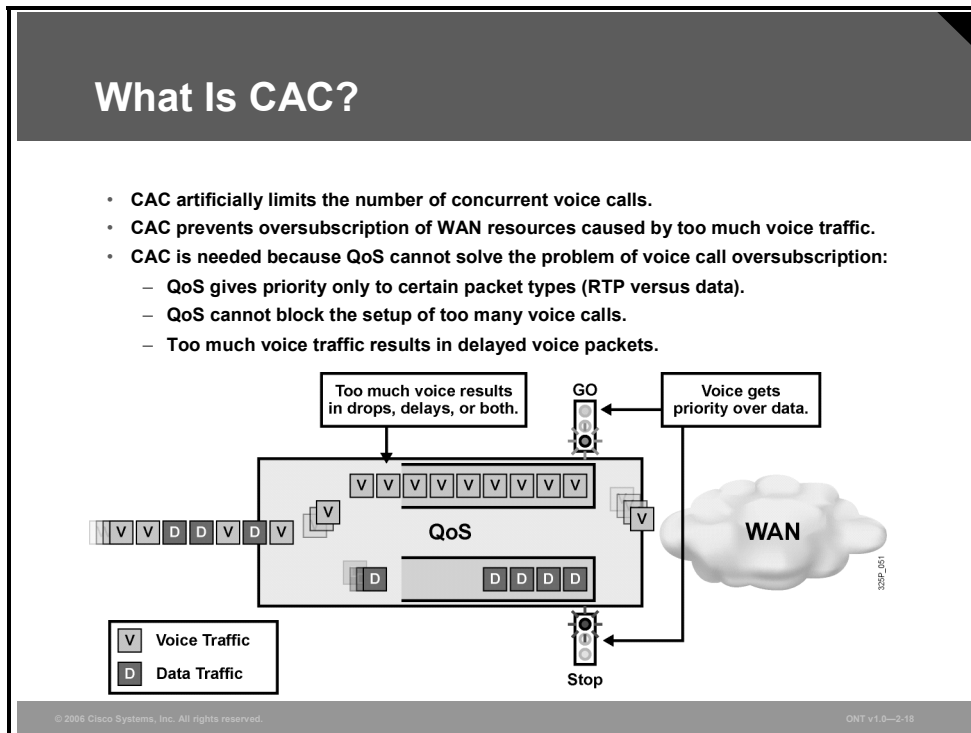


## Voice-Specific Commands

Command	Description
<b>dial-peer voice</b> <i>tag</i> <i>type</i>	Use the <b>dial-peer voice</b> command to enter the dial peer subconfiguration mode. The <i>tag</i> value is a number that has to be unique for all dial peers within the same gateway. The <i>type</i> value indicates the type of dial peer (for example, POTS or VoIP).
<b>destination-pattern</b> <i>telephone_number</i>	The <b>destination-pattern</b> command, entered in dial peer subconfiguration mode, defines the telephone number that applies to the dial peer. A call placed to this number will be routed according to the configuration type and port (in the case of a POTS type dial peer) or session target (in the case of a VoIP type dial peer) of the dial peer.
<b>port</b> <i>port-number</i>	The <b>port</b> command, entered in POTS dial peer subconfiguration mode, defines the port number that applies to the dial peer. Calls that are routed using this dial peer are sent to the specified port. The <b>port</b> command can be configured only on a POTS dial peer.
<b>session target</b> <b>ipv4:</b> <i>ip-address</i>	The <b>session target</b> command, entered in VoIP dial peer subconfiguration mode, defines the IP address of the target VoIP device that applies to the dial peer. Calls that are routed using this dial peer are sent to the specified IP address. The session target command can be configured only on a VoIP dial peer.

# What Is CAC?

This topic describes the purpose of CAC in VoIP networks.

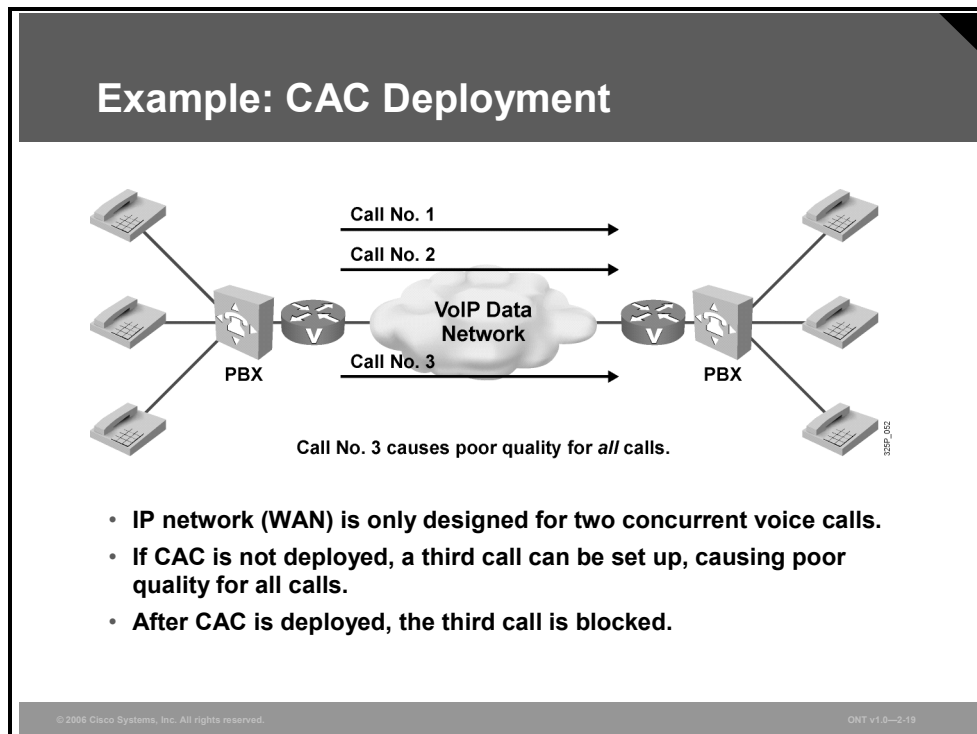


IP telephony solutions offer Call Admission Control (CAC), a feature that artificially limits the number of concurrent voice calls to prevent oversubscription of WAN resources.

Without CAC, if too many calls are active and too much voice traffic is sent, delays and packet drops occur. Even giving Real-Time Transport Protocol (RTP) packets absolute priority over all other traffic does not help when the physical bandwidth is not sufficient to carry all voice packets. Quality of service (QoS) mechanisms do not associate individual RTP packets with individual calls; therefore, *all* RTP packets are treated equally. *All* RTP packets will experience delays, and *any* RTP packets may be dropped. The effect of this behavior is that *all* voice calls experience voice quality degradation when oversubscription occurs. It is a common misconception that only calls that are beyond the bandwidth limit will suffer from quality degradation. CAC is the only method that prevents *general* voice quality degradation caused by too many concurrent active calls.

## Example: CAC Deployment

The example shows a scenario with two sites, both of which have three phones connected to a VoIP gateway through a PBX. The two gateways are connected via an IP network. The network is designed for a maximum of two concurrent calls.



Initially, CAC was not used. Whenever there were three active calls, all of them experienced severe voice quality issues.

CAC is deployed to avoid this problem. The gateways are configured to allow no more than two calls at the same time. When a third call is attempted, the call is blocked. With the new configuration using CAC, no voice quality problems should be experienced.

# Summary

This topic summarizes the key points that were discussed in this lesson.

## Summary

- **Converged enterprise networks include components supporting VoIP, such as gateways, gatekeepers, Cisco Unified CallManager, and IP phones.**
- **Cisco ISRs provide voice capabilities, including gateway, call agent, and DSP functions.**
- **Cisco Unified CallManager provides call processing and signaling services and provides access to applications from IP phones.**
- **IP deployment models include single site, multisite (centralized and distributed), and clustering over WAN.**
- **Cisco IOS dial peers are used at the gateway to configure a local dial plan.**
- **CAC is a method that prevents bandwidth exhaustion caused by too many voice calls.**

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—2.20

# Module Summary

This topic summarizes the key points that were discussed in this module.

## Module Summary

- **VoIP networks are composed of multiple components, using either distributed or centralized call control methods.**
- **In VoIP networks, analog signals have to be converted into digital format. DSPs provide this conversion by sampling, quantization, encoding, and optional compression.**
- **Digitized voice is encapsulated into RTP, UDP, and IP headers. To reduce bandwidth requirements, these headers can be compressed on a link-by-link basis.**
- **The total bandwidth required for a VoIP call depends on the codec, packetization period, and encapsulation overhead.**
- **Based on the network topology and size, different IP telephony deployment models can be utilized.**

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—2-1

VoIP networks include different components in order to be able to assure voice calls. First analog voice signals must be converted to digital format. DSP modules can be used to perform conversion by sampling, quantization and encoding. Optionally compression can be used too. Payload of the voice packet is small and RTP header compression is used as voice headers are significant part of the packet. RTP header compression is compressing IP RTP and UDP part of the header to reduce bandwidth. Bandwidth must take into account layer 2 header too and optionally additional overhead is added in case of encryption. Total bandwidth depends on different codecs used too. IP telephony deployments can be using distributed or centralized call control methods and different deployment models can be utilized.

# Module Self-Check

Use the questions here to review what you learned in this module. The correct answers and solutions are found in the Module Self-Check Answer Key.

- Q1) Which is not a benefit of packet telephony networks? (Source: Introducing VoIP Networks)
- A) lower transmission costs
  - B) access to new communication devices
  - C) improved productivity
  - D) less packetization overhead
- Q2) A \_\_\_\_\_ is used for mixing audio streams in conference calls. (Source: Introducing VoIP Networks)
- 
- Q3) Which stage of a call involves admission control? (Source: Introducing VoIP Networks)
- A) call setup
  - B) call blocking
  - C) call teardown
  - D) call maintenance
- Q4) Which protocol is an example of centralized call control? (Source: Introducing VoIP Networks)
- A) RSVP
  - B) H.235
  - C) H.323
  - D) MGCP
- Q5) Cisco IOS voice gateways use the \_\_\_\_\_ interface when connecting to an analog phone. (Source: Introducing VoIP Networks)
- 
- Q6) Which two statements are true about digital interfaces? (Choose two.) (Source: Introducing VoIP Networks)
- A) T1 CAS has 24 clear 64-kbps voice channels.
  - B) E1 CCS has 30 voice channels.
  - C) E1 CAS has 30 voice channels.
  - D) T1 CCS has 24 clear 64-kbps voice channels.
  - E) T1 CCS has 23 voice channels with robbed-bit signaling.
  - F) E1 CCS has 31 voice channels.
- Q7) Which is the correct order for analog-to-digital conversion? (Source: Digitizing and Packetizing Voice)
- A) compression, sampling, encoding, quantization
  - B) encoding, sampling, quantization, compression
  - C) sampling, encoding, quantization, compression
  - D) sampling, quantization, encoding, compression

- Q8) During digital-to-analog conversion, what is reconstructed from the PAM signals? (Source: Digitizing and Packetizing Voice)
- A) PCM
  - B) analog signal
  - C) digital signal
  - D) codeword
- Q9) According to the Nyquist theorem, what minimum rate is needed to sample analog frequencies up to 9000 Hz? (Source: Digitizing and Packetizing Voice)
- A) 8000 Hz
  - B) 8 kHz
  - C) 18 kHz
  - D) 4500 Hz
- Q10) During encoding, how many bits are used to indicate the segment number? (Source: Digitizing and Packetizing Voice)
- 
- Q11) Which statement is true about codec bit rates? (Source: Digitizing and Packetizing Voice)
- A) G.729 and G.728 both use 8 kbps.
  - B) G.729A uses 8 kbps, and G.711 uses 16, 24, or 32 kbps.
  - C) G.728 uses 16 kbps.
  - D) G.711 and G.276 both use 64 kbps.
- Q12) A DSP used for a \_\_\_\_\_-mode conference cannot mix different codecs. (Source: Digitizing and Packetizing Voice)
- 
- Q13) Which is not a characteristic of VoIP packet delivery? (Source: Encapsulating Voice Packets for Transport)
- A) Packets can arrive in incorrect order.
  - B) Packets experience varying delays.
  - C) VoIP packets are sent over a dedicated circuit of 64 kbps.
  - D) The total bandwidth of a link is shared by all IP packets.
- Q14) Which header is not used for VoIP media packets? (Source: Encapsulating Voice Packets for Transport)
- A) TCP
  - B) IP
  - C) UDP
  - D) RTP
- Q15) Which headers does an RTP header compression reduce to 2 or 4 bytes? (Source: Encapsulating Voice Packets for Transport)
- A) IP, UDP, and RTP
  - B) IP and TCP
  - C) RTP
  - D) UDP and RTP

- Q16) Which two statements are true of packetization of voice? (Choose two.) (Source: Calculating Bandwidth Requirements)
- A) The shorter the packetization period, the smaller the IP packet size.
  - B) The longer the packetization period, the smaller the IP packet size.
  - C) The longer the packetization period, the lower the packet rate.
  - D) The shorter the packetization period, the lower the packet rate.
  - E) The longer the packetization period, the lower the codec bandwidth.
- Q17) Which data-link protocol adds 18 bytes during encapsulation to the VoIP packet? (Source: Calculating Bandwidth Requirements)
- 

- Q18) Which is the approximate amount of overhead added to a 40-byte VoIP packet by IPsec tunnel mode? (Source: Calculating Bandwidth Requirements)
- A) more than 300 percent
  - B) more than 35 percent
  - C) more than 100 percent
  - D) up to 35 percent
- Q19) Which is not a factor for the calculation of total bandwidth? (Source: Calculating Bandwidth Requirements)
- A) packetization period
  - B) packetization size
  - C) codec bandwidth
  - D) encapsulation overhead
  - E) buffer overhead
- Q20) \_\_\_\_\_ can save bandwidth by suppressing transmission during periods of silence. (Source: Calculating Bandwidth Requirements)
- 

- Q21) Which is not a component of an enterprise voice network? (Source: Implementing Voice Support in an Enterprise Network)
- A) gateways
  - B) central office switch
  - C) IP phones
  - D) Cisco Unified CallManager
- Q22) Which telephony feature supports fallback scenarios for remote IP phones when the IP WAN is down? (Source: Implementing Voice Support in an Enterprise Network)
- A) VAD
  - B) AAR
  - C) SRST
  - D) cRTP



- Q23) Which is not a Cisco Unified CallManager function? (Source: Implementing Voice Support in an Enterprise Network)
- A) providing DSPs for transcoding
  - B) call processing
  - C) signaling and device control
  - D) phone feature administration
- Q24) Which Cisco Unified CallManager deployment model causes all signaling traffic to cross the IP WAN? (Source: Implementing Voice Support in an Enterprise Network)
- A) single site
  - B) multisite centralized call processing
  - C) multisite distributed call processing
  - D) clustering over WAN
- Q25) Which command is not used when configuring a POTS dial peer? (Source: Implementing Voice Support in an Enterprise Network)
- A) **dial-peer**
  - B) **port**
  - C) **session target**
  - D) **destination-pattern**
- Q26) CAC limits the \_\_\_\_\_ in a VoIP environment. (Source: Implementing Voice Support in an Enterprise Network)
-

## Module Self-Check Answer Key

- Q1) D
- Q2) multipoint control unit
- Q3) A
- Q4) D
- Q5) FXS
- Q6) B, C
- Q7) D
- Q8) B
- Q9) C
- Q10) three
- Q11) C
- Q12) single
- Q13) C
- Q14) A
- Q15) A
- Q16) B, C
- Q17) Ethernet
- Q18) C
- Q19) E
- Q20) VAD
- Q21) B
- Q22) C
- Q23) A
- Q24) B
- Q25) C
- Q26) number of calls

# Introduction to IP QoS

---

## Overview

As user applications continue to drive network growth and evolution, demand for support of different types of traffic is also increasing. Applications with differing network requirements create a need for administrative policies that mandate how individual applications are to be treated by the network. Network traffic from business-critical applications must be protected. Requests from business-critical and delay-sensitive applications must be serviced with priority. The employment and enforcement of quality of service (QoS) policies within a network plays an essential role in enabling network administrators and architects to meet networked application demands in converged networks.

This module introduces the concept of QoS, explains key issues of networked applications, lists models for providing QoS in the network (best effort, Integrated Services [IntServ], and Differentiated Services [DiffServ]), and describes various methods for implementing QoS, including the Cisco Modular QoS CLI (MQC) and Cisco Router and Security Device Manager (SDM) QoS Wizard.

## Module Objectives

Upon completing this module, you will be able to describe the need to implement QoS and the methods for implementing QoS on a converged network using Cisco routers and Cisco Catalyst switches. This ability includes being able to meet these objectives:

- Describe the conditions and nature of traffic in enterprise networks that lead to QoS problems and explain the IP QoS mechanisms and Cisco QoS best practices that ensure the best possible network performance
- Explain the use of the three models for providing QoS in a network
- Explain how to implement QoS policies using both MQC and Cisco SDM QoS Wizard



# Introducing QoS

---

## Overview

Networks must provide secure, predictable, measurable, and, sometimes, guaranteed services. Network administrators and architects can better achieve this performance from the network by managing delay, delay variation (jitter), bandwidth provisioning, and packet loss parameters with quality of service (QoS) techniques.

This lesson introduces the concept of QoS and issues that arise with a converged network. The lesson identifies the four problems that could lead to poor QoS and the solutions to those problems.

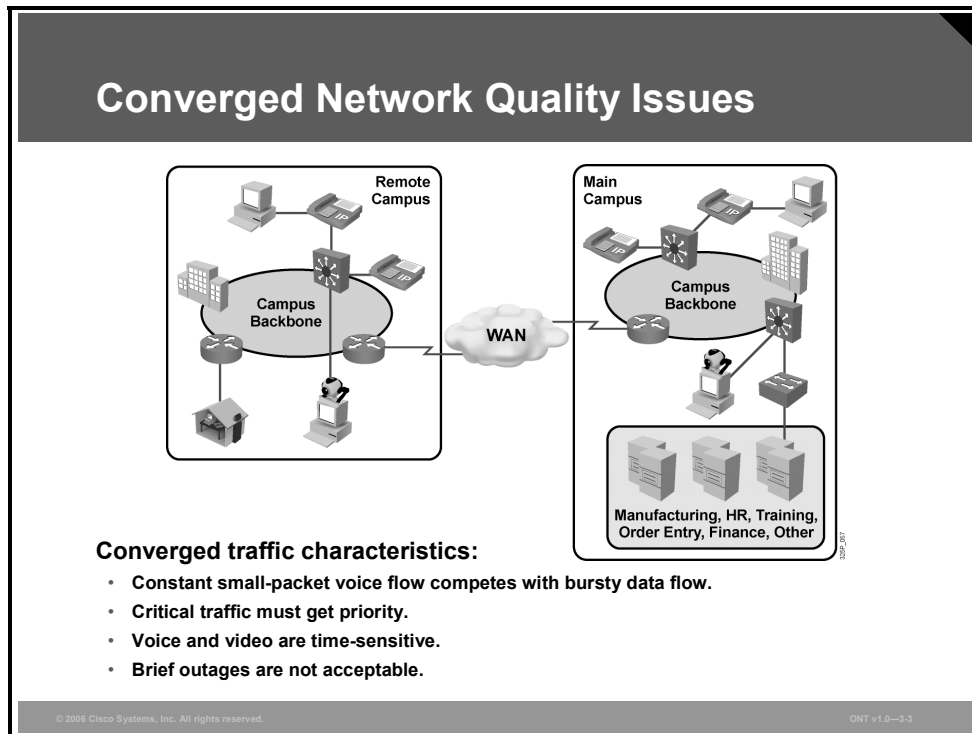
## Objectives

Upon completing this lesson, you will be able to describe the conditions and nature of traffic in enterprise networks that lead to QoS problems and explain the IP QoS mechanisms and Cisco QoS best practices that ensure the best possible network performance. This ability includes being able to meet these objectives:

- Describe the four key quality issues with converged networks
- Explain how a lack of bandwidth can adversely impact QoS in a network and ways to effectively increase bandwidth on a link
- Explain how end-to-end delay can adversely impact QoS in a network and ways to effectively reduce delay
- Explain how packet loss can adversely impact QoS in a network and ways to manage packet loss so that QoS is not affected
- Define QoS with respect to traffic in a network
- Explain the three key steps in implementing a QoS policy on a network
- Describe how traffic is recognized by type in a network and how those types resolve to QoS traffic classes
- Describe how to define QoS policies after traffic classes have been defined

# Converged Network Quality Issues

This topic describes the four key quality-related characteristics of converged networks: bandwidth, end-to-end delay, variation of delay, and packet loss.



The figure illustrates a converged network in which voice, video, and data traffic use the same network facilities. Merging these traffic streams with dramatically differing requirements can lead to a number of problems.

Although packets that carry voice traffic are typically very small, they cannot tolerate delay and delay variation as they traverse the network. Voices will break up, and words will become incomprehensible.

On the other hand, packets that carry file-transfer data are typically large and can survive delays and drops. It is possible to retransmit part of a dropped data file, but it is not feasible to retransmit a part of a voice conversation.

The constant, small-packet voice flow competes with bursty data flows. Unless some mechanism mediates the overall flow, voice quality will be severely compromised at times of network congestion. The critical voice traffic must get priority.

Voice and video traffic is very time-sensitive. It cannot be delayed or dropped, or the quality of voice and video will suffer.

Finally, converged networks cannot fail. A file transfer or an e-mail packet can wait until the network recovers, but voice and video packets cannot. Even a brief network outage on a converged network can seriously disrupt business operations.

## Converged Network Quality Issues (Cont.)

- **Lack of bandwidth:** Multiple flows compete for a limited amount of bandwidth.
- **End-to-end delay (fixed and variable):** Packets have to traverse many network devices and links that add up to the overall delay.
- **Variation of delay (jitter):** Sometimes there is a lot of other traffic, which results in increased delay.
- **Packet loss:** Packets may have to be dropped when a link is congested.

© 2006 Cisco Systems, Inc. All rights reserved.

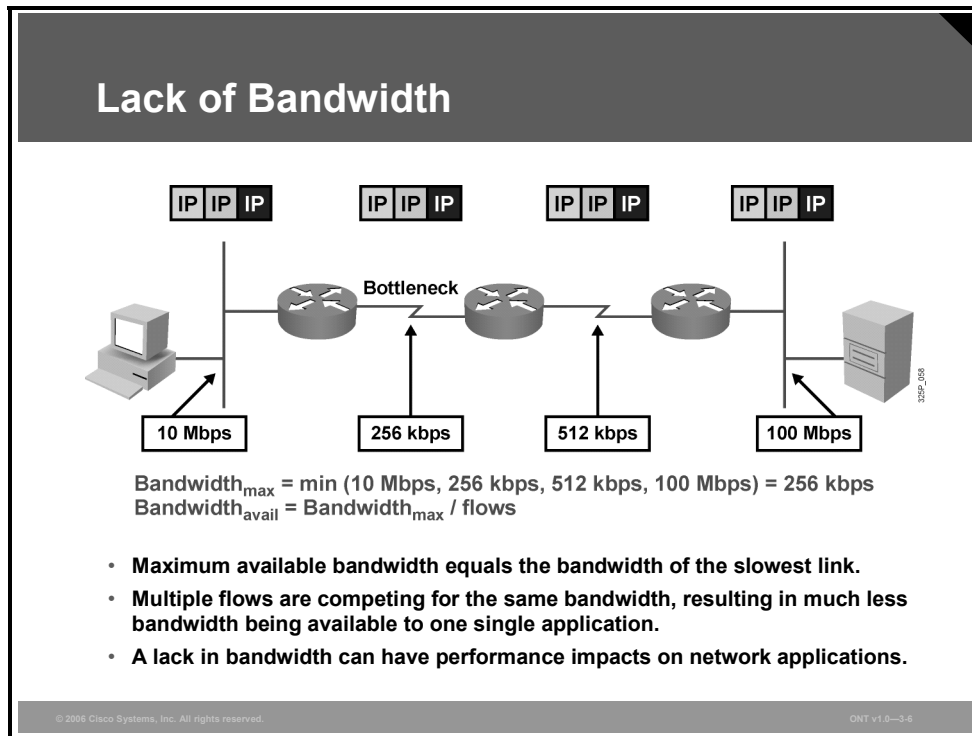
ONT v1.0—3-4

The four major issues that face converged enterprise networks are:

- **Bandwidth capacity:** Large graphics files, multimedia uses, and increasing use of voice and video cause bandwidth capacity problems over data networks.
- **End-to-end delay (fixed and variable components):** End-to-end delay is the time that it takes for a packet to reach the receiving endpoint after being transmitted from the sending endpoint.
- **Variation of delay (also called “jitter”):** Jitter is the difference in the arrival times of packets belonging to the same stream in a total end-to-end communication. The reduction of jitter is very important in VoIP transmissions.
- **Packet loss:** Loss of packets is usually caused by congestion in the WAN, which results in speech dropouts during VoIP calls and slow or even cut-off file transfers.

# Available Bandwidth

This topic describes how inadequate bandwidth can adversely impact QoS in a network and describes ways to effectively increase bandwidth on a link.



The figure illustrates an empty network with four hops between a server and a client. Each hop is using different media with different bandwidths. The maximum available bandwidth is equal to the bandwidth of the slowest link:

$$\text{Bandwidth}_{\text{max}} = \min (10 \text{ Mbps}, 256 \text{ kbps}, 512 \text{ kbps}, 100 \text{ Mbps}) = 256 \text{ kbps}$$

The calculation of the available bandwidth, however, is much more complex in cases where multiple flows are traversing the network. The average bandwidth available per flow can be calculated with this formula:

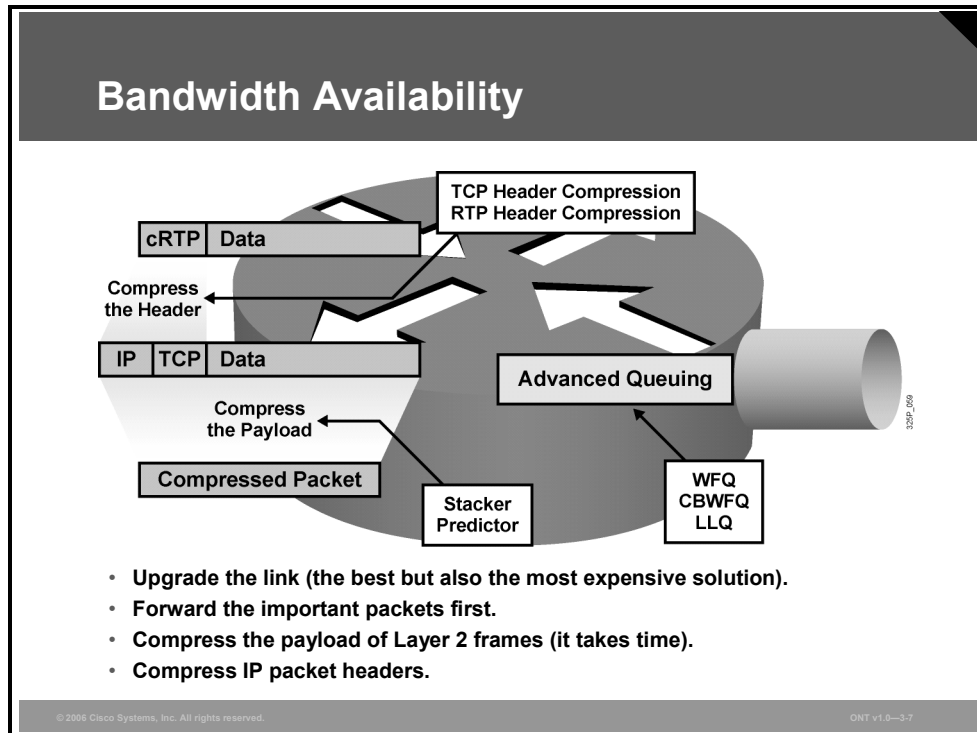
$$\text{Bandwidth}_{\text{avail}} = \text{Bandwidth}_{\text{max}} / \text{flows}$$

Inadequate bandwidth can have performance impacts on network applications; especially those that are time-sensitive (such as voice) or consume a lot of bandwidth (for example, videoconferencing). These impacts result in poor voice and video quality. Also, interactive network services, such as terminal services and remote desktops, may also suffer from lower bandwidth, which results in slow application responses.



# Bandwidth Availability

Availability of bandwidth is one of the factors that affect the quality of a network. The maximum available bandwidth is equal to the bandwidth of the slowest link.



The best way to increase bandwidth is to increase the link capacity to accommodate all applications and users, with some extra bandwidth to spare. Although this solution sounds simple, increasing bandwidth is expensive and takes time to implement. There are often technological limitations in upgrading to a higher bandwidth.

Another option is to classify traffic into quality of service (QoS) classes and prioritize it according to importance. The basic queuing mechanism is FIFO. Other queuing mechanisms provide additional granularity to serve voice and business-critical traffic. Such traffic types should receive sufficient bandwidth to support their application requirements. Voice traffic should receive prioritized forwarding, and the least important traffic should receive whatever unallocated bandwidth remains. A variety of mechanisms are available in Cisco IOS QoS software that provide bandwidth priority to specific classes of traffic:

- Weighted fair queuing (WFQ)
- Class-based weighted fair queuing (CBWFQ)
- Low latency queuing (LLQ)

Optimizing link usage by compressing the payload of frames (virtually) increases link bandwidth. Compression, however, also increases delay because of the complexity of compression algorithms. Using hardware compression can accelerate packet payload compression. Stacker and Predictor are two compression algorithms available in Cisco IOS software.

Another link efficiency mechanism is header compression. This mechanism is especially effective in networks where most packets carry small amounts of data (that is, where the payload-to-header ratio is small). Typical examples of header compression are TCP header compression and Real-Time Transport Protocol (RTP) header compression.

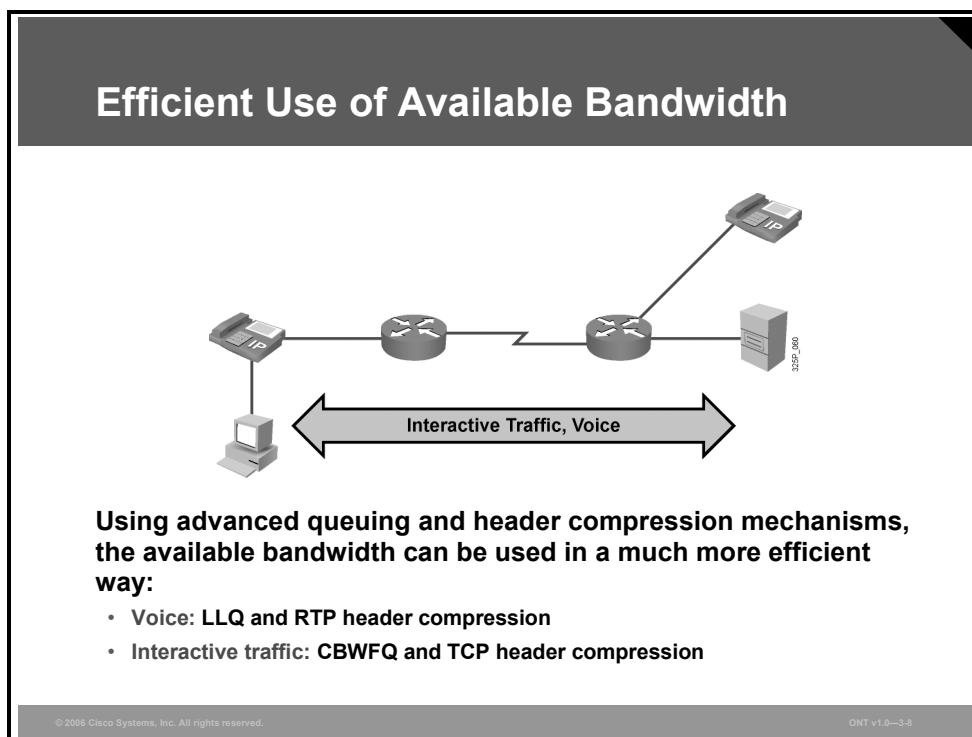
---

<b>Note</b>	Payload compression is always end-to-end compression, and header compression is hop-by-hop compression.
-------------	---

---

## Example: Efficient Use of Available Bandwidth

In this scenario, two office sites are connected over a low-speed WAN link. Both sites are equipped with IP phones, PCs, and servers that run interactive applications, such as terminal services. Because the available bandwidth is limited, an appropriate strategy for efficient bandwidth usage must be determined.



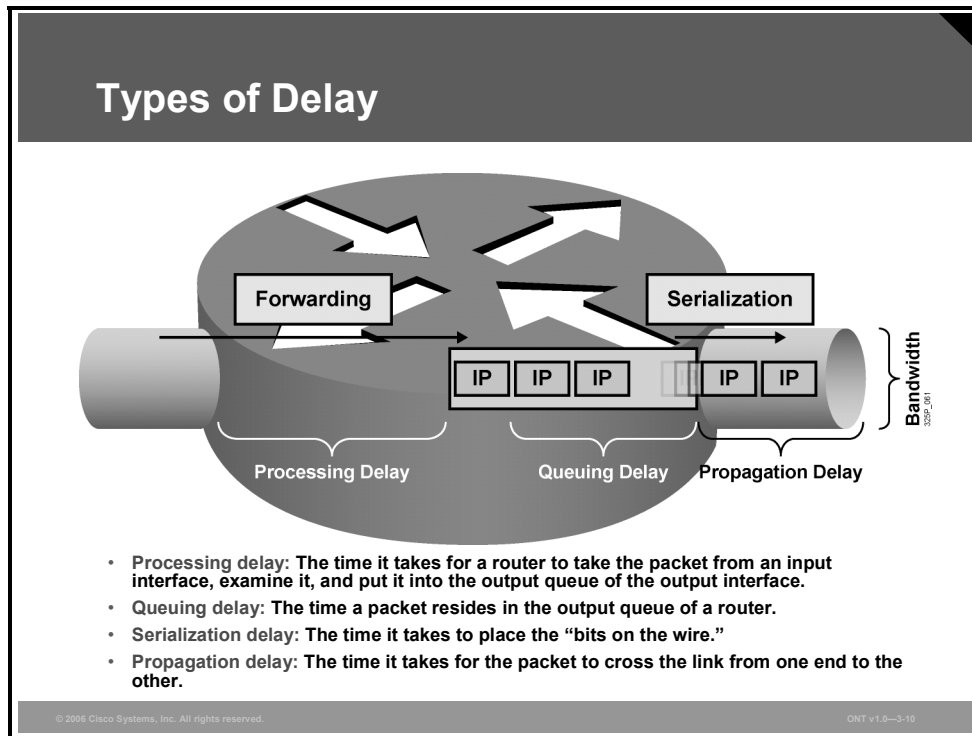
In a network with remote sites, where interactive traffic and voice are used for daily business, bandwidth availability is an issue.

In some regions, broadband bandwidth services are difficult to obtain or, in the worst case, not available. This situation means that available bandwidth resources must be efficiently used. Advanced queuing techniques, such as CBWFQ or LLQ, and header compression mechanisms, such as TCP and RTP header compression, are needed to use the bandwidth much more efficiently.

Depending on the kind of traffic traversing the network, suitable queuing and compression mechanisms must be selected. LLQ and RTP header compression are used to provide the optimal quality for voice traffic. CBWFQ and TCP header compression are effective for managing interactive data traffic.

# End-to-End Delay

This topic describes how end-to-end delay can adversely affect QoS in a network and describes ways to effectively reduce delay.



There are four types of delay:

- **Processing delay:** The time that it takes for a router (or Layer 3 switch) to take the packet from an input interface and put it into the output queue of the output interface. The processing delay depends on various factors:
  - CPU speed
  - CPU utilization
  - IP switching mode
  - Router architecture
  - Configured features on both the input and output interfaces

---

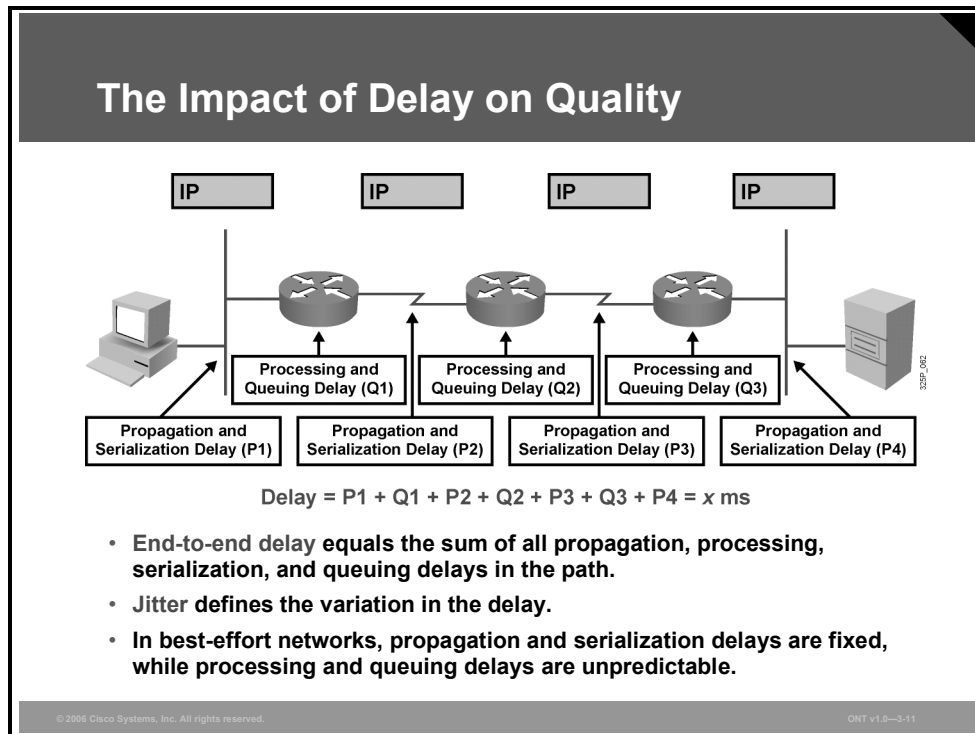
**Note** Many high-end routers or Layer 3 switches use advanced hardware architectures that speed up the packet processing and thus do not require the main CPU to process the packets.

---

- **Queuing delay:** The time that a packet resides in the output queue of a router. Queuing delay depends on the number of packets already in the queue and their sizes. Queuing delay also depends on the bandwidth of the interface and the queuing mechanism.
- **Serialization delay:** The time that it takes to place a frame on the physical medium for transport. This delay is typically inversely proportional to the link bandwidth.
- **Propagation delay:** The time that it takes for the packet to cross the link from one end to the other. This time usually depends on the type of media. (For example, satellite links produce the longest propagation delay because of the high altitudes of communications satellites.)

# The Impact of Delay on Quality

Another quality impact on the network is caused by end-to-end delay and jitter.



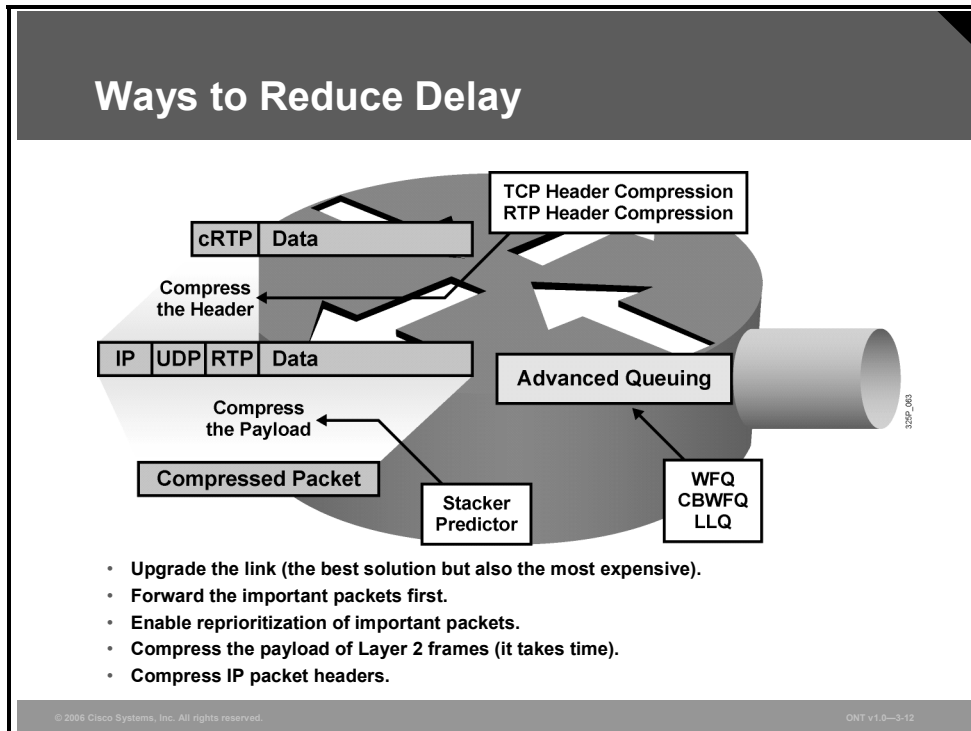
End-to-end delay of packets is the sum of all types of delays.

Another factor is jitter as packets traverse a network. Each hop in the network has its own set of variable processing and queuing delays, which can result in jitter. Processing and queuing delays are related to devices and are bound to the behavior of the operating system. Propagation and serialization delays are related to the media.

Propagation delay is generally ignored, but it can be significant—for example, about 40 ms coast to coast in North America over an optical link. Internet Control Message Protocol (ICMP) echo (ping) is one way to measure the round-trip time of IP packets in a network.

# Ways to Reduce Delay

There are many ways to reduce the delay on routers.



Assuming that the router being used is powerful enough to make forwarding decisions rapidly, most queuing and serialization delays are influenced by these factors:

- Average length of the queue
- Average length of packets in the queue
- Link bandwidth

There are several approaches for accelerating the packet dispatching of delay-sensitive flows:

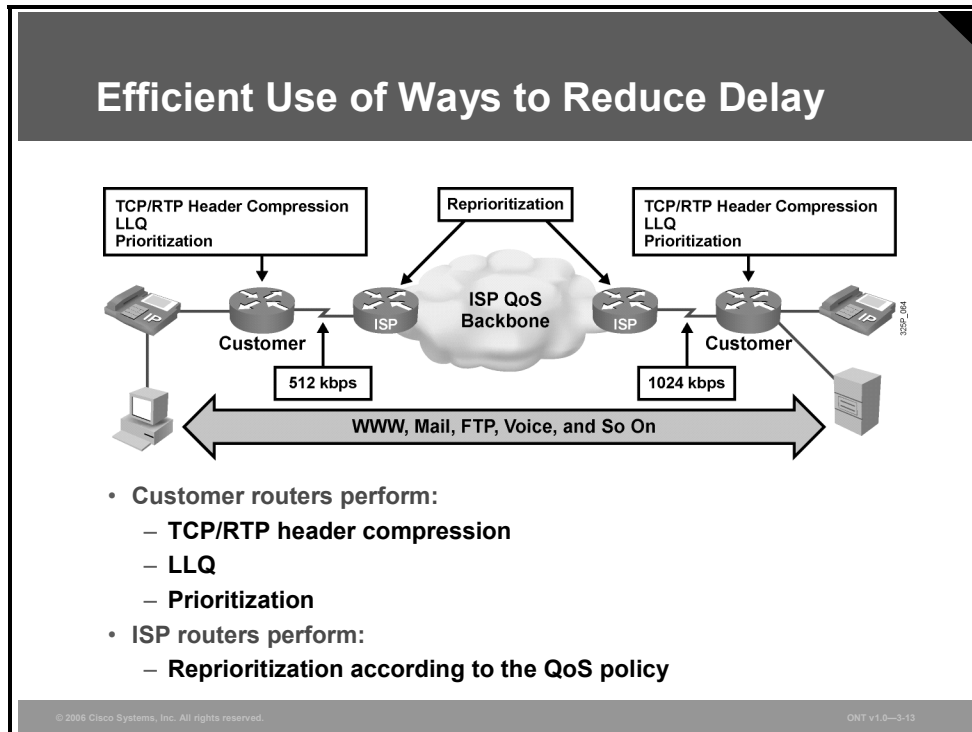
- **Increase link capacity:** Sufficient bandwidth causes queues to shrink so that packets do not wait long before transmittal. Increasing bandwidth reduces serialization time. This approach can be unrealistic because of the costs that are associated with the upgrade.
- **Prioritize delay-sensitive packets:** This approach can be more cost-effective than increasing link capacity. WFQ, CBWFQ, and LLQ can each serve certain queues first (a pre-emptive way of servicing queues).
- **Reprioritize packets:** In some cases, important packets need to be reprioritized when they are entering or exiting a device. For example, when packets leave a private network to transit an Internet service provider (ISP) network, the ISP may require that the packets be reprioritized.
- **Compress payload:** Payload compression reduces the size of packets, virtually increasing link bandwidth. Compressed packets are smaller and take less time to transmit. Compression uses complex algorithms that add delay. If you are using payload compression to reduce delay, make sure that the time needed to compress the payload does not negate the benefits of having less data to transfer over the link.

- **Use header compression:** Header compression is not as CPU-intensive as payload compression and is used with other mechanisms to reduce delay. Header compression is especially useful for voice packets that have a bad payload-to-header ratio (relative large header in comparison to the payload), which is improved by reducing the header of the packet (RTP header compression).

By minimizing delay, network administrators can also reduce jitter (delay is more predictable).

## Example: Efficient Use of Ways to Reduce Delay

In this scenario, the offices of the customer are connected via an ISP that supports QoS. The branch office of the customer is connected via a low-speed link (512 kbps), while the main office is connected with a higher-speed link (1024 kbps). The customer uses both IP phones and TCP/IP-based applications to conduct daily business. Because bandwidth of only 512 kbps is provided to the branch office, an appropriate QoS strategy must be implemented to produce the highest possible quality for voice and data traffic.

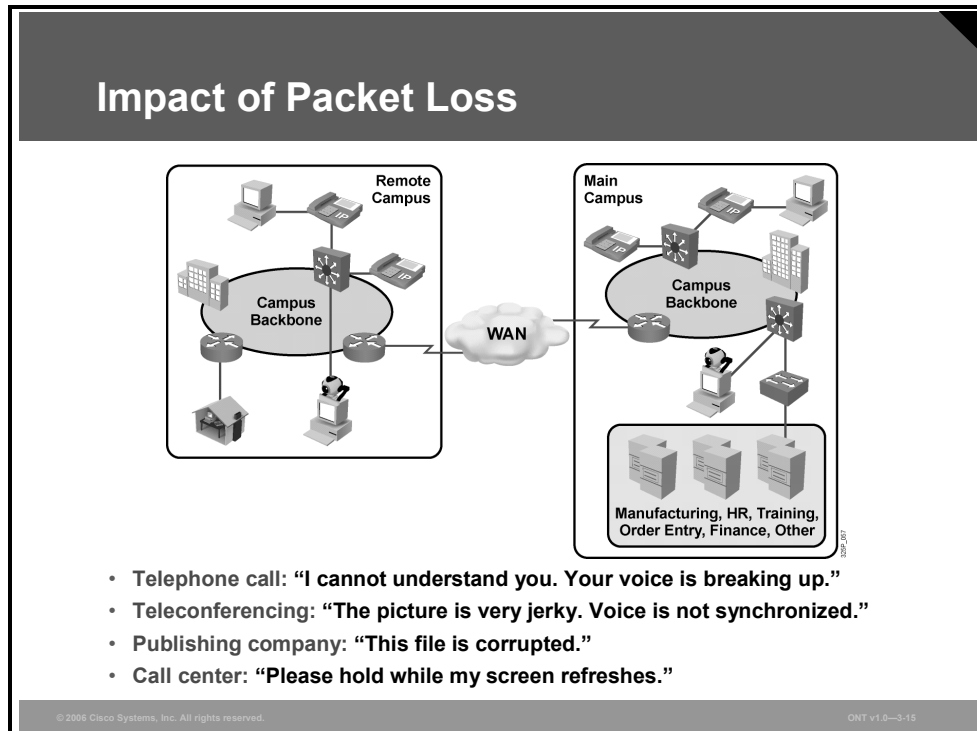


In this example, the customer needs to communicate with HTTP, FTP, e-mail, and voice services in the main office. Because the available bandwidth at the customer site is only 512 kbps, most traffic, but especially voice traffic, would suffer from end-to-end delays. In this example, the customer performs TCP and RTP header compression, LLQ, and prioritization of the various types of traffic. These mechanisms will give voice traffic a higher priority than HTTP or e-mail traffic. In addition to these measures, the customer has chosen an ISP that supports QoS in the backbone. The ISP performs reprioritization for customer traffic according to the QoS policy for the customer, so that the traffic streams arrive on time at the main office of the customer. This design guarantees that voice traffic will have high priority and a guaranteed bandwidth of 128 kbps; FTP and e-mail traffic will receive medium priority and a bandwidth of 256 kbps; and HTTP traffic will receive low priority and a bandwidth of 64 kbps. The remaining 64 kbps is needed for signaling and other management traffic.



# Packet Loss

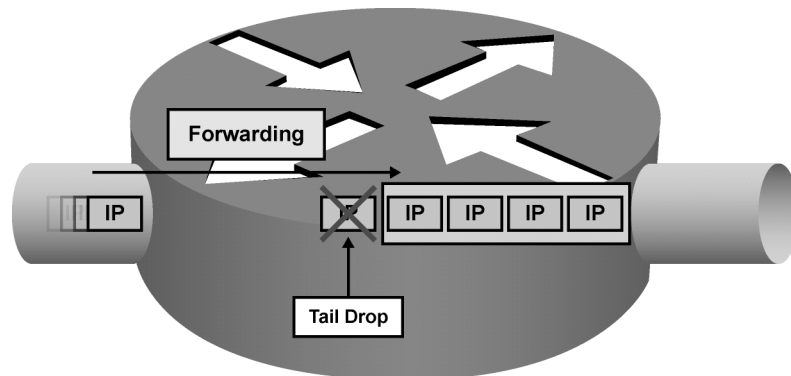
This topic describes how packet loss can adversely impact QoS in a network and describes ways to manage packet loss so that QoS is not affected.



A further issue in networks is packet loss. Usually, packet loss occurs when routers run out of buffer space for a particular interface (output queue). The figure illustrates the results of packet loss. Packet loss results in loss of information.

Multimedia streams, such as those used in IP telephony or videoconferencing, may be extremely sensitive to delivery delays and may create unique QoS demands on the underlying networks. When packets are delivered using the best-effort delivery model, they may not arrive in order or in a timely manner, or, because of heavy congestion, they may not arrive at all. The result would be an unclear picture, with jerky and slow movement and sound that is out of synchronization with the image.

## Impact of Packet Loss (Cont.)



- Tail drops occur when the output queue is full. Tail drops are common and happen when a link is congested.
- Many other types of drops occur, usually the result of router congestion, that are uncommon and may require a hardware upgrade (such as, input drop, ignore, overrun, frame errors).

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0-3-16

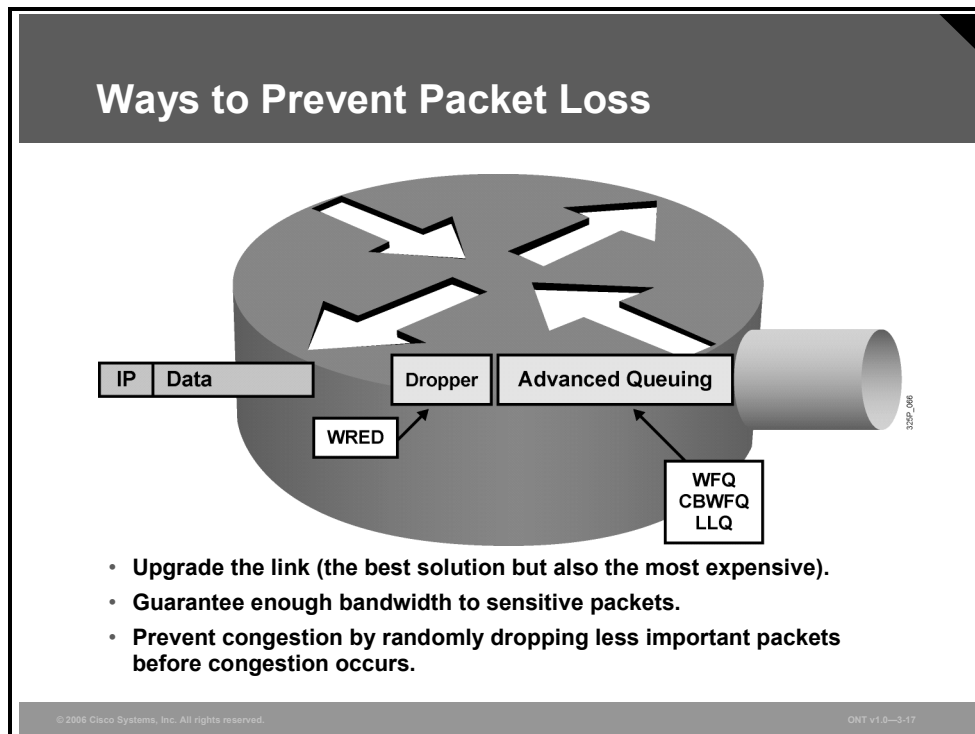
This figure illustrates a full interface output queue, which causes newly arriving packets to be dropped. The term that is used for such drops is simply “output drop” or “tail drop” (packets are dropped at the tail of the queue).

Routers might also drop packets for other, less common reasons:

- **Input queue drop:** The main CPU is busy and cannot process packets (the input queue is full).
- **Ignore:** The router runs out of buffer space.
- **Overrun:** The CPU is busy and cannot assign a free buffer to the new packet.
- **Frame errors:** The hardware detected an error in a frame; for example, cyclic redundancy checks (CRCs), runt, and giant.

# Ways to Prevent Packet Loss

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

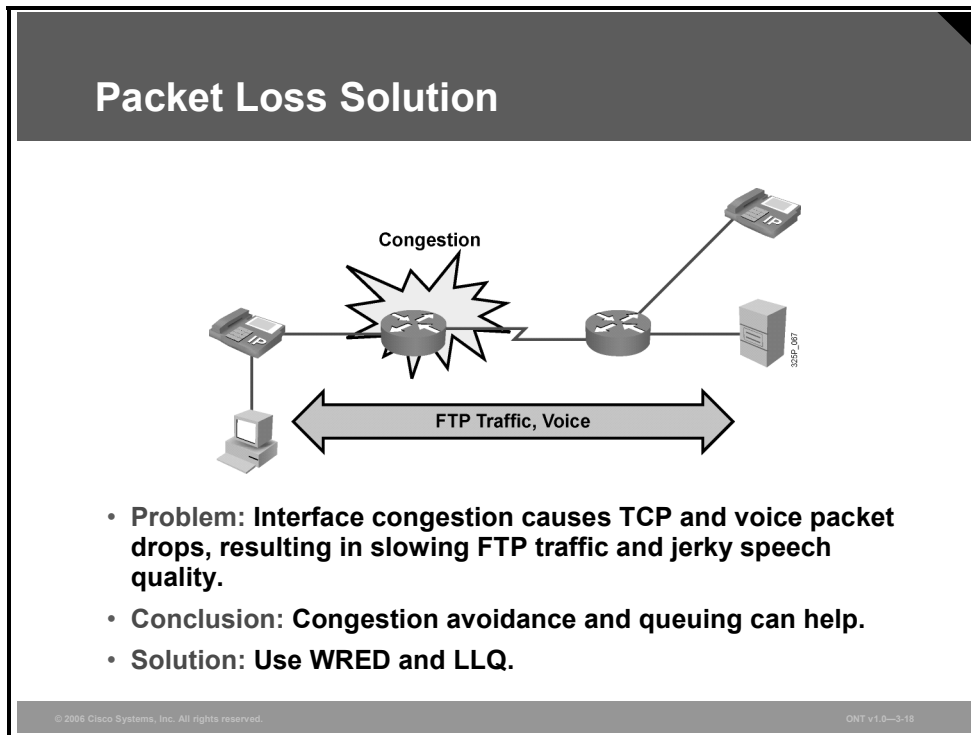


These approaches can be taken to prevent drops in sensitive applications:

- Increase link capacity to ease or prevent congestion.
- Guarantee enough bandwidth and increase buffer space to accommodate bursts of traffic from fragile flows. LLQ is the mechanism available in Cisco IOS QoS software that can both guarantee bandwidth and provide prioritized forwarding for drop-sensitive applications.
- Prevent congestion by dropping lower-priority packets before congestion occurs. Use weighted random early detection (WRED) to start dropping lower-priority packets before congestion occurs.

## Example: Packet Loss Solution

In this scenario, the customer connected via the WAN suffers from packet loss as a result of interface congestion. This behavior results in poor voice quality and slow data traffic. An upgrade of the WAN link is not considered an option. A number of actions must be taken to solve this problem and restore network quality.

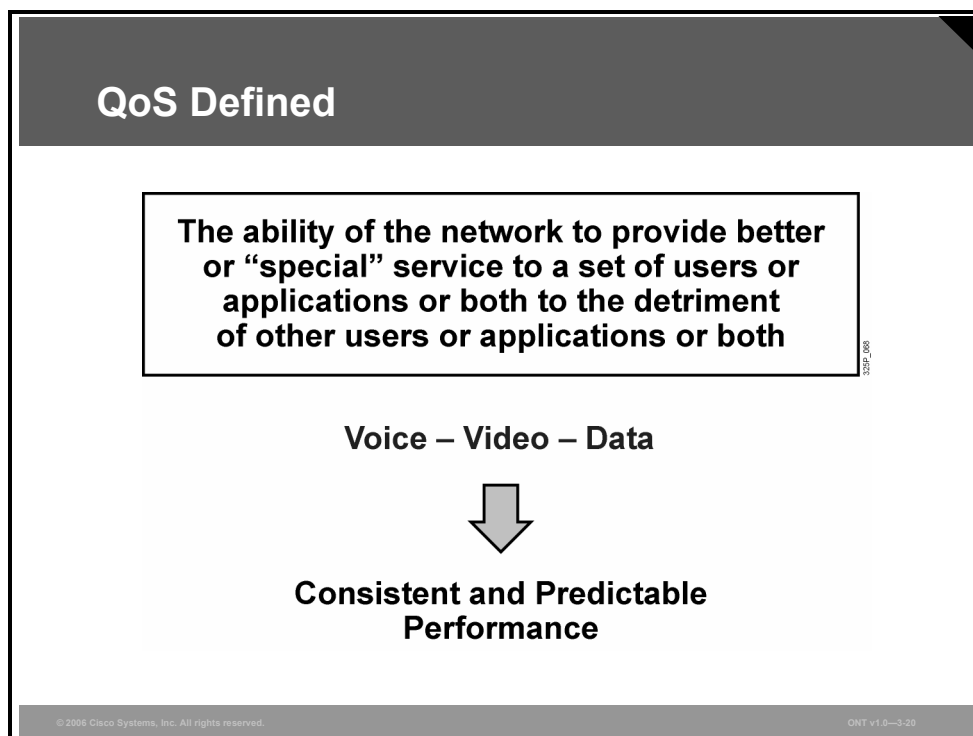


Congestion-avoidance techniques monitor network traffic loads in an effort to anticipate and avoid congestion at common network and internetwork bottlenecks before it becomes a problem. These techniques are designed to provide preferential treatment for premium (priority) traffic when there is congestion while concurrently maximizing network throughput and capacity utilization and minimizing packet loss and delay. WRED is one of the Cisco IOS QoS congestion-avoidance features.

The WRED algorithm provides for congestion avoidance on network interfaces by providing buffer management and allowing TCP traffic to throttle back before buffers are exhausted. The use of WRED helps avoid tail drops and global synchronization issues, maximizing network utilization and TCP-based application performance. There is no such congestion avoidance for User Datagram Protocol (UDP)-based traffic, such as voice traffic. In case of UDP-based traffic, methods such as queuing and compression techniques help to reduce and even prevent UDP packet loss. As the figure indicates, congestion avoidance combined with queuing can be a very powerful tool for avoiding packet drops.

# QoS Defined

This topic defines QoS with respect to traffic in a network.



In any bandwidth-limited network, QoS is used to reduce jitter, delay, and packet loss for time-sensitive and mission-critical applications.

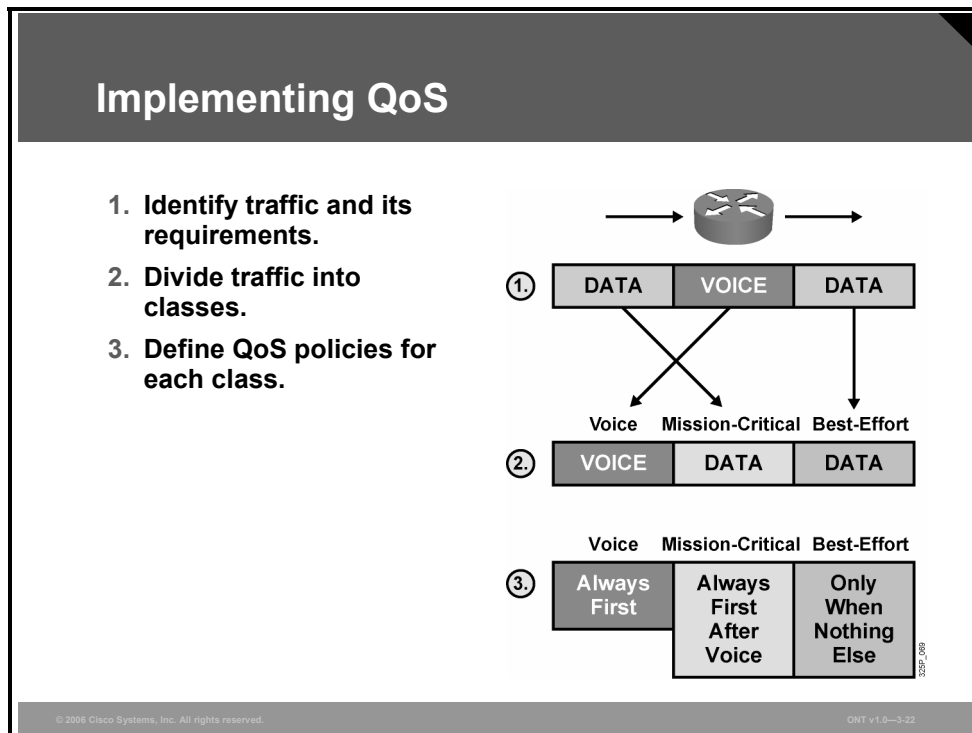
QoS is the ability of the network to provide better or “special” services to selected users and applications, to the detriment of other users and applications.

Cisco IOS QoS features enable network administrators 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 that, when correctly applied, help to provide consistent and predictable performance.

# Implementing QoS

This topic describes the three key steps involved in implementing a QoS policy on a network.

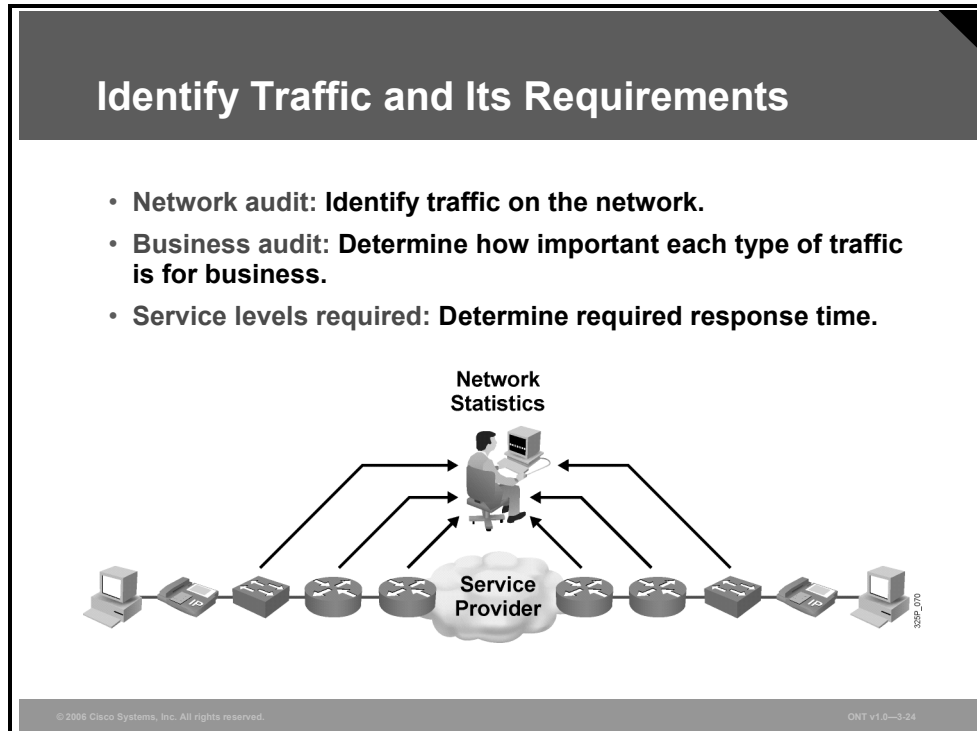


There are three basic steps involved in implementing QoS on a network:

- Step 1** Identify traffic and its requirements. Study the network to determine the type of traffic running on the network and then determine the QoS requirements for the different types of traffic.
- Step 2** Group the traffic into classes with similar QoS requirements. In the example, three classes of traffic might be defined: voice, mission-critical, and best-effort.
- Step 3** Define QoS policies that will meet the QoS requirements for each traffic class.

# QoS Traffic Classes—The Requirements of Different Traffic Types

This topic describes how traffic is recognized by type in a network and how those types resolve to QoS traffic classes.



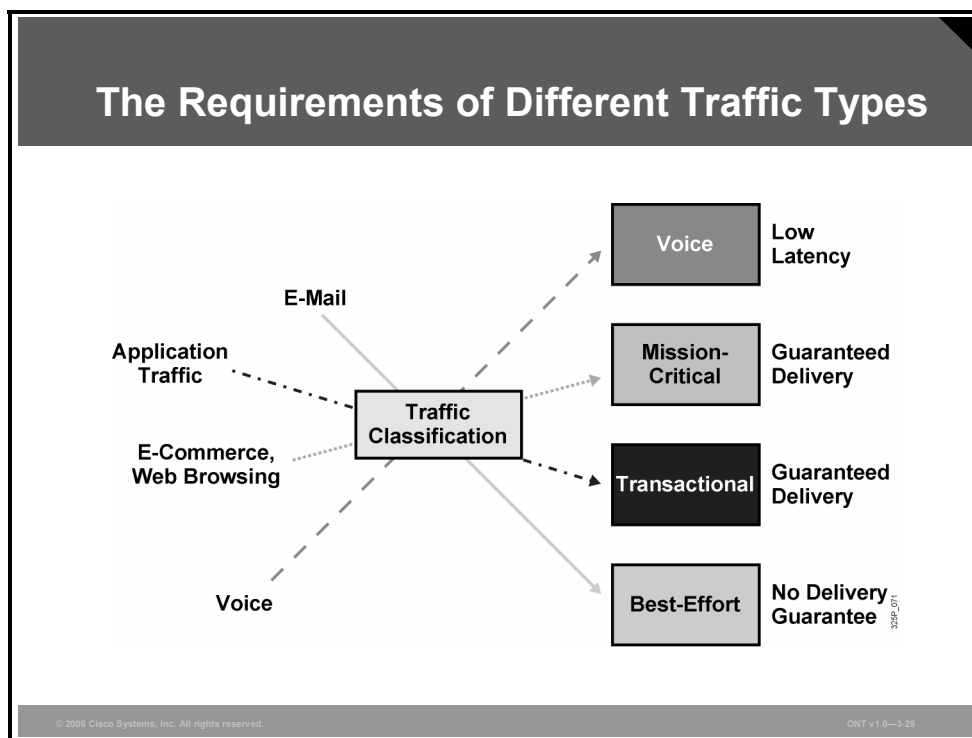
## Identify Traffic and Its Requirements

The first step in implementing QoS is identifying the traffic on the network and determining QoS requirements and the importance of the various traffic types. This step consists of these activities:

- Determine the QoS problems of users. Measure the traffic on the network during congested periods. Conduct CPU utilization assessment on each of the network devices during busy periods to determine where problems might be occurring.
- Determine the business model and goals and obtain a list of business requirements. This activity helps define the number of classes and allows you to determine the business requirements for each traffic class.
- Define the service levels required by different traffic classes in terms of response time and availability. What is the impact on business if a transaction is delayed by two or three seconds? Can file transfers wait until the network is underutilized?

# The Requirements of Different Traffic Types

After most network traffic has been identified and measured, use the business requirements to perform the second step: Define the traffic classes.



Because of its stringent QoS requirements, voice traffic is almost always in a class by itself. Cisco has developed specific QoS mechanisms, such as LLQ, that ensure that voice always receives priority treatment over all other traffic.

After the applications with the most critical requirements have been defined, the remaining traffic classes are defined using business requirements.

## Example: Traffic Classification

A typical enterprise might define five traffic classes:

- **Voice:** Absolute priority for VoIP traffic.
- **Mission-critical:** Small set of locally defined critical business applications.
- **Transactional:** Database access, transaction services, interactive traffic, and preferred data services.
- **Best effort:** E-mail.
- **Scavenger:** The unspecified traffic is considered as less than best effort. Scavenger applications, such as BitTorrent and other point-to-point applications, will be served by that class.



# QoS Policy

This topic describes how to define QoS policies after traffic classes have been defined.

## QoS Policy

- A networkwide definition of the specific levels of QoS assigned to different classes of network traffic

ABC Corporation

Network QoS Policy

Voice Traffic  
Absolute Priority

ERP System  
Critical Priority

Manufacturing System  
Critical Priority

Net Surfing  
Not Allowed During Business Hours

© 2006 Cisco Systems, Inc. All rights reserved. QNT v1.0—3-27

In the third step, define a QoS policy for each traffic class. Defining a QoS policy involves one or more of these activities:

- Setting a minimum bandwidth guarantee
- Setting a maximum bandwidth limit
- Assigning priorities to each class
- Using QoS technologies, such as advanced queuing, to manage congestion

## Example: Defining QoS Policies

Using the traffic classes previously defined QoS policies could be mandated based on the following priorities (with 5 being the highest and 1 the lowest):

- **Priority 5—Voice:** Minimum bandwidth of 1 Mbps. Use LLQ to always give voice priority.
- **Priority 4—Mission-critical:** Minimum bandwidth of 1 Mbps. Use CBWFQ to prioritize critical-class traffic flows.
- **Priority 3—Transactional:** Minimum bandwidth of 1 Mbps. Use CBWFQ to prioritize transactional traffic flows.
- **Priority 2—Best-effort:** Maximum bandwidth of 500 kbps. Use CBWFQ to prioritize best-effort traffic flows that are below mission-critical and voice.
- **Priority 1—Scavenger (less-than-best-effort):** Maximum bandwidth of 100 kbps. Use WRED to drop these packets whenever the network has a tendency toward congestion.

# Summary

This topic summarizes the key points that were discussed in this lesson.

## Summary

- **Converged networks that support voice, video, and data create new requirements for managing network traffic. QoS meets those requirements.**
- **Converged networks suffer from different quality issues, including lack of adequate bandwidth, end-to-end and variable delay, and lost packets.**
- **Packet loss can adversely affect QoS in a network.**
- **QoS is a way to improve the performance of converged networks.**

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—3-28

## Summary (Cont.)

- **Lack of recourses causes networks to experience different types of delay, including processing delay, queuing delay, serialization delay, and propagation delay.**
- **QoS traffic classes need to be defined to implement a QoS policy.**
- **Implementing QoS requires three steps: identify requirements, classify network traffic, and define networkwide policies for quality.**
- **A QoS policy is a networkwide definition of the specific levels of QoS assigned to classes of network traffic.**

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—3-29

# Identifying Models for Implementing QoS

---

## Overview

There are three models for implementing quality of service (QoS) in a network. The best-effort model is the simplest way of packet delivery. The Integrated Services (IntServ) model expects applications to signal their requirements to the network on demand. In the Differentiated Services (DiffServ) model, the network recognizes packets without signaling and provides the appropriate services to the packets. Modern IP networks can use all three models at the same time. This lesson briefly describes the three models.

## Objectives

Upon completing this lesson, you will be able to explain the use of the three models for providing QoS in a network. This ability includes being able to meet these objectives:

- List the models for providing QoS on a network
- Explain the key features of the best-effort model for QoS
- Explain the key features of the IntServ model for QoS
- Explain how RSVP enables the IntServ model to provide end-to-end QoS
- Explain the key features of the DiffServ model for QoS

# QoS Models

This topic describes the models for providing QoS on a network.

QoS Models	
Model	Characteristics
Best effort	No QoS is applied to packets.
IntServ	Applications signal to the network that they require certain QoS parameters.
DiffServ	The network recognizes classes that require QoS.

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0-3.3

There are three models for implementing quality of service (QoS) in a network:

- **Best-effort model:** With the best-effort model, QoS is not applied to packets. If it is not important when or how packets arrive, the best-effort model is appropriate.
- **Integrated Services (IntServ):** IntServ can provide very high QoS to IP packets. Essentially, applications signal to the network that they will require special QoS for a period of time and that bandwidth should be reserved. With IntServ, packet delivery is guaranteed. However, the use of IntServ can severely limit the scalability of a network.
- **Differentiated Services (DiffServ):** DiffServ provides the greatest scalability and flexibility in implementing QoS in a network. Network devices recognize traffic classes and provide different levels of QoS to different traffic classes.

# Best-Effort Model

This topic describes key features of the best-effort model for QoS.

## Best-Effort Model

- **Internet was initially based on a best-effort packet delivery service.**
- **Best-effort is the default mode for all traffic.**
- **There is no differentiation among types of traffic.**
- **Best-effort model is similar to using standard mail—It will get there when it gets there.**

© 2006 Cisco Systems, Inc. All rights reserved. OMT v1.0—3.5

The Internet was designed for best-effort, no-guarantee delivery of packets. This behavior is still predominant on the Internet today.

If QoS policies are not implemented, traffic is forwarded using the best-effort model. All network packets are treated exactly the same—an emergency voice message is treated exactly like a digital photograph attached to an e-mail. Without QoS implemented, the network cannot tell the difference and, as a result, cannot treat packets preferentially.

When you drop a letter in standard postal mail, you are using a best-effort model. Your letter will be treated exactly the same as every other letter. With the best-effort model, the letter may never arrive and, unless you have a separate notification arrangement with the letter recipient, you may never know that the letter did not arrive.

# Benefits and Drawbacks

The best-effort model has several benefits and also some drawbacks.

## Benefits and Drawbacks of the Best-Effort Model

- **Benefits:**
  - **Highly scalable**
  - **No special mechanisms required**
- **Drawbacks:**
  - **No service guarantees**
  - **No service differentiation**

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—3-6

The best-effort model has these significant benefits:

- The model has nearly unlimited scalability. The only way to reach scalability limits is to reach bandwidth limits, in which case all traffic is equally affected.
- You do not need to employ special QoS mechanisms to use the best-effort model. It is the easiest and quickest model to deploy.

The best-effort model also has some drawbacks:

- Nothing is guaranteed. Packets will arrive whenever they can, in any order possible, if they arrive at all.
- Packets are not given preferential treatment. Critical data is treated the same as casual e-mail.

# IntServ Model

This topic describes the key features of the IntServ model for QoS.

## IntServ Model

- **Introduction of IntServ model (RFC 1633) was driven by real-time applications, such as remote video and conferencing.**
- **IntServ end-to-end model ensures guaranteed delivery and predictable behavior of the network for applications.**
- **Resource Reservation Protocol (RSVP) is used as a signaling protocol.**
- **The requested QoS parameters are then linked to a packet stream.**
- **End-to-end streams are not established if the required QoS parameters are not available.**

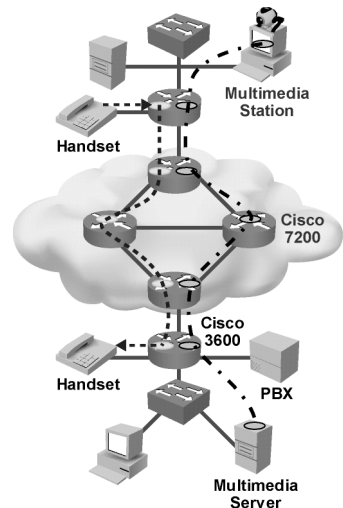
© 2006 Cisco Systems, Inc. All rights reserved. INT v1.0—3.8

Development of the IntServ architecture model (RFC 1633, June 1994) was motivated by the needs of real-time applications, such as remote video, multimedia conferencing, visualization, and virtual reality. The model provides a way to deliver the end-to-end QoS that real-time applications require by explicitly managing network resources to provide QoS to specific user packet streams (flows). The IntServ model uses resource reservation and admission-control mechanisms as key building blocks to establish and maintain QoS. This practice is similar to a concept known as “hard QoS.” With hard QoS, traffic characteristics, such as bandwidth, delay, and packet-loss rates, are guaranteed end to end. This guarantee ensures both predictable and guaranteed service levels for mission-critical applications.

IntServ uses Resource Reservation Protocol (RSVP) to explicitly signal the QoS needs of traffic of an application along the devices in the end-to-end path through the network. If network devices along the path can reserve the necessary bandwidth, the originating application can begin transmitting. If the requested reservation fails along the path, the originating application will not send any data.

## IntServ Model (Cont.)

- Provides multiple service levels
- Requests specific kind of service from the network before sending data
- Uses RSVP to reserve resources for specified QoS parameters
- Intelligent queuing mechanisms required to provide resource reservation in terms of:
  - Guaranteed rate
  - Controlled load (low delay, high throughput)



© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0-3.0

IntServ is a multiple-service model that can accommodate multiple QoS requirements. IntServ inherits the connection-oriented approach from telephony network design. Every individual communication must explicitly specify its traffic descriptor and requested resources to the network. The edge router performs admission control to ensure that available resources are sufficient in the network. The IntServ standard assumes that routers along a path set and maintain the state for each individual communication.

RSVP is used in the Cisco QoS architecture as one of several methods for providing Call Admission Control (CAC) for voice in a VoIP network. The RSVP method for CAC is the only method that makes an actual bandwidth reservation for each allowed voice call. Other CAC methods can only make a “best guess look-ahead” decision based on the state of the network at the initiation of the call. The use of RSVP not only provides CAC, it also guarantees QoS for the duration of the call regardless of changing network conditions. RSVP is the method used by Cisco Unified CallManager 5.0 to perform CAC.

If resources are available, RSVP accepts a reservation and installs a traffic classifier to assign a temporary QoS class for that traffic flow 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.

In the IntServ model, the application requests a specific kind of service from the network before sending data. The application informs the network of its traffic profile and requests a particular kind of service that can encompass its bandwidth and delay requirements. The application is expected to send data only *after* it gets a confirmation from the network. The application 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. The network commits to meeting the QoS requirements of the application as long as the traffic remains within the profile specifications. The network fulfills its commitment by maintaining the per-flow state and then performing packet classification, policing, and intelligent queuing based on that state.



The QoS feature set in Cisco IOS software includes these features that provide controlled traffic volume services:

- RSVP, which can be used by applications to signal their QoS requirements to the router
- Intelligent queuing mechanisms, which can be used with RSVP to provide these QoS service levels:
  - **Guaranteed-rate:** Allows applications to reserve bandwidth to meet their requirements. For example, a VoIP application can reserve 32 Mbps end to end using this type of service. Cisco IOS QoS uses low latency queuing (LLQ) with RSVP to provide a guaranteed-rate type of service.
  - **Controlled-load:** Allows applications to have low delay and high throughput, even during times of congestion. For example, adaptive real-time applications, such as the playback of a recorded conference, can use this service. Cisco IOS QoS uses RSVP with weighted random early detection (WRED) to provide a controlled-load type of service.

# IntServ Functions

Besides end-to-end signaling, IntServ requires several functions on routers and switches along the path.

## IntServ Functions

**IntServ requires several functions on routers and switches along the path:**

- **Admission control**
- **Classification**
- **Policing**
- **Queuing**
- **Scheduling**

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—5.10

These functions include the following:

- **Admission control:** Determines whether a new flow requested by users or systems can be granted the requested QoS without affecting existing reservations in order to guarantee end-to-end QoS.
- **Classification:** Entails using a traffic descriptor to categorize a packet within a specific group to define that packet and make it accessible for QoS handling on the network. Classification is pivotal to policy techniques that select packets traversing a network element or a particular interface for different types of QoS service.
- **Policing:** Takes action, including possibly dropping packets, when traffic does not conform to its specified characteristics. Policing is defined by rate and burst parameters, as well as by actions for in-profile and out-of-profile traffic.
- **Queuing:** Is designed to accommodate temporary congestion on an interface of a network device by storing excess packets in buffers until access to the bandwidth becomes available.
- **Scheduling:** A QoS component, the QoS scheduler, negotiates simultaneous requests for network access and determines which queue receives priority. Generally, queues are scheduled in a round-robin fashion.

# Benefits and Drawbacks

IntServ model has several benefits and also some drawbacks.

## Benefits and Drawbacks of the IntServ Model

- **Benefits:**
  - **Explicit resource admission control (end to end)**
  - **Per-request policy admission control (authorization object, policy object)**
  - **Signaling of dynamic port numbers (for example, H.323)**
- **Drawbacks:**
  - **Continuous signaling because of stateful architecture**
  - **Flow-based approach not scalable to large implementations, such as the public Internet (can be made more scalable when combined with elements of the DiffServ model)**

© 2006 Cisco Systems, Inc. All rights reserved. INT v1.0—3-11

IntServ has these key benefits:

- IntServ supports admission control that allows a network to reject or downgrade new RSVP sessions if one of the interfaces in the path has reached the limit (that is, all reservable bandwidth is booked).
- RSVP signals QoS requests per individual flow. In the request, the authorized user (authorization object) and needed traffic policy (policy object) are sent. The network can then provide guarantees to these individual flows.
- RSVP informs network devices of flow parameters (IP addresses and port numbers). Some applications use dynamic port numbers, such as H.323-based applications, which can be difficult for network devices to recognize. Network-Based Application Recognition (NBAR) is a mechanism that has been introduced to supplement RSVP for applications that use dynamic port numbers but do not use RSVP.

IntServ also has these drawbacks:

- There is continuous signaling because of the stateful RSVP architecture.
- The flow-based approach is not scalable to large implementations, such as the public Internet, because RSVP has to track each individual flow. This circumstance would make end-to-end signaling very difficult. A possible solution is to combine IntServ with elements from the DiffServ model to provide the needed scalability.

# RSVP and the IntServ QoS Model

This topic describes how RSVP enables the IntServ model to provide end-to-end QoS.

## Resource Reservation Protocol

- **Is carried in IP—protocol ID 46**
- **Can use both TCP and UDP port 3455**
- **Is a signaling protocol and works in conjunction with existing routing protocols**
- **Requests QoS parameters from all devices that are between the source and the destination**
- **Is intended to provide divergent performance requirements for multimedia applications:**
  - **Rate-sensitive traffic**
  - **Delay-sensitive traffic**

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—3-13

## Resource Reservation Protocol

RSVP is a network control protocol that enables applications to obtain differing QoS for their data flows. Such a capability recognizes that different applications have different network performance requirements. Some applications, including the more traditional interactive and batch applications, require reliable delivery of data but do not impose any stringent requirements for the timeliness of delivery. Newer application types, including videoconferencing, IP telephony, and other forms of multimedia communications, require almost the exact opposite: Data delivery must be timely but not necessarily reliable. Thus, RSVP was intended to provide IP networks with the ability to support the divergent performance requirements of differing application types.

RSVP is an IP protocol that uses IP protocol ID 46 and TCP and UDP port 3455.

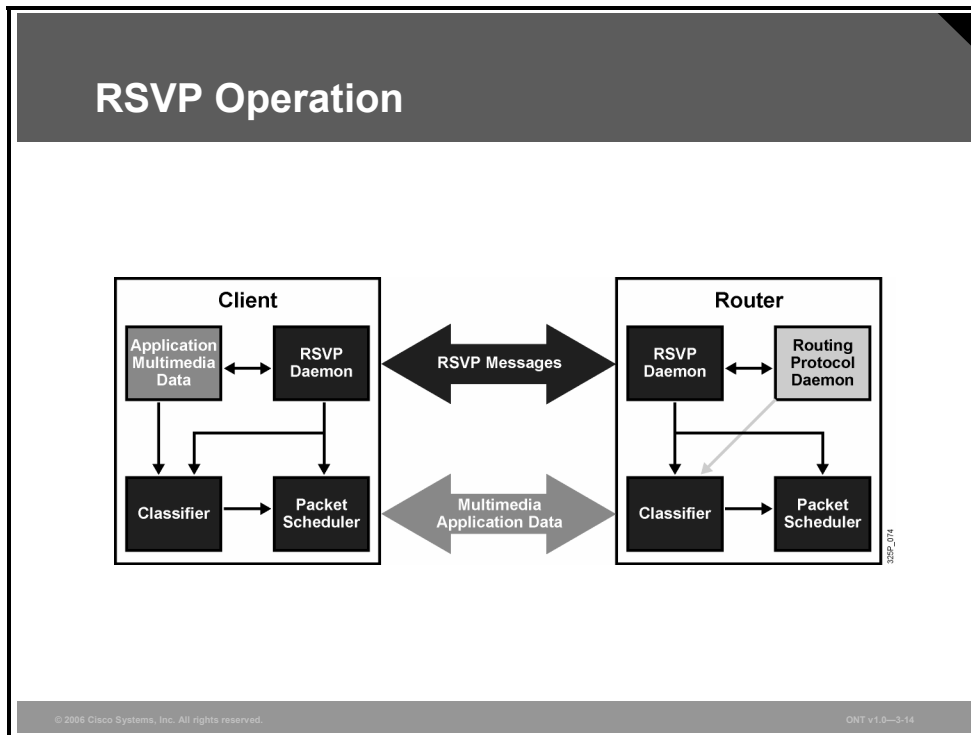
It is important to note that RSVP is not a routing protocol. RSVP works in conjunction with routing protocols and installs the equivalent of dynamic access control lists (ACLs) along the routes that routing protocols calculate. Thus, implementing RSVP in an existing network does not require migration to a new routing protocol.

In RSVP, a data flow is a sequence of datagrams that have the same source, destination (regardless of whether that destination is one or more physical machines), and QoS requirements. QoS requirements are communicated through a network via a flow specification, which is a data structure used by internetwork hosts to request special services from the internetwork. A flow specification describes the level of service required for that data flow. RSVP focuses on two main traffic types:

- **Rate-sensitive traffic:** Traffic that requires a guaranteed and a constant (or nearly constant) transmission rate from its source to its destination. An example of such an application is H.323 videoconferencing. RSVP enables constant-bit-rate service in packet-switched networks via its rate-sensitive level of service. This service is sometimes referred to as guaranteed-bit-rate service.
- **Delay-sensitive traffic:** Traffic that requires timeliness of delivery and that varies its rate accordingly. MPEG-II video, for example, averages about 3 to 7 Mbps, depending on the amount of changes in the picture. RSVP services supporting delay-sensitive traffic are referred to as controlled-delay service (non-real-time service) and predictive service (real-time service).

## RSVP Operation

Under RSVP, resources are reserved for simple data streams (unidirectional data flows). Each sender is logically distinct from a receiver, but any application can act as a sender and a receiver. Receivers are responsible for requesting resource reservations. The figure illustrates this general operational environment, and the text provides an outline of the specific sequence of events.



The RSVP resource-reservation process initiation begins when an RSVP daemon consults the local routing protocols to obtain routes. A host sends Internet Group Management Protocol (IGMP) messages to join a multicast group to participate in a videoconference, for example, and RSVP messages to reserve resources along the delivery paths for that group. Each router that is capable of participating in resource reservation passes incoming data packets to a packet classifier and then queues them as necessary in a packet scheduler. The RSVP packet classifier determines the route and QoS class for each packet. The RSVP scheduler allocates resources for transmission on the particular data link layer medium used by each interface. If the data link layer medium has its own QoS management capability, the packet scheduler is responsible for negotiation with the data link layer to obtain the QoS requested by RSVP.

The scheduler itself allocates packet-transmission capacity on a QoS-passive medium, such as a leased line, and can also allocate other system resources, such as CPU time or buffers. A QoS request, typically originating in a receiver host application, is passed to the local RSVP implementation as an RSVP daemon.

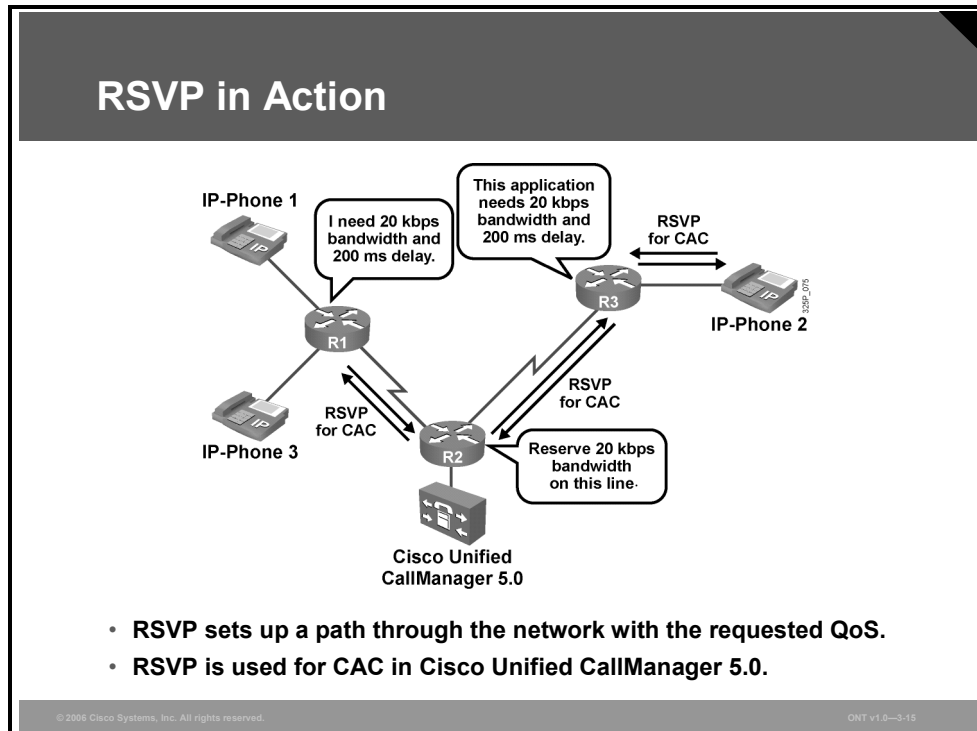
The RSVP protocol is then used to pass the request to all the nodes (routers and hosts) along the reverse data paths to the data sources. At each node, the RSVP program applies a local decision procedure called admission control to determine whether it can supply the requested QoS. If admission control succeeds, the RSVP program sets the parameters of the packet classifier and scheduler to obtain the desired QoS. If admission control fails at any node, the RSVP program returns an error indication to the application that originated the request.

It is impossible to deploy RSVP or any new protocol at the same moment throughout the entire Internet. Indeed, RSVP might never be deployed everywhere. To support connection of RSVP networks through non-RSVP networks, RSVP supports tunneling, which occurs automatically through non-RSVP clouds.

## Example: RSVP in Action

To better understand the basic principles of how RSVP performs Call Admission Control (CAC) and bandwidth reservation in a network from a functionality perspective, consider this example in which RSVP is enabled on each router interface in the network.

In this scenario, three Cisco IP phones and Cisco Unified CallManager 5.0 are connected with each other over an IntServ-enabled WAN. Because bandwidth is limited on the WAN links, RSVP will determine whether the requested bandwidth for a successful call is available. For performing CAC, Cisco Unified CallManager 5.0 uses RSVP.



An RSVP-enabled voice application wants to reserve 20 kbps of bandwidth for a data stream from IP-Phone 1 to IP-Phone 2.

Recall that RSVP does not perform its own routing; instead, it uses underlying routing protocols to determine where it should carry reservation requests. As routing changes paths to adapt to topology changes, RSVP adapts its reservation to the new paths wherever reservations are in place.

The RSVP protocol attempts to establish an end-to-end reservation by checking for available bandwidth resources on all RSVP-enabled routers along the path from IP-Phone 1 to IP-Phone 2. As the RSVP messages progress through the network from R1 via R2 to R3, the available RSVP bandwidth is decremented by 20 kbps on the router interfaces. For voice calls, a reservation must be made in both directions.

The available bandwidth on all interfaces is sufficient to accept the new data stream, so the reservation succeeds and the application is notified.

# DiffServ Model

This topic describes the key features of DiffServ for QoS.

## DiffServ Model

- **DiffServ (RFC 2474 and RFC 2475) was designed to overcome the limitations of both the best-effort and IntServ models.**
- **Network traffic is identified by classes.**
- **Network QoS policy enforces differentiated treatment of traffic classes.**
- **You choose level of service for each traffic class.**

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—3-17

DiffServ was designed to overcome the limitations of both the best-effort and IntServ models. The DiffServ model is described in Internet Engineering Task Force (IETF) RFC 2474 and RFC 2475. DiffServ can provide an “almost guaranteed” QoS while still being cost-effective and scalable.

The DiffServ model is similar to a concept known as “soft QoS.” With soft QoS, QoS mechanisms are used without prior signaling. In addition, QoS characteristics (for example, bandwidth and delay), are managed on a hop-by-hop basis by policies that are established independently at each intermediate device in the network. This action is also known as per-hop behavior (PHB). The soft QoS approach is not considered an end-to-end QoS strategy because end-to-end guarantees cannot be enforced. However, soft QoS is a more scalable approach to implementing QoS than hard QoS (the IntServ model), because many (hundreds or potentially thousands) of applications can be mapped into a small set of classes upon which similar sets of QoS behaviors are implemented. Although QoS mechanisms in this approach are enforced and applied on a hop-by-hop basis, uniformly applying global meaning to each traffic class provides both flexibility and scalability.

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

DiffServ works like a packet delivery service. You request (and pay for) a level of service when you send your package. Throughout the package network, the level of service is recognized and your package is given either preferential or normal service, depending on what you requested.



# Benefits and Drawbacks

The DiffServ model has several benefits and also some drawbacks.

## Benefits and Drawbacks of the DiffServ Model

- **Benefits:**
  - **Highly scalable**
  - **Many levels of quality possible**
- **Drawbacks:**
  - **No absolute service guarantee**
  - **Complex mechanisms**

© 2006 Cisco Systems, Inc. All rights reserved. OMT v1.0—3-18

Key benefits of DiffServ include:

- It is highly scalable.
- It provides many different levels of quality.

DiffServ also has these drawbacks:

- No absolute guarantee of service quality can be made.
- It requires a set of complex mechanisms to work in concert throughout the network.

# Summary

This topic summarizes the key points that were discussed in this lesson.

## Summary

- There are three models for providing QoS: best effort, IntServ, and DiffServ.
- Although the best-effort model is highly scalable, it has no provision for differentiating among types of network traffic and, as a result, does not provide QoS.
- The IntServ model offers absolute QoS guarantees by explicitly reserving bandwidth by using RSVP. Scalability is achieved in conjunction with elements of the DiffServ model.
- RSVP is not a routing protocol; thus, implementing RSVP in an existing network does not require migration to a new routing protocol.
- The DiffServ model provides the ability to classify network traffic and offer many levels of QoS while being highly scalable.

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—3-19

# Identifying Methods for Implementing QoS

---

## Overview

Traditionally, the Cisco IOS command-line interface (CLI) has been used to implement quality of service (QoS) in a network. For converged networks, Cisco recommends using either the Modular QoS CLI (MQC) or Cisco AutoQoS. The MQC offers a highly modular way to fine-tune a network. Cisco AutoQoS offers automated methods for quickly incorporating consistent voice QoS in a converged network of routers and switches. Network administrators and architects can also benefit from using the Cisco Router and Security Device Manager (SDM) QoS wizard. The Cisco SDM QoS wizard provides centralized QoS design, administration, and traffic monitoring that scales to large QoS deployments. This lesson explores in detail the four methods for implementing and managing QoS.

## Objectives

Upon completing this lesson, you will be able to explain how to implement QoS policies using both MQC and the Cisco SDM QoS wizard. This ability includes being able to meet these objectives:

- List and describe methods for configuring and monitoring QoS on a network
- Explain, at a high level, the CLI (nonmodularized) method of configuring QoS
- Explain, at a high level, the MQC method of configuring QoS
- Explain, at a high level, the Cisco AutoQoS methods of configuring QoS
- Explain the Cisco SDM QoS wizard, including how to access it and the high-level configuration that it can do
- Identify and explain the advantages and disadvantages of using each of the methods of implementing QoS on a network

# Methods for Implementing QoS Policy

This topic describes four methods for implementing and managing a QoS policy.

Methods for Implementing QoS Policy	
Method	Description
Legacy CLI	<ul style="list-style-type: none"><li>• CLI</li><li>• Configures QoS on interface level</li><li>• Time-consuming</li></ul>
MQC	<ul style="list-style-type: none"><li>• CLI</li><li>• Makes configurations modular</li><li>• Best way for QoS fine tuning</li></ul>
Cisco AutoQoS	<ul style="list-style-type: none"><li>• Applies a possible QoS configuration to the interfaces</li><li>• Fastest way to implement QoS</li></ul>
Cisco SDM QoS wizard	<ul style="list-style-type: none"><li>• Application for simple QoS configurations</li></ul>

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0-3.3

A few years ago, the only way to implement quality of service (QoS) in a network was by using the command-line interface (CLI) to individually configure QoS policies at each interface. This was a time-consuming and error-prone task that involved cutting and pasting configurations from one interface to another.

Cisco introduced the Modular QoS CLI (MQC) to simplify QoS configuration by making configurations modular. With MQC, QoS can be configured in a building-block approach using a single module repeatedly to apply policy to multiple interfaces.

Cisco AutoQoS represents innovative technology that simplifies the challenges of network administration by reducing QoS complexity, deployment time, and cost to enterprise networks. Cisco AutoQoS incorporates value-added intelligence in Cisco IOS software and Cisco Catalyst software to provision and assist in the management of large-scale QoS deployments.

The first phase of Cisco AutoQoS VoIP offers straightforward capabilities to automate VoIP deployments for customers that want to deploy IP telephony but lack the expertise and staffing to plan and deploy IP QoS and IP services. The second phase, Cisco AutoQoS Enterprise, which is supported only on router interfaces, uses Network-Based Application Recognition (NBAR) to discover the traffic. After this discovery phase, the AutoQoS process can then configure the interface to support up to 10 traffic classes.

Customers can easily configure and manage and successfully troubleshoot QoS deployments by using Cisco Router and Security Device Manager (SDM) QoS wizard. The Cisco SDM QoS wizard provides centralized QoS design, administration, and traffic monitoring that scales to large QoS deployments.

# Legacy CLI

This topic describes the CLI method for implementing QoS.

## Legacy CLI

- **Uses the CLI via console and Telnet**
- **Traditional method**
- **Nonmodular**
- **Cannot separate traffic classification from policy definitions**
- **Time-consuming and potentially error-prone task**
- **Used to augment, fine-tune newer Cisco AutoQoS method**

© 2006 Cisco Systems, Inc. All rights reserved. OMT v1.0—3-5

The CLI can be accessed via the console or Telnet. The CLI method offers less granularity of supported QoS features than other QoS configuration techniques. The CLI offers basic QoS functionalities with limited options; for example, the traffic classification cannot be fully separated from the QoS mechanisms.

---

**Note**      The CLI is not recommended for implementing QoS policy.

---

# Legacy CLI Usage Guidelines

This section describes the legacy CLI usage guidelines.

## Legacy CLI Usage Guidelines

- **Build a traffic policy:**
  - **Identify the traffic pattern.**
  - **Classify the traffic.**
  - **Prioritize the traffic.**
  - **Select a proper QoS mechanism:**
    - **Queuing**
    - **Compression**
- **Apply the traffic policy to the interface.**

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—3-4

The general guidelines for using the legacy CLI method are these:

- **Build a QoS policy (traffic policy):**
  - Identify the traffic patterns in your network by using a packet analyzer. This activity gives you the ability to identify the traffic types, for example, IP, TCP, User Datagram Protocol (UDP), DECnet, AppleTalk, and Internetwork Packet Exchange (IPX).
  - After you have performed the traffic identification, start classifying the traffic. For example, separate the voice traffic class from the business-critical traffic class.
  - For each traffic class, specify the priority for the class. For example, voice will be assigned a higher priority than business-critical traffic.
  - After applying the priorities to the traffic classes, select a proper QoS mechanism, such as queuing, compression, or a combination of both. This action determines which traffic will leave the device first and how.
- **Apply the QoS policy to the device interface.**

---

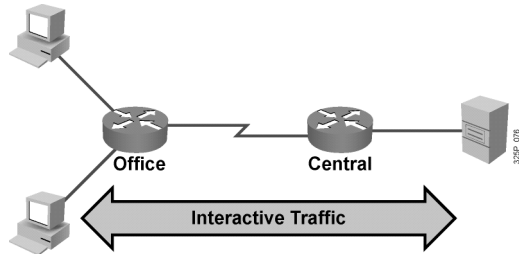
**Note** Using the legacy CLI method allows basic QoS features.

---

## Legacy CLI Example

In this example, a possible implementation scenario for legacy CLI is shown, followed by a sample configuration.

### Legacy CLI Example



**For interactive traffic, CQ and TCP header compression can be used.**

```
interface multilink
ip address 10.1.61.1 255.255.255.0
load-interval 30
custom-queue-list 1
ppp multilink
ppp multilink fragment-delay 10
ppp multilink interleave
multilink-group 1
ip tcp header-compression iphc-format
!
queue-list 1 protocol ip 2 tcp 23
```

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—3-7

In this scenario, the office site is connected over a low-speed WAN link to the central site. Both sites are equipped with PCs and servers that run interactive applications, such as terminal services. Because the available bandwidth is limited, an appropriate strategy for efficient bandwidth usage must be devised.

In a network with remote sites, where interactive traffic is used for daily business, bandwidth availability is an issue.

The available bandwidth resources need to be efficiently used. Because only simple services are run, basic queuing techniques, such as priority queuing (PQ) or custom queuing (CQ), and header compression mechanisms, such as TCP header compression, are needed to use the bandwidth much more efficiently.

---

**Note** PQ and CQ are traditional Cisco priority mechanisms that have been mostly replaced by more advanced mechanisms, such as weighted fair queuing (WFQ), class-based weighted fair queuing (CBWFQ), and low latency queuing (LLQ).

---

Depending on the kind of traffic that uses in the network, suitable queuing and compression mechanisms need to be chosen. In this example, CQ and TCP header compression are a strategy for interactive traffic quality assurance.

The printout represents an example of complex configuration tasks involved in using the CLI.

For each QoS feature, a separate line is needed. Two lines are needed for CQ: one line that sets up the queue list, in this example, for Telnet traffic, and a second line that binds the queue list to an interface and activates it.

Four lines are needed for PPP multilink configuration and another line for TCP header compression.

# Modular QoS CLI

This topic describes the MQC method for implementing QoS.

## Modular QoS CLI

- **A command syntax for configuring QoS policy**
- **Reduces configuration steps and time**
- **Configures policy, not “raw” per-interface commands**
- **Uniform CLI across major Cisco IOS platforms**
- **Uniform CLI structure for all QoS features**
- **Separates classification engine from the policy**

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—3.9

The MQC allows users to create traffic policies and then attach these policies to interfaces. A QoS policy contains one or more traffic classes and one or more QoS features. A traffic class is used to classify traffic, and the QoS features in the QoS policy determine how to treat the classified traffic.

The MQC offers significant advantages over the legacy CLI method for implementing QoS. By using MQC, a network administrator can significantly reduce the time and effort it takes to configure QoS in a complex network. Rather than configuring “raw” CLI commands interface by interface, the administrator develops a uniform set of traffic classes and QoS policies that can be applied on interfaces.

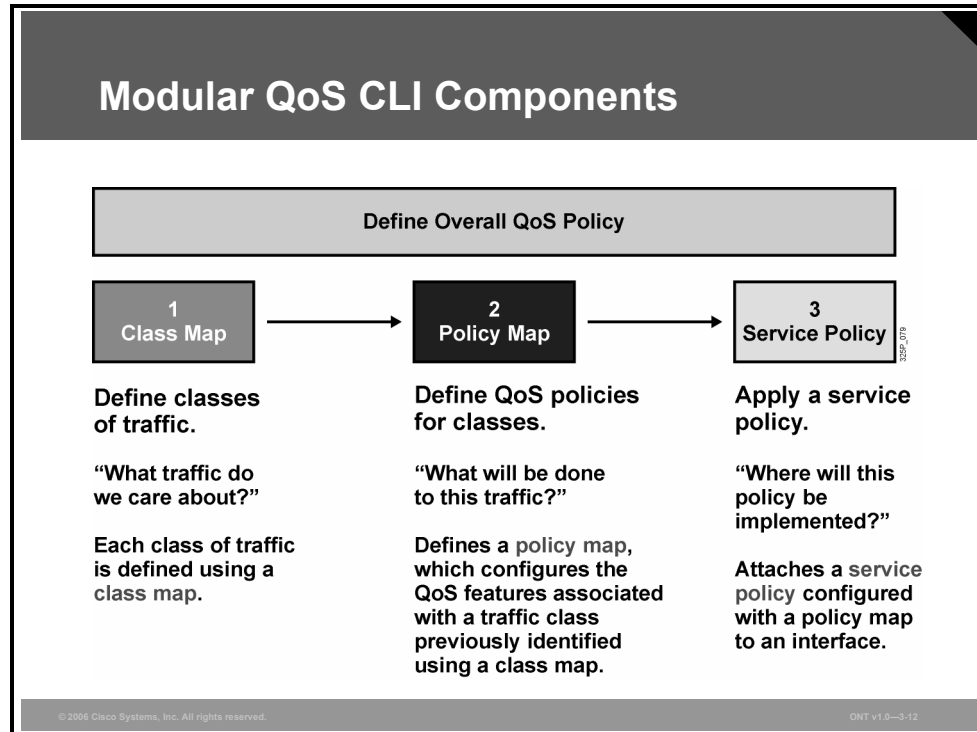
The use of the MQC allows the separation of traffic classification from the definition of QoS policy. This capability enables easier initial QoS implementation and maintenance as new traffic classes emerge and QoS policies for the network evolve.



# Modular QoS CLI Components

To define a QoS policy, the following questions need to be answered:

- What traffic patterns will QoS be used for?
- What will be done with this traffic after it is classified?
- Where will this policy be applied?



Complete these steps to implement QoS using the MQC:

- Step 1** Configure traffic classification by using the **class-map** command.
- Step 2** Configure traffic policy by associating the traffic class with one or more QoS features using the **policy-map** command.
- Step 3** Attach the traffic policy to inbound or outbound traffic on interfaces, subinterfaces, or virtual circuits by using the **service-policy** command.

# Class Maps

Class maps are used to create classification templates that are later used in policy maps where QoS mechanisms are bound to classes.

## Class Maps

- “What traffic do we care about?”
- Each class is identified using a class map.
- A traffic class contains three major elements:
  - A case-sensitive name
  - A series of match commands
  - An instruction on how to evaluate the match commands if more than one match command exists in the traffic class
- Class maps can operate in two modes:
  - Match all: All conditions have to succeed.
  - Match any: At least one condition must succeed.
- The default mode is match all.

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—3-13

Routers can be configured with a large number of class maps (currently limited to 256).

A class map is created using the **class-map** global configuration command. Class maps are identified by case-sensitive names. Each class map contains one or more conditions that determine whether the packet belongs to the class.

There are two ways of processing conditions when there is more than one condition in a class map:

- **Match all:** All conditions have to be met to bind a packet to the class.
- **Match any:** At least one condition has to be met to bind the packet to the class.

The default match strategy of class maps is match all.

# Configuring Class Maps

Use the **class-map** global configuration command to create a class map and enter class-map configuration mode. A class map is identified by a case-sensitive name; therefore, all subsequent references to the class map must use exactly the same name.

## Configuring Class Maps

```
router(config)#  
class-map [match-all | match-any] class-map-name
```

- Enters class-map configuration mode.
- Specifies the matching strategy.

```
router(config-cmap)#  
match any  
match not match-criteria
```

- Use at least one condition to match packets.

```
router(config-cmap)#  
description description
```

- You should use descriptions in large and complex configurations.
- The description has no operational meaning.

© 2006 Cisco Systems, Inc. All rights reserved. OMT v1.0—3-14

The **match** commands are used to specify various criteria for classifying packets. Packets are checked to determine whether they match the criteria specified in the match commands. If a packet matches the specified criteria, that packet is considered a member of the class and is forwarded according to the QoS specifications set in the traffic policy. Packets that fail to meet any of the matching criteria are classified as members of the default traffic class. The MQC does not necessarily require that users associate a single traffic class to one traffic policy. Multiple types of traffic can be associated with a single traffic class using the **match any** command.

The **match not** command inverts the condition specified. This command 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.

---

**Note** At least one **match** command should be used within the class-map configuration mode.

---

The **description** command is used for documenting a comment about the class map.

## ACLs for Traffic Classification

There are many ways to classify traffic when configuring class maps. One possible way is the use of access control lists (ACLs) to specify the traffic that needs to match for the QoS policy. Class maps support standard ACLs and extended ACLs.

### ACLs for Traffic Classification

```
router(config)#  
access-list access-list-number {permit | deny | remark}  
source [mask]
```

- **Standard ACL**

```
router(config)#  
access-list access-list-number {permit | deny} protocol  
source source-wildcard [operator port] destination  
destination-wildcard [operator port] [established] [log]
```

- **Extended ACL**

```
router(config-cmap)#  
match access-group access-list-number
```

- **Uses an ACL as a match criterion.**

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—3-15

The **match access-group** command will allow an ACL to be used as a match criterion for traffic classification.

# Policy Maps

The **policy-map** command is used to create a traffic policy. The purpose of a traffic policy is to configure the QoS features that should be associated with the traffic that has been classified in a traffic class or classes. A traffic policy contains three elements: a case-sensitive name, a traffic class (specified with the **class** command), and the QoS policies.

## Policy Maps

- “What will be done to this traffic?”
- Defines a traffic policy, which configures the QoS features associated with a traffic class previously identified using a class map.
- A traffic policy contains three major elements:
  - A case-sensitive name
  - A traffic class
  - The QoS policy associated with that traffic class
- Up to 256 traffic classes can be associated with a single traffic policy.
- Multiple policy maps can be nested to influence the sequence of QoS actions.

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—3-16

The name of a traffic policy is specified in the **policy-map** command (for example, issuing the **policy-map class1** command would create a traffic policy named class1). After you issue the **policy-map** command, you enter policy-map configuration mode. You can then enter the name of a traffic class. Here is where you enter QoS features to apply to the traffic that matches this class.

The MQC does not necessarily require that you associate only one traffic class to a single traffic policy. When packets match to more than one match criterion, multiple traffic classes can be associated with a single traffic policy.

A packet can match only one traffic class within a traffic policy. If a packet matches more than one traffic class in the traffic policy, the first traffic class defined in the policy is used.

# Configuring Policy Maps

Service policies are configured using the **policy-map** command. Up to 256 classes can be used within one policy map using the **class** command with the name of a preconfigured class map.

## Configuring Policy Maps

```
router(config)#  
policy-map policy-map-name
```

- Enters policy-map configuration mode.
- Policy maps are identified by a case-sensitive name.

```
router(config-pmap)#  
class {class-name | class-default}
```

- Enters the per-class policy configuration mode by using the name of a previously configured class map.
- Use the class-default name to configure the policy for the default class.

```
router(config-pmap)#  
class class-name condition
```

- Optionally, you can define a new class map by entering the condition after the name of the new class map.
- Uses the match any strategy.

© 2006 Cisco Systems, Inc. All rights reserved.ONT v1.0—3-17

A nonexistent class can also be used within the policy-map configuration mode if the match condition is specified after the name of the class. The running configuration will reflect such a configuration by using the match-any strategy and inserting a full class map configuration.

The table shows starting and resulting configuration modes for the **class-map**, **policy-map**, and **class** commands.

## Configuration Modes

Starting Configuration Mode	Command	Configuration Mode
Router(config)#	<b>class-map</b>	Router(config-cmap)#
Router(config)#	<b>policy-map</b>	Router(config-pmap)#
Router(config-pmap)#	<b>class</b>	Router(config-pmap-c)#

All traffic that is not classified by any of the class maps that are used within the policy map is part of the default class “class-default.” This class has no QoS guarantees, by default. The default class, when used on output, can use one FIFO queue or flow-based WFQ. The default class is part of every policy map, even if it is not configured.

# Service Policy

The last configuration step when configuring QoS mechanisms using the MQC is to attach a policy map to the inbound or outbound packets using the **service-policy** command.

## Service Policy

- “Where will this policy be implemented?”
- **Attaches a traffic policy configured with a policy map to an interface.**
- **Service policies can be applied to an interface for inbound or outbound packets.**

© 2006 Cisco Systems, Inc. All rights reserved. OMT v1.0--3.18

Using the **service-policy** command, you can assign a single policy map to multiple interfaces or assign multiple policy maps to a single interface (a maximum of one in each direction, inbound and outbound). A service policy can be applied for inbound or outbound packets.

# Attaching Service Policies to Interfaces

Use the **service-policy** interface configuration command to attach a traffic policy to an interface and to specify the direction in which the policy should be applied (either on packets coming into the interface or on packets leaving the interface).

## Attaching Service Policies to Interfaces

```
router(config-if)#  
service-policy {input | output} policy-map-name
```

- **Attaches the specified service policy map to the input or output interface**

```
class-map HTTP  
  match protocol http  
!  
policy-map PM  
  class HTTP  
    bandwidth 2000  
  class class-default  
    bandwidth 6000  
!  
interface Serial0/0  
  service-policy output PM
```

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—3-19

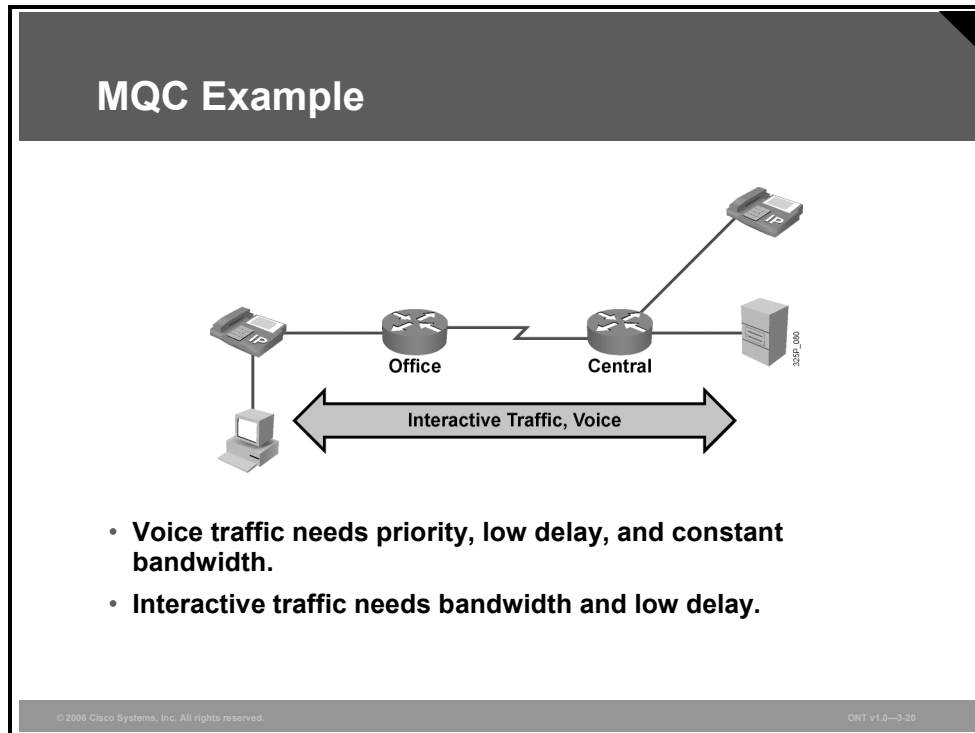
The router immediately verifies the parameters that are used in the policy map. If there is a mistake in the policy map configuration, the router displays a message explaining what is wrong with the policy map.

The sample configuration shows how a policy map is used to separate HTTP from other traffic. HTTP is guaranteed 2 Mbps of bandwidth. All other traffic belongs to the default class and is guaranteed 6 Mbps.



# MQC Example

This example shows a network using interactive traffic and VoIP with an applied MQC configuration.



In this scenario, the office site is connected over a low-speed WAN link to the central site. Both sites are equipped with IP phones, PCs, and servers that run interactive applications, such as terminal services. Because the available bandwidth is limited, an appropriate strategy for efficient bandwidth usage must be devised.

Voice traffic needs high priority, low delay, and constant bandwidth along the communication path. Interactive traffic requires bandwidth as well as low delay.

Classification of the important traffic streams and policing by applying traffic parameters to the classified traffic, such as priority, queuing, and bandwidth, are the major elements of the traffic policy that improve the overall quality. Finally, the traffic policy is applied to the WAN interface of the routers.

## MQC Example (Cont.)

```
hostname Office
!
class-map VoIP
match access-group 100
class-map Application
match access-group 101
!
policy-map QoS-Policy
class VoIP
priority 100
class Application
bandwidth 25
class class-default
fair-queue
!
interface Serial0/0
service-policy output QoS-Policy
!
access-list 100 permit ip any any precedence 5
access-list 100 permit ip any any dscp ef
access-list 101 permit tcp any host 10.1.10.20
access-list 101 permit tcp any host 10.1.10.40
```

Classification

QoS Policy

QoS Policy on Interface

Classification

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0-3.21

The figure represents an example of the complex configuration tasks involved in using MQC on the router Office.

In general, an MQC configuration can be divided into four sections:

- **ACLs:** Matching the traffic that needs QoS. These matches can be based on IP precedence bits, IP differentiated services code point (DSCP), IP addresses, and TCP/IP ports.
- **Class maps:** Classifying the traffic according to its importance.
- **Policy maps:** Applying parameters such as bandwidth, queuing mechanisms, or priorities to the classified traffic.
- **Service policy:** Applied to the interface.

# Basic Verification Commands

This section describes the verification commands.

## Basic Verification Commands

```
router#  
show class-map
```

- Displays the class maps

```
router#  
show policy-map
```

- Displays the policy maps

```
router#  
show policy-map interface type number
```

- Displays the applied policy map on the interface

© 2006 Cisco Systems, Inc. All rights reserved.ONT v1.0—3-22

To display and verify basic QoS classes and policies configured by using the MQC, use the commands listed in the table.

## MQC Verification Commands

Command	Description
<code>show class-map</code>	Displays the configured classes
<code>show policy-map</code>	Displays the configured policy
<code>show policy-map interface</code>	Displays the applied policy map on an interface

# Cisco AutoQoS

This topic describes the use of Cisco AutoQoS for implementing QoS in a network.

## Cisco AutoQoS

- **Automatically discovers applications and provides appropriate QoS treatment**
- **Automatically generates initial and ongoing QoS policies**
- **Provides high-level business knobs and multidevice and domain automation for QoS**
- **Generates intelligent, automatic alerts and summary reports**
- **Enables automatic, seamless interoperability among all QoS features and parameters across a network topology—LAN, MAN, and WAN**

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—3-24

Cisco AutoQoS simplifies and shortens the QoS deployment cycle. Cisco AutoQoS helps in all five major aspects of successful QoS deployments:

- **Application classification:** Cisco AutoQoS leverages intelligent classification on routers using Cisco NBAR to provide deep and stateful packet inspection. AutoQoS uses Cisco Discovery Protocol (CDP) for voice packets to ensure that the device attached to the LAN is really an IP phone.
- **Policy generation:** Cisco AutoQoS evaluates the network environment and generates an initial policy. AutoQoS automatically determines WAN settings for fragmentation, compression, encapsulation, and Frame Relay-to-ATM Service Interworking (FRF.8), eliminating the need to understand QoS theory and design practices in various scenarios. Customers can meet additional or special requirements by modifying the initial policy as they normally would.

The first release of Cisco AutoQoS provides the necessary AutoQoS VoIP feature to automate QoS settings for VoIP deployments. This feature automatically generates interface configurations, policy maps, class maps, and ACLs. Cisco AutoQoS VoIP will automatically employ Cisco NBAR to classify voice traffic and mark the traffic with the appropriate DSCP value. AutoQoS VoIP can be instructed to rely on, or trust, the DSCP markings previously applied to the packets.

- **Configuration:** With one command, Cisco AutoQoS configures the port to prioritize voice traffic without affecting other network traffic, while still offering the flexibility to adjust QoS settings for unique network requirements.

Not only will Cisco AutoQoS automatically detect Cisco IP phones and enable QoS settings, it will disable the QoS settings when a Cisco IP phone is relocated or moved to prevent malicious activity.

Cisco AutoQoS-generated router and switch configurations are customizable using the MQC.

- **Monitoring and reporting:** Cisco AutoQoS provides visibility into the classes of service deployed via system logging and Simple Network Management Protocol (SNMP) traps, with notification of abnormal events (that is, VoIP packet drops).
- **Consistency:** When you deploy QoS configurations using Cisco AutoQoS, the configurations generated are consistent among router and switch platforms. This level of consistency ensures seamless QoS operation and interoperability within the network.

Using Cisco AutoQoS, network administrators can implement the QoS features that are required for VoIP traffic without an in-depth knowledge of these underlying technologies:

- PPP
- Frame Relay
- ATM
- Link efficiency mechanisms, such as link fragmentation and interleaving (LFI)

The AutoQoS VoIP feature simplifies QoS implementation and speeds up the provisioning of QoS technology over a Cisco network. AutoQoS VoIP also reduces human error and lowers training costs. With the AutoQoS VoIP feature, one command (the **auto qos** command) enables QoS for VoIP traffic across every Cisco router and switch.

Network administrators can also use existing Cisco IOS commands to modify the configurations that are automatically generated by the AutoQoS VoIP feature in case the default Cisco AutoQoS configuration is not sufficient.

The Cisco SDM QoS wizard can be used in conjunction with the AutoQoS VoIP feature to provide a centralized, web-based tool to cost-effectively manage and monitor networkwide QoS policies. The AutoQoS VoIP feature, together with the SDM QoS wizard, eases QoS implementation, provisioning, and management.

---

<b>Note</b>	Cisco AutoQoS was introduced in Cisco IOS Software Release 12.2(15)T.
-------------	---

---

# The Features of Cisco AutoQoS

This subtopic describes the features of Cisco AutoQoS.

The Features of Cisco AutoQoS		
DiffServ Function	Cisco IOS and Catalyst Software QoS Feature	Behavior
Classification	NBAR DSCP, port	Classifies VoIP based on packet attributes or port trust
Marking	Class-based marking	Sets Layer 2 and Layer 3 attributes to categorize packets into a class
Congestion management	Percentage-based LLQ, WRR	Provides Expedited Forwarding treatment to voice and best-effort treatment to data
Shaping	Class-based shaping or FRTS	Shapes to CIR to prevent burst and smooth traffic to configured rate
Link efficiency mechanism	Header compression	Reduces the VoIP bandwidth requirement
Link efficiency mechanism	Link Fragmentation and Interleaving	Reduces jitter experienced by voice packets

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—3-25

Cisco AutoQoS performs these functions in a WAN:

- Automatically classifies Real-Time Transport Protocol (RTP) payload and VoIP control packets (H.323, H.225 unicast, Skinny, Session Initiation Protocol [SIP], and Media Gateway Control Protocol [MGCP]).
- Builds service policies for VoIP traffic that can be modified by using the MQC.
- Provisions LLQ for VoIP bearer and bandwidth guarantees for control traffic.
- Enables WAN traffic shaping, such as Frame Relay traffic shaping (FRTS), that adheres to Cisco best practices, where required. Parameters such as Committed Information Rate (CIR) and burst can be used for traffic shaping.
- Enables link efficiency mechanisms, such as LFI and compressed RTP (cRTP), where required.
- Provides SNMP and syslog alerts for VoIP packet drops.

Cisco AutoQoS performs these functions in a LAN:

- Enforces the trust boundary on Cisco Catalyst switch access ports, uplinks, and downlinks
- Enables Cisco Catalyst strict-priority queuing (also known as expedited queuing) with weighted round robin (WRR) scheduling for voice and data traffic, where appropriate
- Configures queue admission criteria (maps class of service [CoS] values in incoming packets to the appropriate queues)
- Enables NBAR for different traffic types
- Modifies queue sizes and weights, where required

# Cisco AutoQoS Usage Guidelines

This section describes the Cisco AutoQoS usage guidelines.

## Cisco AutoQoS Usage Guidelines

- **Make sure that:**
  - **CEF is enabled.**
  - **NBAR is enabled.**
  - **Correct bandwidth statement is configured on the interface.**
- **Finally, enable Cisco AutoQoS on the interface.**

© 2006 Cisco Systems, Inc. All rights reserved. OMT v1.0—3-26

As a general guideline to Cisco AutoQoS implementation, you must ensure that these services and parameters are available:

- Step 1** Cisco Express Forwarding (CEF) needs to be active. CEF is a prerequisite for NBAR.
- Step 2** NBAR needs to be enabled, because Cisco AutoQoS uses it for the traffic classification.
- Step 3** Make sure that the correct bandwidth is configured on the interface. AutoQoS takes the interface type and bandwidth into consideration when implementing these QoS features:
- **LLQ:** The LLQ (specifically, PQ) is applied to the voice packets to meet the latency requirements.
  - **cRTP:** With cRTP, the 40-byte IP header of the voice packet is reduced to 2 or 4 bytes (without or with cyclic redundancy check [CRC]), reducing voice bandwidth requirements. cRTP must be applied at both ends of a network link.
  - **LFI:** LFI is used to reduce the jitter of voice packets by preventing voice packets from being delayed behind large data packets in a queue. LFI must be applied at both ends of a network link.
- Step 4** Finally, enable Cisco AutoQoS on the interface.

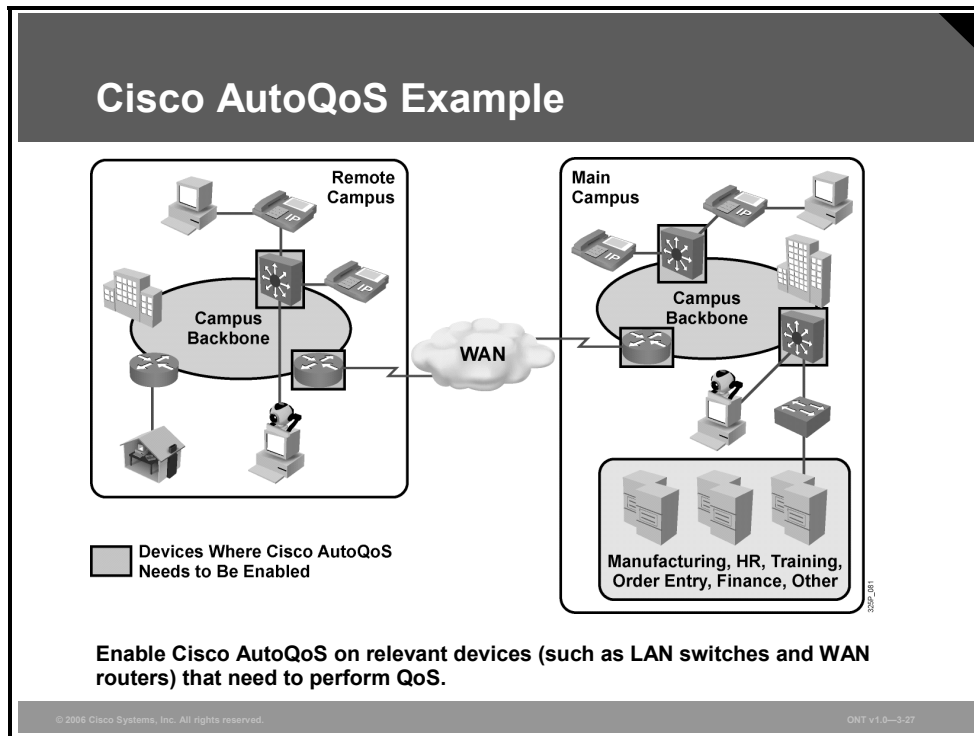
---

**Note** A prerequisite for Cisco AutoQoS Enterprise is that any pre-existing QoS configuration be removed from the (WAN) interface as the first step.

---

## Cisco AutoQoS Example

This example shows an implementation of Cisco AutoQoS in a campus network environment, followed by the configuration output.



In this scenario, two campus sites are connected over a WAN. For ease of deployment, AutoQoS VoIP is chosen for the campuswide QoS setup. Only the relevant devices, such as the LAN switches on which servers, IP phones, PCs, and videoconferencing systems are connected, as well as the WAN routers that carry the traffic into the Internet, are enabled with the AutoQoS VoIP feature.

AutoQoS VoIP will install the appropriate QoS policy for these devices.



## Cisco AutoQoS Example (Cont.)

```
interface Serial1/3
  ip cef
  bandwidth 1540
  ip address 10.10.100.1 255.255.255.0
  auto qos voip
```

} IP CEF and Bandwidth

} AutoQoS for VoIP Traffic Recognized by NBAR

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—3-28

The figure represents an example of the configuration tasks involved in using Cisco AutoQoS:

- **ip cef** command
- **bandwidth** statement
- **auto qos** command

All three commands are applied to the interface. The correct QoS policy will be generated by Cisco AutoQoS.

# Cisco SDM QoS Wizard

This topic explains the Cisco SDM QoS wizard.

## Cisco SDM QoS Wizard

- **Cisco SDM is an intuitive, web-based device management tool for easy and reliable deployment and management of Cisco IOS routers.**
- **Cisco SDM provides wizards for:**
  - **Firewall and NAT**
  - **Intrusion prevention**
  - **IPsec VPNs**
  - **QoS**
  - **Routing**

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0-3-30

Cisco Router and Security Device Manager (SDM) allows you to easily configure routing, security, and QoS services on Cisco routers while helping to enable proactive management through performance monitoring. Whether you are deploying a new router or installing Cisco SDM on an existing router, you can now remotely configure and monitor these routers without using the Cisco IOS software CLI. The Cisco SDM GUI aids nonexpert users of Cisco IOS software in day-to-day operations, provides easy-to-use smart wizards, automates router security management, and assists you through comprehensive online help and tutorials.

Cisco SDM smart wizards guide you step by step through router and security configuration workflow by systematically configuring the LAN and WAN interfaces, firewall, Network Address Translation (NAT) intrusion prevention system (IPS), and IPsec virtual private network (VPNs) routing, and QoS. Cisco SDM smart wizards can intelligently detect incorrect configurations and propose fixes. Online help embedded within Cisco SDM contains appropriate background information, in addition to step-by-step procedures to help you enter correct data in Cisco SDM. In the QoS configuration section of Cisco SDM, there are several options to define traffic classes and configure QoS policies in the network.

## QoS Features

Cisco SDM supports a wide range of Cisco IOS software releases and is available free on Cisco router models from the Cisco 830 Routers to the Cisco 7301 Router. It ships preinstalled on all new Cisco 850 Series, Cisco 870 Series, Cisco 1800 Series, Cisco 2800 Series, and Cisco 3800 Series Integrated Services Routers.

### QoS Features

- **Cisco SDM QoS wizard provides:**
  - **QoS policing**
  - **NBAR**
  - **Traffic monitoring**
- **Supported and preinstalled on Cisco 850, 870, 1800, 2800, and 3800 Cisco Integrated Services Routers**
- **Supported on devices 830, 1700, 2600 XM, 2800, 3700, 7200 VXR, and 7301**

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—3-31

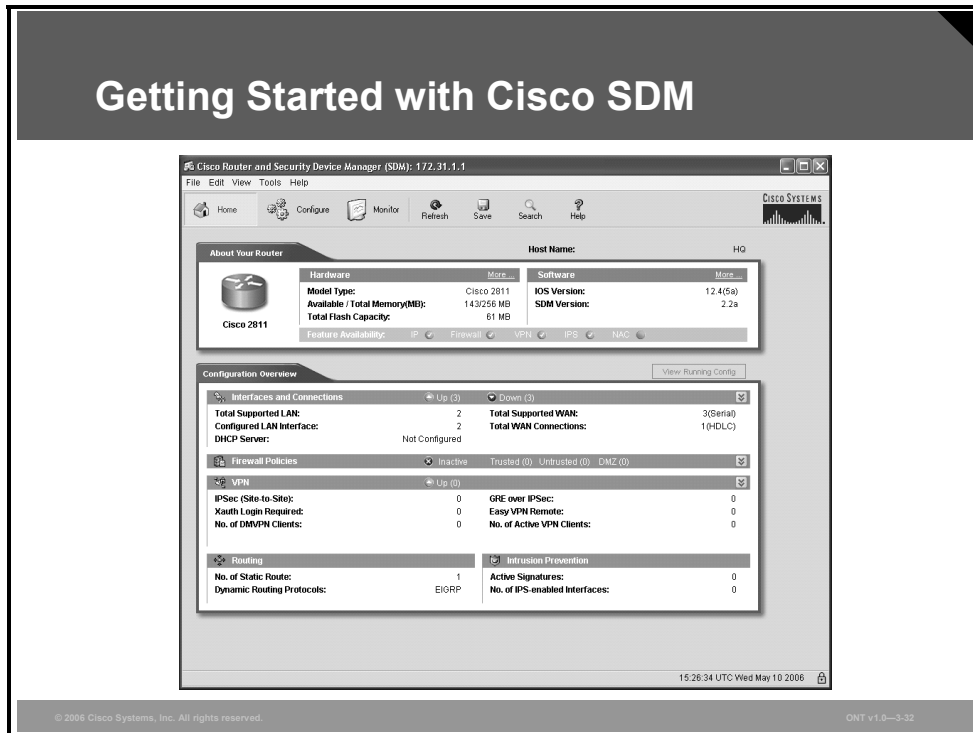
The Cisco SDM QoS wizard offers easy and effective optimization of LAN, WAN, and VPN bandwidth and application performance for different business needs (for example, voice and video, enterprise applications, and web). Three predefined categories are:

- Real-time
- Business-critical
- Best-effort

In addition, the Cisco SDM QoS wizard supports NBAR, which provides real-time validation of application usage of WAN bandwidth against predefined service policies as well as QoS policing and traffic monitoring.

# Getting Started with Cisco SDM

This section explains the Cisco SDM QoS wizard user interface.



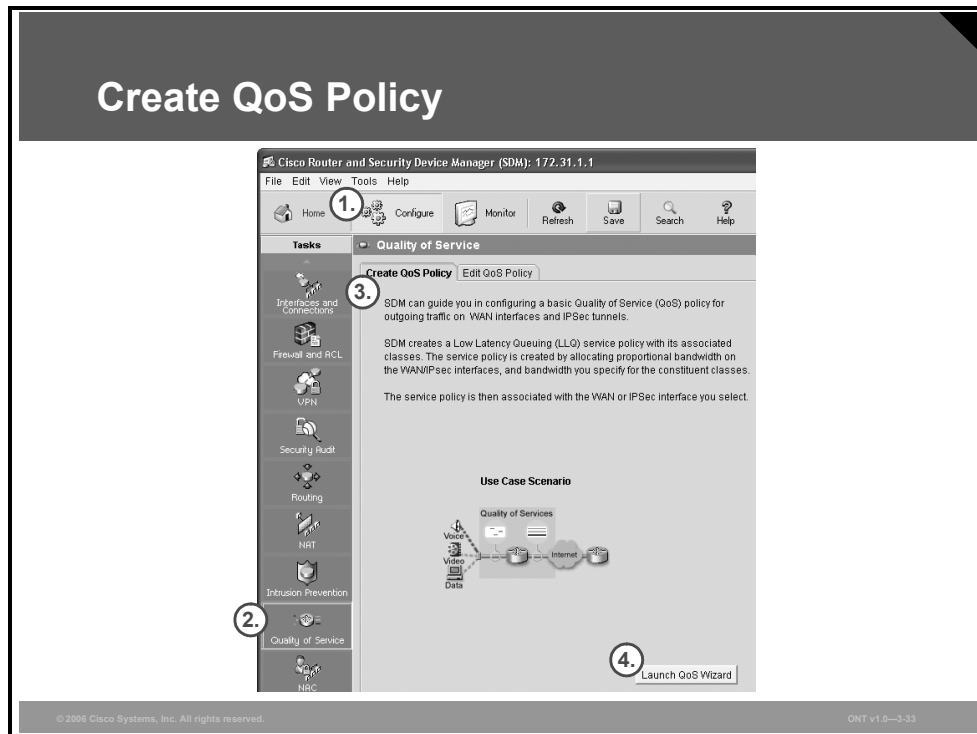
The main page of Cisco SDM consists of two sections:

- **About Your Router:** This section displays the hardware and software configuration of the router.
- **Configuration Overview:** This section displays basic traffic statistics.

There are two important icons in the top horizontal navigation bar:

- Clicking the Configure icon opens the configuration page.
- Clicking the Monitor icon opens the page where the status of the tunnels, interfaces and device can be monitored.

# Create QoS Policy

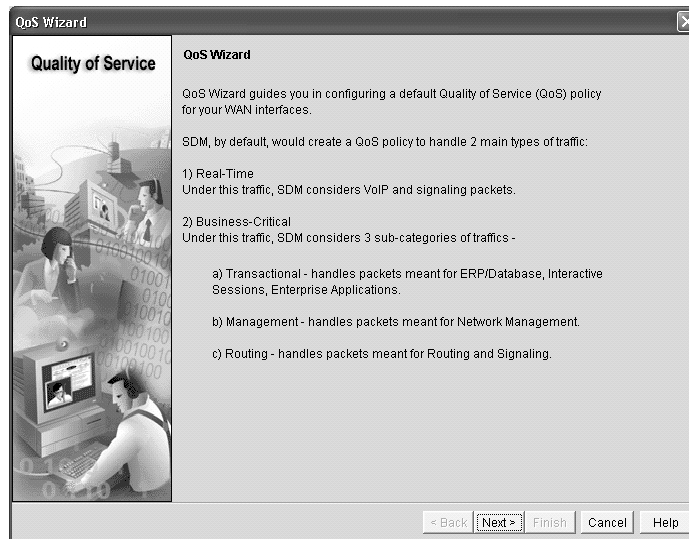


## Creation of a QoS Policy

To create a QoS policy using the Cisco SDM GUI, complete these steps:

- Step 1** Enter configuration mode by clicking **Configure** in the top toolbar of the SDM window.
- Step 2** Click the **Quality of Service** button in the Tasks toolbar at the left side of the SDM window.
- Step 3** Click the **Create QoS Policy** tab.
- Step 4** Click **Launch QoS Wizard** to launch the wizard.

## QoS Wizard



## QoS Wizard

- Step 5** Cisco SDM informs you that it will configure two classes: real-time and business-critical. Click **Next** to proceed.

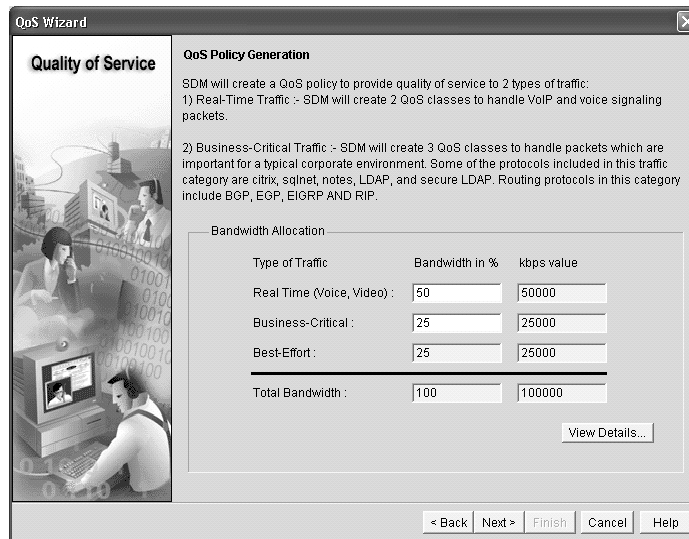
## Interface Selection



## Interface Selection

- Step 6** You are asked to select an interface on which you want a QoS policy. Click **Next** to proceed.

## QoS Policy Generation



**Quality of Service**

**QoS Policy Generation**

SDM will create a QoS policy to provide quality of service to 2 types of traffic:

- 1) Real-Time Traffic :- SDM will create 2 QoS classes to handle VoIP and voice signaling packets.
- 2) Business-Critical Traffic :- SDM will create 3 QoS classes to handle packets which are important for a typical corporate environment. Some of the protocols included in this traffic category are citrix, sqlnet, notes, LDAP, and secure LDAP. Routing protocols in this category include BGP, EGP, EIGRP AND RIP.

**Bandwidth Allocation**

Type of Traffic	Bandwidth in %	kbps value
Real Time (Voice, Video) :	50	50000
Business-Critical :	25	25000
Best-Effort :	25	25000
<b>Total Bandwidth :</b>	<b>100</b>	<b>100000</b>

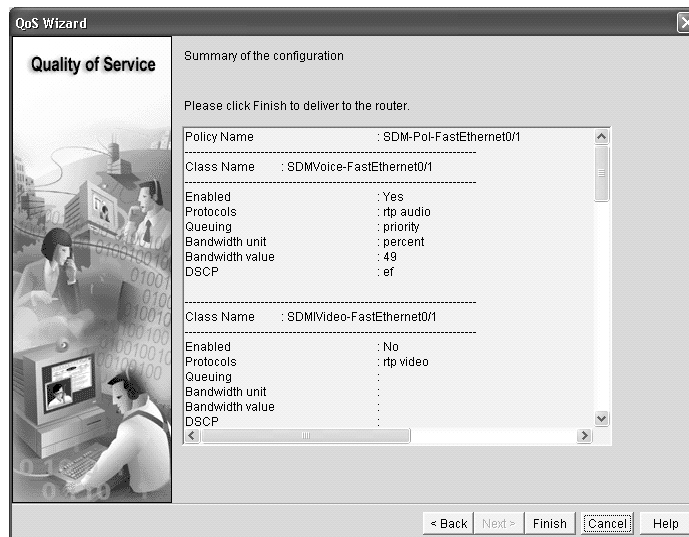
[View Details...](#)

< Back Next > Finish Cancel Help

## QoS Policy Generation

- Step 7** You are prompted to enter the percentages for each class. After you enter the numbers, Cisco SDM will automatically calculate the best-effort class and the bandwidth requirements for each class. Click **Next** to proceed.

## Summary



**Quality of Service**

**Summary of the configuration**

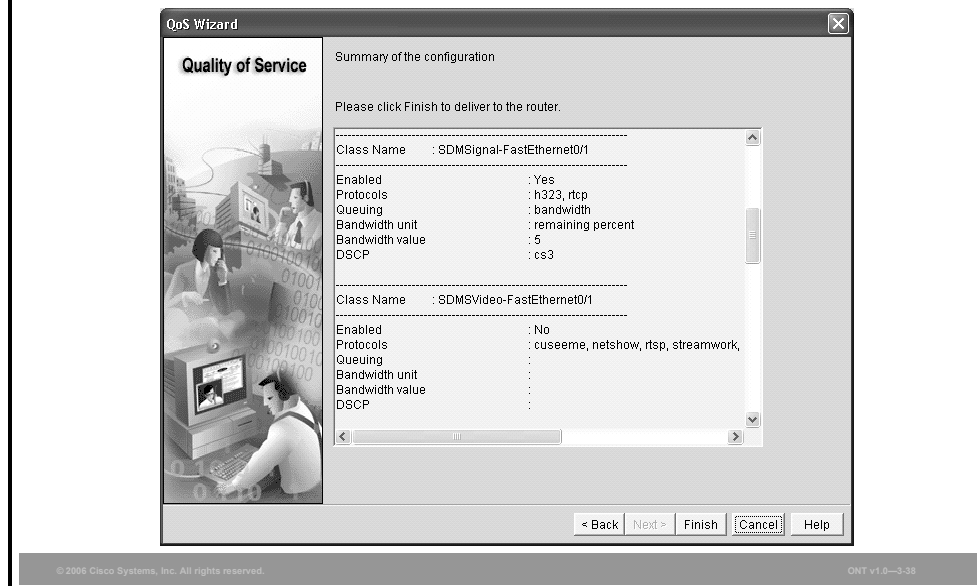
Please click Finish to deliver to the router.

Policy Name	: SDM-Pol-FastEthernet0/1
Class Name	: SDMVoice-FastEthernet0/1
Enabled	: Yes
Protocols	: rtp audio
Queuing	: priority
Bandwidth unit	: percent
Bandwidth value	: 49
DSCP	: ef
Class Name	: SDMMVideo-FastEthernet0/1
Enabled	: No
Protocols	: rtp video
Queuing	:
Bandwidth unit	:
Bandwidth value	:
DSCP	:

< Back Next > Finish Cancel Help



## Summary (Cont.)



In the Summary window, Cisco SDM shows the QoS configuration that you configured and that will be applied to your router after you click the **Finish** button.

These two figures present the settings that will be applied with the SDMVoice-FastEthernet0/1, SDMVideo-FastEthernet0/1, SDMSignal-FastEthernet0/1, and SDMSVideo-FastEthernet0/1 classes.

## Summary (Cont.)

**QoS Wizard**

**Quality of Service**

Summary of the configuration

Please click Finish to deliver to the router.

Class Name : SDMTrans-FastEthernet0/1	
Enabled	: Yes
Protocols	: ctrip, finger, notes, novadgm, panywf
Queueing	: bandwidth
Bandwidth unit	: remaining percent
Bandwidth value	: 79
DSCP	: af21

Class Name : SDMMManage-FastEthernet0/1	
Enabled	: Yes
Protocols	: dhcp, dns, imap, kerberos, ldap, secur
Queueing	: bandwidth
Bandwidth unit	: remaining percent
Bandwidth value	: 8
DSCP	: cs2

< Back Next > Finish Cancel Help

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0-3-39

## Summary (Cont.)

**QoS Wizard**

**Quality of Service**

Summary of the configuration

Please click Finish to deliver to the router.

Class Name : SDMRout-FastEthernet0/1	
Enabled	: Yes
Protocols	: bgp, egp, eigrp, ospf, rip, rsvp
Queueing	: bandwidth
Bandwidth unit	: remaining percent
Bandwidth value	: 8
DSCP	: cs6

Class Name : SDMBulk-FastEthernet0/1	
Enabled	: No
Protocols	: exchange, ftp, irc, nntp, pop3, printer, s
Queueing	:
Bandwidth unit	:
Bandwidth value	:
DSCP	:

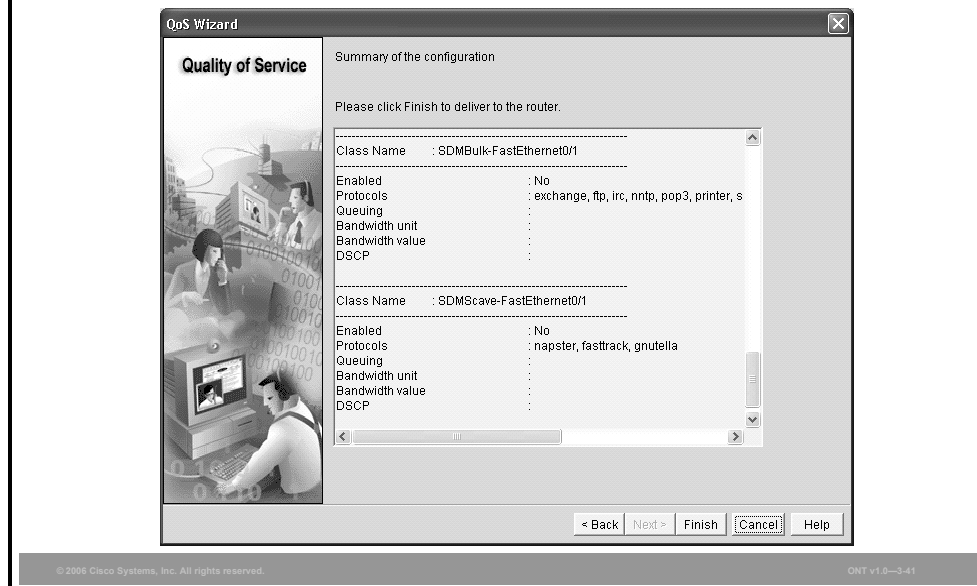
< Back Next > Finish Cancel Help

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0-3-40

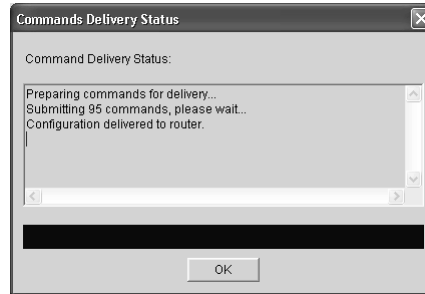
These two figures present the settings that will be applied with the SDMTrans-FastEthernet0/1, SDMMManage-FastEthernet0/1, SDMRout-FastEthernet0/1, and SDMBulk-FastEthernet0/1 classes.

## Summary (Cont.)



This figure presents the settings that will be applied with the SDMSave-FastEthernet0/1 class.

## Command Delivery Status

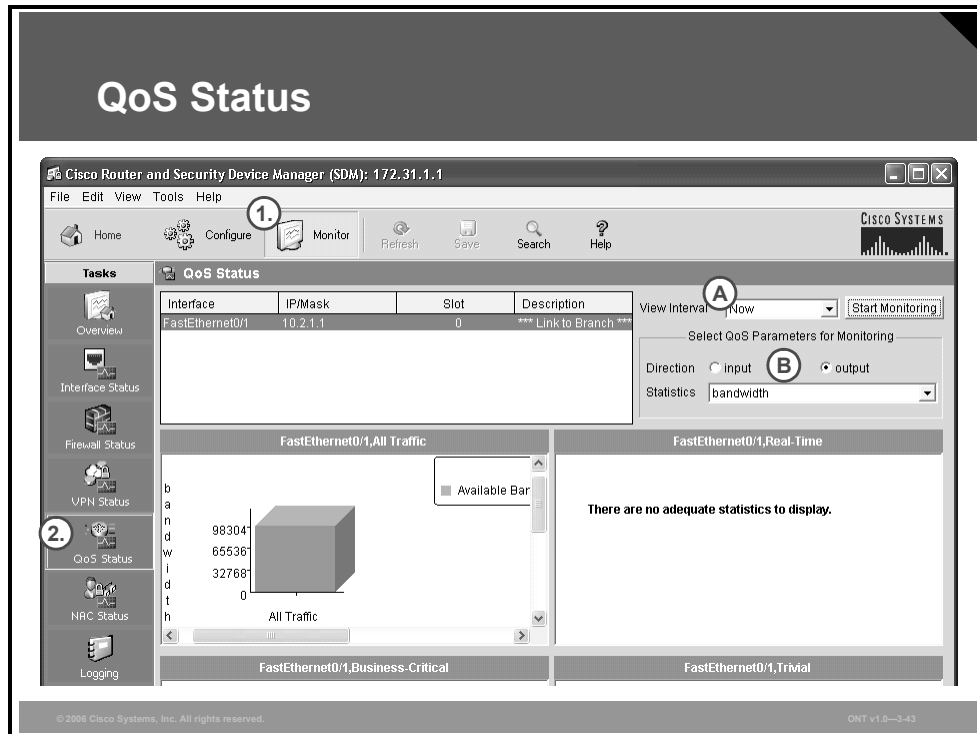


© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0-3-42

## Command Delivery Status

- Step 8** The next window shows the progress of the delivery of the configuration to the window. When the commands have been delivered to the router, click **OK**.



## QoS Status

After QoS is configured, you can monitor its status:

- Step 1** To enter the monitor mode, click the **Monitor** icon in the toolbar at the top of the Cisco SDM window.
- Step 2** Click the **QoS Status** icon in the Tasks toolbar at the left side of the SDM window.

The traffic statistics are displayed in bar charts based on the combination of the selected interval and QoS parameters for monitoring:

- The interval can be changed using the View Interval drop-down menu. The options available are Now, Every 1 Minute, Every 5 Minutes, and Every 1 Hour.
- QoS parameters for monitoring include Direction (input and output) and Statistics (bandwidth, bytes, and packets dropped).

# QoS Implementation Methods Compared

This topic describes the advantages and disadvantages of each of the methods of implementing QoS on a network.

QoS Implementation Methods Compared				
	Legacy CLI	MQC	Cisco AutoQoS	Cisco SDM QoS Wizard
Ease of use	Poor	Easier	Simple	Simple
Ability to fine-tune	OK	Very good	Limited	Limited
Time to implement	Longest	Average	Shortest	Short
Modularity	Poor	Excellent	Excellent	Very good

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0-3.45

The four methods for configuring QoS on a network are the legacy CLI method, the MQC method, Cisco AutoQoS, and Cisco SDM QoS wizard. It is recommended that you use MQC or Cisco AutoQoS for implementing QoS.

Although MQC is much easier to use than the CLI method, Cisco AutoQoS can simplify the configuration of QoS. As a result, the fastest implementation possible can usually be accomplished with Cisco AutoQoS.

MQC offers excellent modularity and the ability to fine-tune complex networks. AutoQoS offers the fastest way to implement QoS, but has limited fine-tuning capabilities. When an AutoQoS configuration has been generated, use CLI commands to fine-tune the configuration if necessary.

---

**Note** On most networks, fine-tuning will not be necessary for Cisco AutoQoS.

---

The Cisco SDM QoS wizard allows you to easily configure QoS services and other features, such as IPsec and VPNs, on Cisco routers, while enabling proactive management through performance monitoring. You can now remotely configure and monitor your Cisco routers without using the Cisco IOS software CLI. The GUI aids nonexpert users of Cisco IOS software in day-to-day operations and provides easy-to-use smart wizards for configuring QoS policies.

# Summary

This topic summarizes the key points that were discussed in this lesson.

## Summary

- There are four methods for implementing QoS: legacy CLI, MQC, Cisco AutoQoS, and Cisco SDM QoS wizard.
- CLI QoS configuration can be complex and in many cases requires learning different syntax for different QoS mechanisms.
- MQC separates the classification of network traffic from the definition of the QoS policy.
- Cisco AutoQoS is used to automatically implement a set of QoS policies on a router or a switch.
- Cisco SDM QoS wizard provides a GUI to ease QoS configuration.
- MQC is the recommended manual approach to configure QoS. MQC reduces configuration steps and time compared to the legacy approach.

© 2006 Cisco Systems, Inc. All rights reserved.

QNT v1.0—3-46

# Module Summary

This topic summarizes the key points that were discussed in this module.

## Module Summary

- **The problems that can lead to poor QoS for applications running on a converged network include lack of bandwidth, excessive delay, jitter, and packet loss.**
- **To implement QoS on a converged network, follow this process:**
  - **Identify the traffic types and their requirements.**
  - **Classify the traffic.**
  - **Define and implement QoS policies for each traffic class.**
- **The three QoS models are best effort, IntServ, and DiffServ.**
- **In the best effort model, no special QoS mechanisms are applied.**
- **With IntServ, applications signal their QoS requirements.**
- **With DiffServ, network devices recognize the traffic classes and provide different QoS levels.**
- **The four techniques to implement QoS are the legacy CLI method, MQC, Cisco AutoQoS, and Cisco SDM QoS wizard.**

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—3-1

Converged IP networks can suffer from poor quality of service (QoS) for several reasons, such as low bandwidth, excessive delay, jitter, and packet loss. The goal of implementing QoS is to decrease the effect of these factors with the use of queuing, compression, prioritization, and link efficiency mechanisms.

There are three basic QoS models: best effort, Integrated Services (IntServ), and Differentiated Services (DiffServ). The best-effort model does not provide any QoS, while the IntServ model relies on applications signaling QoS requirements to the network. DiffServ is the most scalable model for implementing the QoS required for modern converged networks. With DiffServ, traffic is classified into different traffic classes and then marked. The network QoS policy then enforces differentiated services based on the markings.

To implement the QoS policy, different mechanisms are required. There are four different implementation techniques for QoS: the legacy command-line interface (CLI) method used for basic QoS deployments, the Modular QoS CLI (MQC) for high-level deployments and QoS fine-tuning, Cisco AutoQoS for general QoS setups, and, finally, Cisco Router and Security Device Monitor (SDM) QoS wizard, a web-based application for QoS deployments in the enterprise environment.



# Module Self-Check

Use the questions here to review what you learned in this module. The correct answers and solutions are found in the Module Self-Check Answer Key.

- Q1) Which two are issues that affect QoS? (Choose two.) (Source: Introducing QoS)
- A) router platform
  - B) available bandwidth
  - C) Cisco IOS version
  - D) end-to-end delay
  - E) packet size
- Q2) What is the maximum available bandwidth in a path equal to? (Source: Introducing QoS)
- A) the bandwidth of the fastest link
  - B) the bandwidth of the longest link
  - C) the bandwidth of the slowest link
  - D) the bandwidth of the shortest link
- Q3) Which delay represents the time that it takes a router to put a packet on the wire? (Source: Introducing QoS)
- A) serialization delay
  - B) packet delay
  - C) queuing delay
  - D) propagation delay
- Q4) There are several techniques to reduce packet loss that is caused by congestion. Which technique prevents congestion by dropping some packets before congestion occurs? (Source: Introducing QoS)
- A) LLQ
  - B) WRED
  - C) DPD
  - D) MQC
- Q5) Which is the initial step to deploy QoS in a network? (Source: Introducing QoS)
- A) shape traffic
  - B) queue traffic
  - C) classify traffic
  - D) filter traffic
- Q6) In a converged network, VoIP traffic should be classified as \_\_\_\_\_. (Source: Introducing QoS)
- A) best-effort
  - B) mission-critical
  - C) scavenger
  - D) high-priority

- Q7) Which traffic should typically have the highest priority in a converged network? (Source: Introducing QoS)
- A) web browsing
  - B) e-mail
  - C) VoIP
  - D) file transfer
- Q8) Which model for implementing QoS is the least scalable? (Source: Identifying Models for Implementing QoS)
- A) best-effort
  - B) IntServ
  - C) DiffServ
  - D) FirstServ
- Q9) How are different packets treated in a best-effort network? (Source: Identifying Models for Implementing QoS)
- A) All the packets get high priority.
  - B) All the packets get medium priority.
  - C) All the packets get low priority.
  - D) Packets are not prioritized.
- Q10) Which two IP QoS mechanisms work together to provide a set of complete integrated services on a network? (Choose two.) (Source: Identifying Models for Implementing QoS)
- A) GTS
  - B) CAR
  - C) LLQ
  - D) GTI
  - E) RSVP
- Q11) What is RSVP used for in an IntServ network? (Source: Identifying Models for Implementing QoS)
- A) RSVP is used as a routing protocol.
  - B) RSVP is used for signalization.
  - C) RSVP is used for packet tracing.
  - D) RSVP is used as a helper protocol for CDP.
- Q12) To which entities are the services in the DiffServ model provided? (Source: Identifying Models for Implementing QoS)
- A) frames
  - B) packets
  - C) applications
  - D) classes of traffic
- Q13) Which implementation can be used to provide the simplest QoS configuration to accommodate VoIP traffic? (Source: Identifying Methods for Implementing QoS)
- A) AutoVoIP
  - B) AutoQoS
  - C) MQC
  - D) legacy CLI

- Q14) Using the legacy CLI method is the most time-consuming way to implement QoS. (Source: Identifying Methods for Implementing QoS)
- A) true
  - B) false
- Q15) Which is the most important benefit of MQC? (Source: Identifying Methods for Implementing QoS)
- A) separation of classification from QoS mechanisms
  - B) no traffic policy needed
  - C) QoS implementation in a single command
  - D) number of commands
- Q16) What are the two prerequisites to enable AutoQoS VoIP? (Choose two.) (Source: Identifying Methods for Implementing QoS)
- A) LLQ
  - B) CDP
  - C) RSVP
  - D) Cisco Express Forwarding
  - E) bandwidth of the interface
- Q17) Which three are the purposes of the Cisco SDM QoS wizard? (Choose three.) (Source: Identifying Methods for Implementing QoS)
- A) implementing QoS
  - B) monitoring QoS
  - C) troubleshooting QoS
  - D) securing QoS
- Q18) Is Cisco AutoQoS the fastest way to set up QoS in a network? (Source: Identifying Methods for Implementing QoS)
- A) yes
  - B) no

## Module Self-Check Answer Key

Q1)	B, D
Q2)	C
Q3)	A
Q4)	B
Q5)	C
Q6)	D
Q7)	C
Q8)	B
Q9)	D
Q10)	C, E
Q11)	B
Q12)	D
Q13)	B
Q14)	A
Q15)	A
Q16)	D, E
Q17)	A, B, C
Q18)	A

# Implement the DiffServ QoS Model

---

## Overview

In any network where networked applications require differentiated levels of service, traffic must be sorted into different classes to which quality of service (QoS) is applied. Classification, marking, and queuing are critical functions of any successful QoS implementation. Classification allows network devices to identify traffic as belonging to a specific class with the specific QoS requirements determined by an administrative QoS policy. After network traffic is sorted, individual packets are marked so that other network devices can apply QoS features uniformly to those packets in compliance with the defined QoS policy. Queuing dispatches the packets according to their markings. This module introduces classification, marking, and queuing using various methods of implementation.

## Module Objectives

Upon completing this module, you will be able to explain the key IP QoS mechanisms used to implement the DiffServ QoS model. This ability includes being able to meet these objectives:

- Explain the purpose of classification and marking and how they can be used to define a QoS service class
- Explain Cisco MQC class-based classification and marking operations and configuration using NBAR
- Explain Cisco queuing operations and basic configurations
- Explain the procedure for configuring the WFQ queuing mechanisms
- Explain the procedure for configuring queuing mechanisms, including CBWFQ and LLQ, on a router
- Explain Cisco CBWRED operations and basic configurations
- Explain Cisco class-based traffic-policing and class-based traffic-shaping operations and basic configurations
- Explain Cisco class-based header-compression operations and basic configurations

- Explain the purpose and basic configuration of QoS preclassify for traffic going over IPsec and GRE tunnels
- Describe the set of QoS mechanisms used to implement Cisco end-to-end QoS best practices in a typical enterprise network connected through a service provider that is providing Layer 3 IP services

# Introducing Classification and Marking

---

## Overview

Quality of service (QoS) offers the ability to provide different levels of treatment to specific classes of traffic. Before any QoS applications or mechanisms can be applied, traffic must be identified and grouped into classes. QoS is then applied to these traffic classes. Network devices use classification to identify traffic as belonging to a specific class. After network traffic is sorted, marking can be used to color (tag) individual packets so that network devices can apply QoS features uniformly to those packets as they travel through the network. This lesson introduces the concepts of classification and marking, explains the markings that are available at the data link and network layers, and identifies where classification and marking should be used in a network. The concept of a QoS service class and how a service class can be used to represent an application or set of applications is also discussed.

## Objectives

Upon completing this lesson, you will be able to explain the purpose of classification and marking and how they can be used to define a QoS service class. This ability includes being able to meet these objectives:

- Explain the purpose of packet classification
- Explain the purpose of packet marking
- Describe IP packet classification and marking at the data link layer
- Explain the purpose and function of the DiffServ model
- Describe the interoperability between DSCP-based and IP-precedence-based devices in a network
- Describe how DSCP values are determined and assigned to different PHBs
- Describe DSCP settings in the DiffServ model
- Describe data link layer-to-network layer interoperability between QoS markings
- Explain the term “QoS service class” and how service classes can be used to create a service policy throughout a network

- Explain how link layer and network layer markings are used to define QoS service classes and the different applications represented by each of these service classes
- Explain the concept of trust boundaries and how they are used with classification and marking



# Classification

This topic describes the purpose of packet classification.

## Classification

- **Classification is the process of identifying and categorizing traffic into classes, typically based upon:**
  - Incoming interface
  - IP precedence
  - DSCP
  - Source or destination address
  - Application
- **Classification is the most fundamental QoS building block.**
- **Without classification, all packets are treated the same.**

© 2006 Cisco Systems, Inc. All rights reserved. OMT v1.0—4-3

Classification is the process of identifying traffic and categorizing that traffic into classes. Classification uses a traffic descriptor to categorize a packet within a specific group to define that packet. Typically used traffic descriptors include these:

- Incoming interface
- IP precedence
- differentiated services code point (DSCP)
- Source or destination address
- Application

After the packet has been classified or identified, the packet is then accessible for quality of service (QoS) handling on the network.

Using classification, network administrators can partition network traffic into multiple classes of service (CoSs). When traffic descriptors are used to classify traffic, the source implicitly agrees to adhere to the contracted terms and the network promises QoS. Various QoS mechanisms, such as traffic policing, traffic shaping, and queuing techniques, use the traffic descriptor of the packet (that is, the classification of the packet) to ensure adherence to that agreement.

Classification should take place at the network edge, typically in the wiring closet, within IP phones, or at network endpoints. It is recommended that classification occur as close to the source of the traffic as possible.

---

**Note** The term “classification” is interchangeable with the term “packet classification.”

---

# Marking

This topic describes the purpose of packet marking.

## Marking

- **Marking is the QoS feature component that “colors” a packet (frame) so it can be identified and distinguished from other packets (frames) in QoS treatment.**
- **Commonly used markers:**
  - **Link layer:**
    - **CoS (ISL, 802.1p)**
    - **MPLS EXP bits**
    - **Frame Relay**
  - **Network layer:**
    - **DSCP**
    - **IP precedence**

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—4-5

Marking is related to classification. Marking allows network devices to classify a packet or frame at the edge based on a specific traffic descriptor. Typically used traffic descriptors include these:

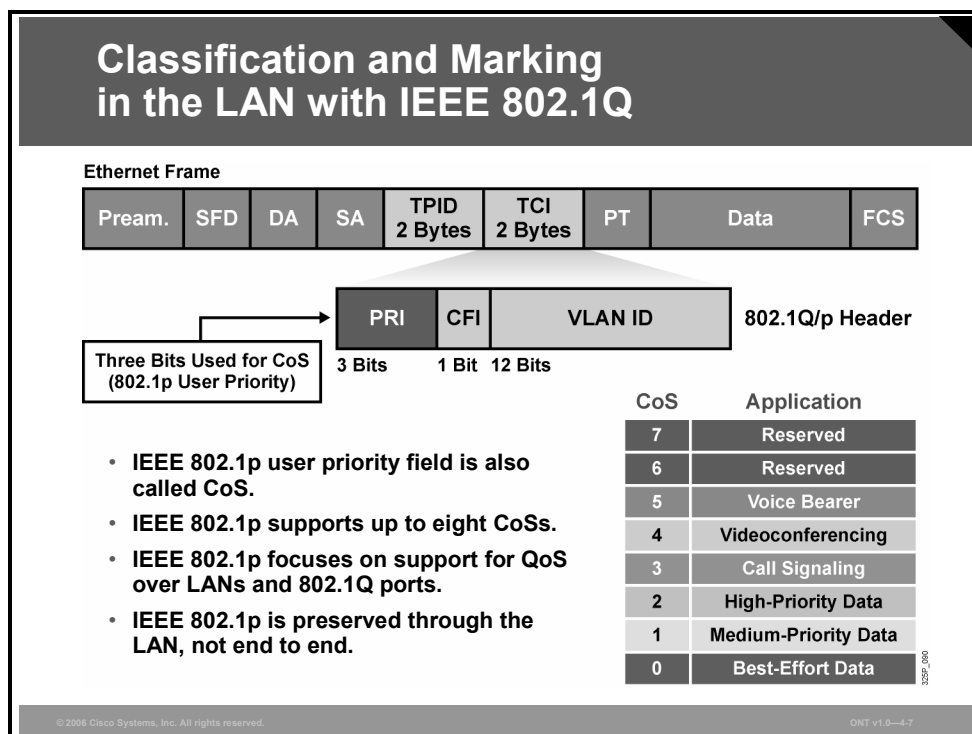
- **Link layer:**
  - CoS (Inter-Switch Link [ISL], 802.1p)
  - Multiprotocol Label Switching (MPLS) experimental (EXP) bits
  - Frame Relay
- **Network layer:**
  - DSCP
  - IP precedence

Marking can be used to set information in the Layer 2 frame or Layer 3 packet headers.

Marking a packet or frame with its classification allows subsequent network devices to easily distinguish the marked packet or frame as belonging to a specific class. After the packets or frames are identified as belonging to a specific class, QoS mechanisms can be uniformly applied to ensure compliance with administrative QoS policies.

# Classification and Marking at the Link Layer

This topic describes IP packet classification and marking options that are available at the data link layer.



The 802.1Q standard is an IEEE specification for implementing VLANs in Layer 2 switched networks. The 802.1Q specification defines two 2-byte fields (tag protocol identifier [TPID] and tag control information [TCI]) that are inserted within an Ethernet frame following the source address field. The TPID field is currently fixed and assigned the value 0x8100. The TCI field is composed of three fields:

- **User priority bits (PRI) (3 bits):** The specifications of this 3-bit field are defined by the IEEE 802.1p standard. These bits can be used to mark packets as belonging to a specific CoS. The CoS marking uses the three 802.1p user priority bits and allows a Layer 2 Ethernet frame to be marked with eight levels of priority (values 0–7). Three bits allow for eight levels of classification, allowing a direct correspondence with IP version 4 (IPv4) (IP precedence) type of service (ToS) values. The table lists the standard definitions the IEEE 802.1p specification defines for each CoS.

## Standard Definitions of CoSs

CoS	Definition
CoS 7 (111)	Network
CoS 6 (110)	Internet
CoS 5 (101)	Critical
CoS 4 (100)	Flash-override
CoS 3 (011)	Flash
CoS 2 (010)	Immediate
CoS 1 (001)	Priority
CoS 0 (000)	Routine

One disadvantage of using CoS markings is that frames lose their CoS markings when transiting a non-802.1Q or non-802.1p link. Therefore, a ubiquitous permanent marking should be used for network transit. This is typically accomplished by translating a CoS marking into another marker or simply using a different marking mechanism.

- **Canonical format indicator (CFI) (1 bit):** This bit indicates whether the bit order is canonical or noncanonical. The CFI bit is used for compatibility between Ethernet and Token Ring networks.
- **VLAN identifier (VLAN ID) (12 bits):** The VLAN ID field is a 12-bit field that defines the VLAN used by 802.1Q. The fact that the field is 12 bits long restricts the number of VLANs supported by 802.1Q to 4096. For most enterprise customers, 4096 VLANs is adequate. For service provider applications, 4096 VLANs may not be enough.

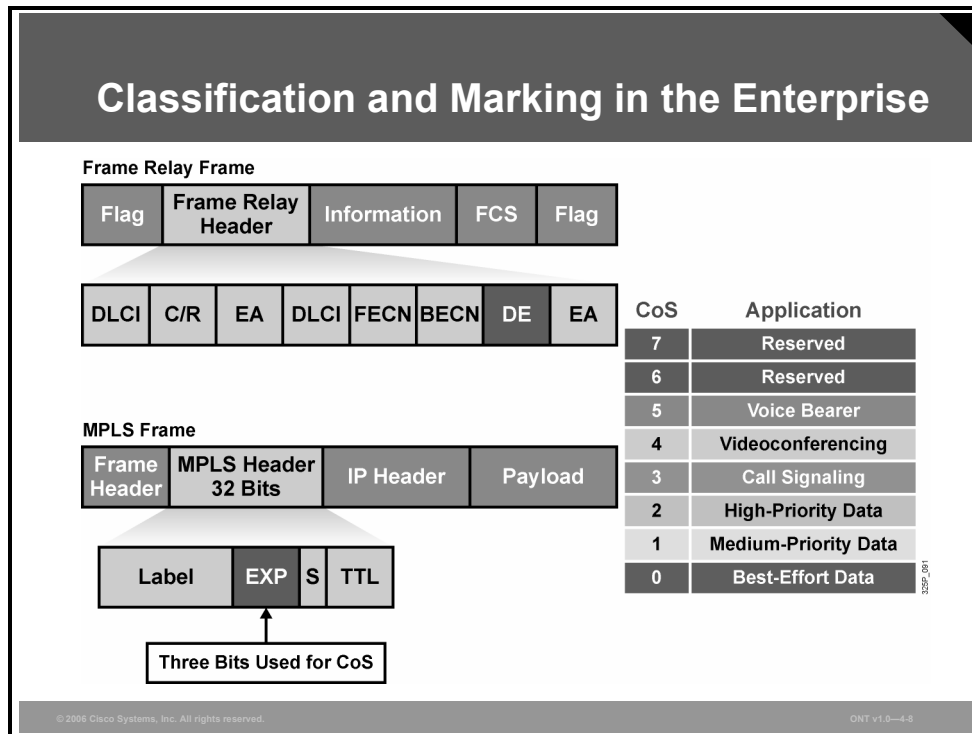
---

<b>Note</b>	ISL uses the same values as 802.1Q.
-------------	-------------------------------------

---

# Classification and Marking in the Enterprise

Before the Internet Engineering Task Force (IETF) defined QoS methods for the network layer, the ITU-T and the Frame Relay Forum (FRF) had already derived standards for link layer QoS in Frame Relay networks. Frame Relay provides a simple set of QoS mechanisms to ensure a committed information rate (CIR): congestion notifications called forward explicit congestion notification (FECN) and backward explicit congestion notification (BECN), in addition to fragmentation of data frames when voice frames are present, as described in Frame Relay Forum standard FRF.12.



One component of Frame Relay QoS is packet discard when congestion is experienced in the network. Frame Relay will allow network traffic to be sent at a rate exceeding its CIR. The frames that exceed the committed rate can be marked as discard eligible (DE) at the ingress Frame Relay switch. If congestion occurs in the network, frames marked DE will be discarded in preference to frames that are not marked.

## Marking in MPLS

When a customer transmits IP packets from one site to another, the IP precedence field (the first three bits of the DSCP field in the header of an IP packet) specifies the CoS. Based on the IP precedence marking, the packet is given the desired treatment, such as guaranteed bandwidth or latency. If the service provider network is an MPLS network, then the IP precedence bits are copied into the MPLS EXP field at the edge of the network. However, the service provider might want to set an MPLS packet QoS to a different value that is determined by the service offering.

The MPLS EXP field allows the service provider to provide QoS without overwriting the value in the customer IP precedence field. The IP header remains available for customer use, and the IP packet marking is not changed as the packet travels through the MPLS network.

- MPLS uses a 32-bit label field (shim header), which is inserted between Layer 2 and Layer 3 headers (frame mode).
- MPLS EXP bits supports up to eight CoSs.
- By default, Cisco IOS software copies the three most significant bits of the DSCP or the IP precedence of the IP packet to the EXP field.
- EXP bits are preserved throughout the MPLS network.

# DiffServ Model

This topic describes the purpose and function of the Differentiated Services (DiffServ) model.

## DiffServ Model

- **Describes services associated with traffic classes.**
- **Complex traffic classification and conditioning is performed at the network edge, resulting in a per-packet DSCP.**
- **No per-flow state in the core.**
- **The core only performs simple PHBs on traffic aggregates.**
- **The goal of the DiffServ model is scalability.**
- **Wide variety of services and provisioning policies.**
- **Decouple service and application in use.**
- **No application modification.**
- **No hop-by-hop signaling.**
- **Interoperability with non-DiffServ-compliant nodes.**
- **Incremental deployment.**

© 2006 Cisco Systems, Inc. All rights reserved.

QNT v1.0—4-10

The DiffServ architecture is based on a simple model in which traffic entering a network is classified and possibly conditioned at the boundaries of the network. The traffic class is then identified with a DSCP or bit marking in the IP header.

The DSCP values are used to mark packets to select a per-hop behavior (PHB). Within the core of the network, packets are forwarded according to the PHB that is associated with the DSCP. The PHB is defined as an externally observable forwarding behavior applied at a DiffServ-compliant node to a collection of packets with the same DSCP value.

One of the primary principles of DiffServ is that you should mark packets as close to the edge of the network as possible. It is often a difficult and time-consuming task to determine which traffic class a data packet belongs to. 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 CoS to apply to a given traffic flow.

The primary goal of DiffServ is scalability.

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

DiffServ describes services and allows many user-defined services to be used in a DiffServ-enabled network.

Services are defined as QoS requirements and guarantees that are provided to a collection of packets with the same DSCP value. Services are provided to classes. A class can be identified as a single application or multiple applications with similar service needs, or it can be based on source or destination IP addresses.

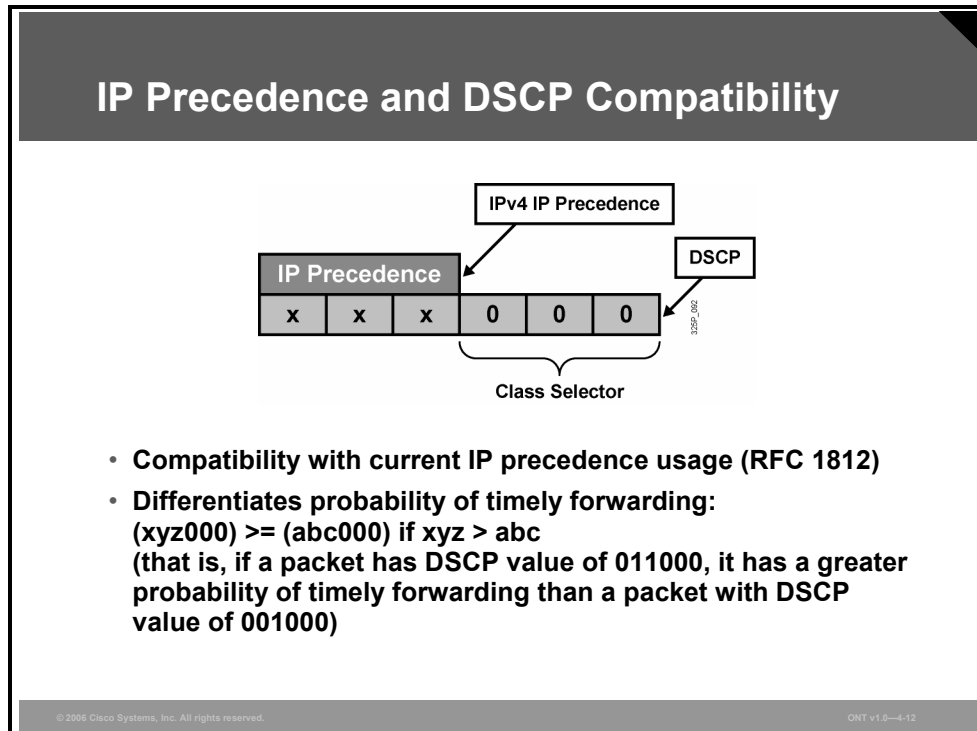
Provisioning is used to allocate resources to defined traffic classes. An example of provisioning would be the set of methods that are used to set up the network configurations on devices to enable the devices to provide the correct set of capabilities for a particular traffic class.

The idea is for the network to recognize a class without having to receive specific requests from applications. This allows the QoS mechanisms to be applied to other applications that do not have Resource Reservation Protocol (RSVP) functionality, which is the case in 99 percent of applications that use IP.



# IP Precedence and DSCP Compatibility

This topic describes the interoperability between DSCP-based and IP-precedence-based devices in a network.



The introduction of DSCP replaces IP precedence, a 3-bit field in the ToS byte of the IP header originally used to classify and prioritize types of traffic. However, DiffServ maintains interoperability with non-DiffServ-compliant devices (those that still use IP precedence). Because of this backward compatibility, DiffServ can be deployed gradually in large networks.

The meaning of the 8 bits in the DiffServ field of the IP packet has changed over time to meet the expanding requirements of IP networks.

Originally, the field was referred to as the ToS field, and the first three bits of the field (bits 7 to 5) defined a packet IP precedence value. A packet could be assigned one of six priorities based on the value of the IP precedence value (eight total values minus two reserved ones). IP precedence 5 (101) was the highest priority that could be assigned (RFC 791).

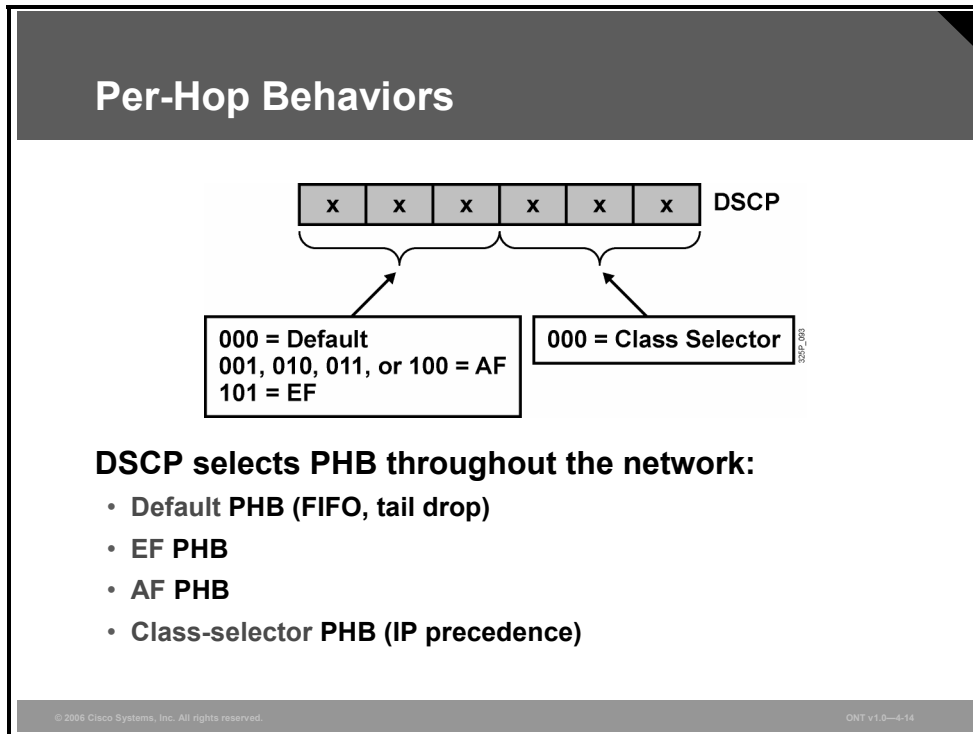
RFC 2474 replaced the ToS field with the DiffServ field, in which a range of eight values (class selector) is used for backward compatibility with IP precedence. There is no compatibility with other bits used by the ToS field.

The class selector PHB was defined to provide backward compatibility for DSCP with ToS-based IP precedence. RFC 1812 simply prioritizes packets according to the precedence value. The PHB is defined as the probability of timely forwarding. Packets with higher IP precedence should be (on average) forwarded in less time than packets with lower IP precedence.

The last 3 bits of the DSCP (bits 2 to 4) set to 0 identify a class-selector PHB.

# Per-Hop Behaviors

This topic describes the PHBs that are used in DSCP.



These PHBs are defined by IETF standards:

- **Default PHB:** Used for best-effort service (bits 5 to 7 of DSCP equal 000)
- **Expedited Forwarding (EF) PHB:** Used for low-delay service (bits 5 to 7 of DSCP equal 101)
- **Assured Forwarding (AF) PHB:** Used for guaranteed bandwidth service (bits 5 to 7 of DSCP equal 001, 010, 011, or 100)
- **Class-selector PHB:** Used for backward compatibility with non-DiffServ-compliant devices (RFC 1812-compliant devices; bits 2 to 4 of DSCP equal 000)

# EF PHB

This subtopic describes the EF PHBs that are used in DSCP.

## EF PHB

• **EF PHB:**

- Ensures a minimum departure rate
- Guarantees bandwidth—class guaranteed an amount of bandwidth with prioritized forwarding
- Polices bandwidth—class not allowed to exceed the guaranteed amount (excess traffic is dropped)

• **DSCP value of 101110:** Looks like IP precedence 5 to non-DiffServ-compliant devices:

- Bits 5 to 7: 101 = 5 (same 3 bits are used for IP precedence)
- Bits 3 and 4: 11 = No drop probability
- Bit 2: Just 0

© 2006 Cisco Systems, Inc. All rights reserved. ORT v1.0—4-15

The EF PHB is identified based on the following:

- **The EF PHB ensures a minimum departure rate:** The EF PHB provides the lowest possible delay to delay-sensitive applications.
- **The EF PHB guarantees bandwidth:** The EF PHB prevents starvation of the application if there are multiple applications using EF PHB.
- **The EF PHB polices bandwidth when congestion occurs:** The EF PHB prevents starvation of other applications or classes that are not using this PHB.

Packets requiring EF should be marked with DSCP binary value 101110 (46 or 0x2E).

Non-DiffServ-compliant devices regard EF DSCP value 101110 as IP precedence 5 (101). This precedence is the highest user-definable IP precedence and is typically used for delay-sensitive traffic (such as VoIP). Bits 5 to 7 of the EF DSCP value are 101, which matches IP precedence 5 and allows backward compatibility.

# AF PHB

This subtopic describes the AF PHBs that are used in DSCP.

## AF PHB

The diagram illustrates the structure of the AF PHB DSCP value. It consists of a 6-bit field divided into three parts: a 3-bit 'Binary Value of the Class' (aaa), a 2-bit 'Drop Probability' (dd), and a 1-bit '0'. The entire field is labeled 'DSCP'.

- **AF PHB:**
  - Guarantees bandwidth
  - Allows access to extra bandwidth, if available
- **Four standard classes: AF1, AF2, AF3, and AF4**
- **DSCP value range of aaadd0:**
  - aaa is a binary value of the class
  - dd is drop probability

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—4-16

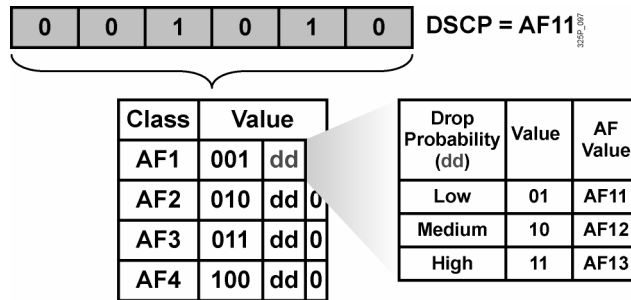
The AF PHB is identified based on the following:

- The AF PHB guarantees a certain amount of bandwidth to an AF class.
- The AF PHB allows access to extra bandwidth, if available.

Packets requiring AF PHB should be marked with DSCP value *aaadd0*, where *aaa* is the number of the class and *dd* is the drop probability.

There are four standard AF classes defined: AF1, AF2, AF3, and AF4. Each class should be treated independently and should have allocated bandwidth that is based on the QoS policy.

## AF PHB (Cont.)



- Each AF class uses three DSCP values.
- Each AF class is independently forwarded with its guaranteed bandwidth.
- Congestion avoidance is used within each class to prevent congestion within the class.

© 2006 Cisco Systems, Inc. All rights reserved.

QNT v1.0—4-18

As illustrated in the figure and the table, there are three DSCP values assigned to each of the four AF classes.

### DSCP Values Assigned to AF Classes

AF Class	Drop Probability	DSCP Value
AF Class 1	AF11 (low)	001 01 0
	AF12 (medium)	001 10 0
	AF13 (high)	001 11 0
AF Class 2	AF21 (low)	010 01 0
	AF22 (medium)	010 10 0
	AF23 (high)	010 11 0
AF Class 3	AF31 (low)	011 01 0
	AF32 (medium)	011 10 0
	AF33 (high)	011 11 0
AF Class 4	AF41 (low)	100 01 0
	AF42 (medium)	100 10 0
	AF43 (high)	100 11 0

# DSCP Summary

This topic describes DSCP settings in the DiffServ model.

PHB		DSCP			Maps to IP Precedence
Default (Best Effort)		0	000000		0
Scavenger (Less-than-Best-Effort)		8	001000		1
Assured Forwarding	Low Drop Pref.	Med Drop Pref.	High Drop Pref.		
Class 1	AF11	AF12	AF13	10 12 14 001010 001100 001110	1
Class 2	AF21	AF22	AF23	18 20 22 010010 010100 010110	2
Class 3	AF31	AF32	AF33	26 28 30 011010 011100 011110	3
Class 4	AF41	AF42	AF43	34 36 38 100010 100100 100110	4
Expedited Forwarding	EF			46 101110	5

A 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 three standard PHBs are:

- Default PHB (as defined in RFC 2474)
- AF PHB (as defined in RFC 2597)
- EF PHB (as defined in RFC 2598) class-selector PHB (as defined in RFC 2474)

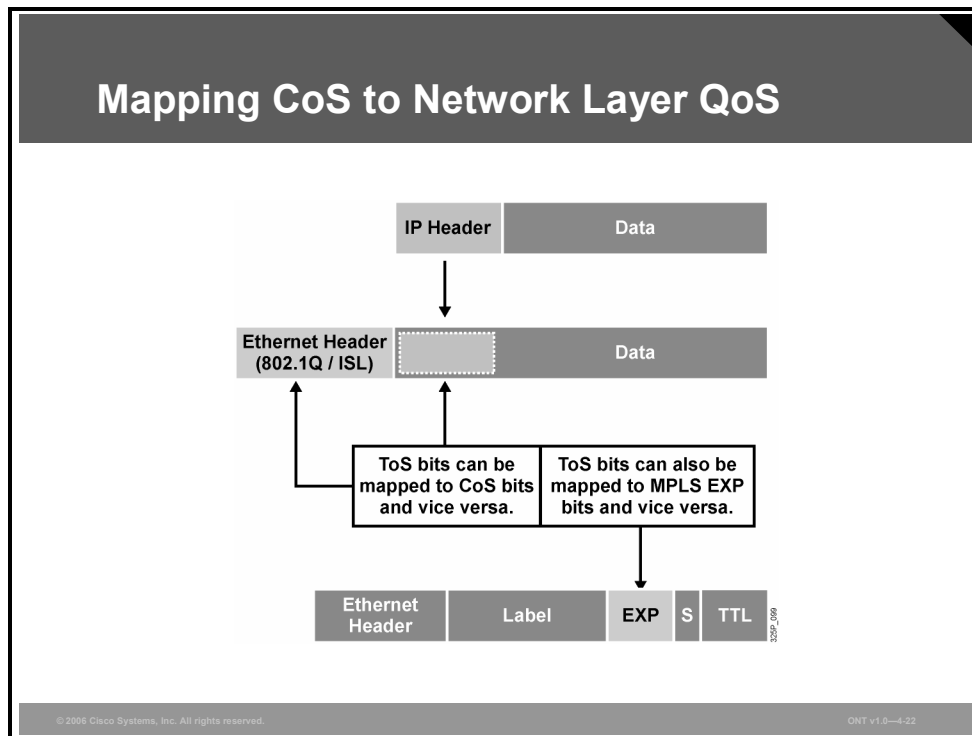
The default PHB specifies that a packet marked with a DSCP value of 000000 (recommended) receives best-effort service from a DiffServ-compliant node. If a packet arrives at a DiffServ-compliant node and the DSCP value is not mapped to any other PHB, the packet is mapped to the default PHB.

The AF PHB defines four AF<sub>n</sub> classes ( $n = 1-4$ : AF1, AF2, AF3, and AF4). Each class is assigned a specific amount of buffer space and interface bandwidth. It is allowed to obtain bandwidth from other AF classes, if bandwidth is available.

The EF PHB defines one class, which assigns a fixed amount of bandwidth only for that class.

# Mapping CoS to Network Layer QoS

This topic describes data link layer-to-network layer interoperability between different QoS markers.



IP headers are preserved end to end when IP packets are transported across a network; data link layer headers are not preserved. This means that the IP layer is the most logical place to mark packets for end-to-end QoS. However, there are edge devices that can mark frames only at the data link layer, and there are many other network devices that operate only at the network layer. To provide true end-to-end QoS, the ability to map QoS markings between the data link layer and the network layer is essential.

Enterprise networks typically consist of a number of remote sites connected to the headquarters campus via a WAN. Remote sites typically consist of a switched LAN, and the headquarters campus network is both routed and switched. Providing end-to-end QoS through such an environment requires that CoS markings that are set at the LAN edge are mapped into QoS markings (such as IP precedence or DSCP) for transit through Campus or WAN routers. Campus and WAN routers can also map the QoS markings to new data-link headers for transit across the LAN. With the mapping, QoS can be preserved and uniformly applied across the enterprise.

Service providers offering IP services have a requirement to provide robust QoS solutions to their customers. The ability to map network layer QoS to link layer CoS allows these service providers to offer a complete end-to-end QoS solution that does not depend on any specific link layer technology.

Compatibility between an MPLS transport layer and network layer QoS is also achieved by mapping between MPLS EXP bits and the IP precedence or DSCP bits. A service provider can map the customer network layer QoS marking as is or change it to fit an agreed-upon SLA. The information in the MPLS EXP bits can be carried end to end in the MPLS network, independent of the transport media. In addition, the network layer marking can remain unchanged so that when the packet leaves the service provider MPLS network, the original QoS markings remain intact. Thus, a service provider with an MPLS network can help provide a true end-to-end QoS solution.

# QoS Service Class Defined

This topic describes the term “QoS service class” and how service classes can be used to create a service policy throughout a network.

## QoS Service Class

- **A QoS service class is a logical grouping of packets that are to receive a similar level of applied quality.**
- **A QoS service class can be:**
  - **A single user (such as, MAC address or IP address)**
  - **A department, customer (such as, subnet or interface)**
  - **An application (such as, port numbers or URL)**
  - **A network destination (such as, tunnel interface or VPN)**

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—4-24

When an administrative policy requiring QoS is created, you must determine how network traffic is to be treated. As part of that policy definition, network traffic must be associated with a specific service class. QoS classification mechanisms are used to separate traffic and identify packets as belonging to a specific service class. QoS marking mechanisms are used to tag each packet as belonging to the assigned service class. After the packets are identified as belonging to a specific service class, QoS mechanisms, such as policing, shaping, and queuing techniques, can be applied to each service class to meet the specifications of the administrative policy. Packets belonging to the same service class are given the same treatment for QoS.

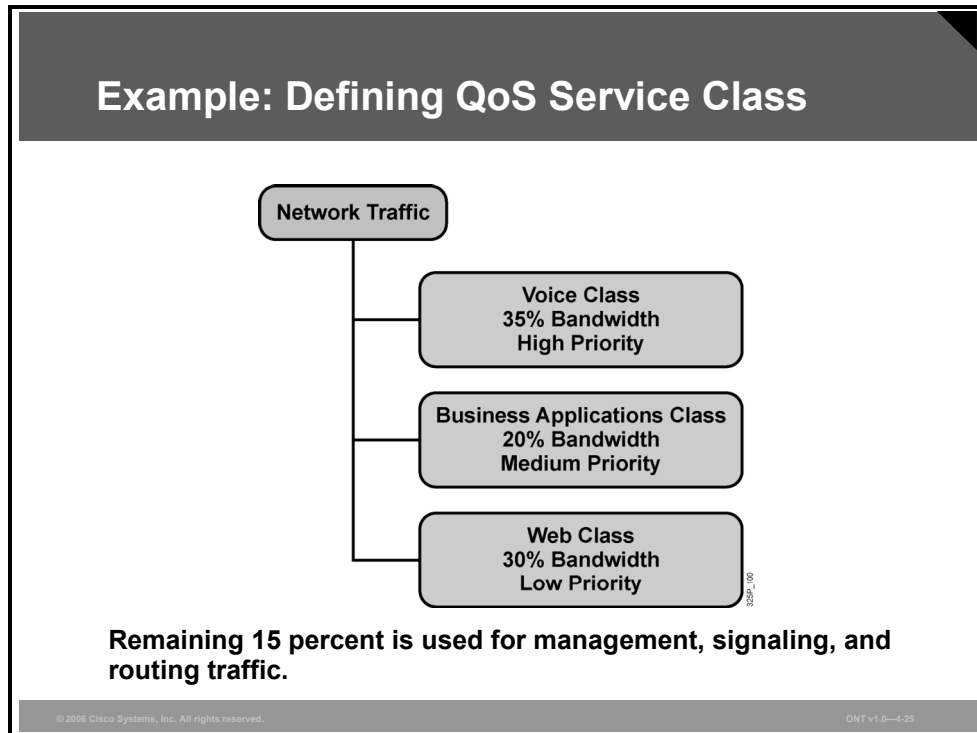
A QoS service class, being a logical grouping, can be defined in many ways:

- Organization or department (for example, marketing, engineering, and sales)
- A specific customer or set of customers
- Specific applications or sets of applications (for example, Telnet, FTP, voice, Session Announcement Protocol [SAP], Oracle, and video)
- Specific users or sets of users (based on MAC address, IP address, and LAN port, for example)
- Specific network destinations (for example, tunnel interfaces and virtual private networks [VPNs])



## Example: Defining QoS Service Class

A network administrator wants to apply QoS to the corporate network to better control bandwidth allocation of different network applications.



Before QoS can be applied, an administrative QoS policy is first defined:

- Voice traffic is to be given a strict priority over all other traffic types and a bandwidth of 35 percent.
- Business applications (FTP, Telnet client TN3270, and Oracle) should be given priority over web traffic and have a guaranteed bandwidth of 20 percent.
- Web traffic should consume no more than 30 percent of any WAN link.

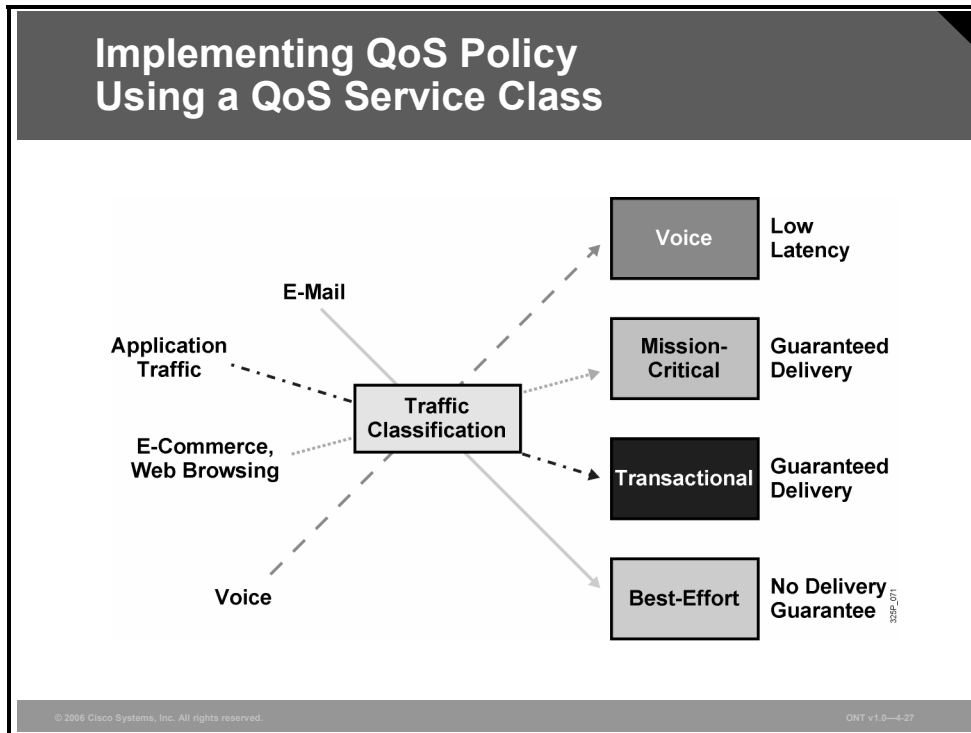
As a result of this policy, three QoS service classes have been defined:

- **Voice class:** Is to be treated with a strict high-priority service.
- **Business applications class:** Requires a guaranteed bandwidth of 20 percent and is to be given priority over web traffic.
- **Web class:** Allowed to consume only up to 30 percent of any WAN link.

The remaining 15 percent is used for management, signaling, and routing.

# Implementing QoS Policy Using a QoS Service Class

This topic describes how link layer and network layer markers are used to define QoS service classes and the applications that can be represented by each of these service classes.



Specifying an administrative policy for QoS requires that a specific set of service classes be defined. QoS mechanisms are uniformly applied to these individual service classes to meet the requirements of the administrative policy. Because the application of QoS mechanisms is applied to different service classes and used to differentiate among applications, users, and traffic, the service class is a key component of a successful QoS implementation.

There are many methods in which service classes can be used to implement an administrative policy. The first step is to identify the traffic in the network and the QoS requirements for each traffic type. Then, traffic can be grouped into a set of service classes for differentiated QoS treatment in the network.

One popular model for the application of QoS service classes is the customer model, a term typically used by service providers when referring to customer traffic. The customer model defines these service classes (although many variations exist):

- **Voice service class:** Delivers low latency for voice services
- **Mission-critical service class:** Guarantees latency and delivery for the transport of mission-critical business applications, such as SAP
- **Transactional service class:** Guarantees delivery and is used for more general applications that are not that sensitive to delay, such as mission-critical applications
- **Best-effort service class:** Used to support small business, e-mail, and other best-effort applications

## Implementing QoS Policy Using a QoS Service Class (Cont.)

- **Profile applications to their basic network requirements.**
- **Do not overengineer provisioning; use no more than four to five traffic classes for data traffic:**
  - **Voice applications: VoIP**
  - **Mission-critical applications: Oracle, SAP, SNA**
  - **Interactive applications: Telnet, TN3270**
  - **Bulk applications: FTP, TFTP**
  - **Best-effort applications: E-mail, WWW**
  - **Scavenger applications: Nonorganizational streaming and video applications**
- **Do not assign more than three applications to mission-critical or transactional classes.**
- **Use proactive policies before reactive (policing) policies.**
- **Seek executive endorsement of relative ranking of application priority prior to rolling out QoS policies for data.**

© 2006 Cisco Systems, Inc. All rights reserved.

QNT v1.0—4-23

One key element of defining QoS service classes is to understand the basic quality needs of network applications. It is essential that applications be given QoS treatment in line with their needs. For example, improperly specifying voice traffic into a service class with guaranteed bandwidth but without a guaranteed low latency (delay) would not meet the needs of the voice traffic.

While it is important to fully understand network application requirements, it is equally important not to overprovision or overdesign the administrative policy. An administrative policy should be proactive in nature and require as few service classes as necessary. One good rule is to limit the number of service classes to no more than four or five. A typical network has these application types:

- Voice applications: VoIP
- Mission-critical applications: Oracle, SAP
- Transactional applications: Telnet
- Best-effort applications: E-mail, web

The QoS requirements of these applications can be met with a few well-designed service classes. The more service classes implemented in support of an administrative QoS policy, the more complex the QoS implementation will be. This complexity also extends to support and troubleshooting as well.

It is also important that the highest-priority classes be reserved for a selected few applications. Marking 90 percent of network traffic as high priority will render most administrative QoS policies useless.

## Example: Application Service Classes

Although there are several sources of information that can be used as guidelines for determining a QoS policy, none of them can determine exactly what is proper for a specific network. Each network presents unique challenges and administrative policies. To properly implement QoS, measurable goals must be declared, and then a plan for achieving these goals must be formulated and implemented.

Example: Application Service Classes				
Application	Layer 3 Classification			Layer 2 CoS
	IPP	PHB	DSCP	
Reserved	7	—	56–62	7
Reserved	6	—	48	6
Voice bearer	5	EF	46	5
Video-data traffic	4	AF41	34	4
Mission-critical data	3	AF31	26	3
Transactional data	2	AF2x	18, 20, 22	2
Scavenger	1	—	8	1
Bulk data	1	AF1x	10, 12, 14	1
Best-effort data	0	BE	0	0
Less-than-best-effort data	0	—	2, 4, 6	0

QoS must be implemented consistently across the entire network. Whether the data is crossing slow WAN links or Gigabit Ethernet, being switched by a Layer 2 switch or routed in a Layer 3 router, the policies must be consistently implemented to satisfy the policy requirements. In this example, a QoS policy is deployed and several classes are defined for the traffic flows. As shown in the table, the traffic flows can be classified by using the IP precedence, PHB, or DSCP according to the importance of the traffic flow. For example, the voice bearer service can be classified with an IP precedence of 5, PHB EF, or a DSCP of 46. Depending on the QoS design of the network, the classification values are mapped into MPLS EXP bit, for example.

If data travels over even a small portion of a network where different policies (or no policies) are applied, the entire QoS policy is destroyed.

The table represents Cisco best practices for marking the traffic with Layer 2 and Layer 3 markers.

### Cisco Best Practices for Marking the Traffic with Layer 2 and Layer 3 Markers

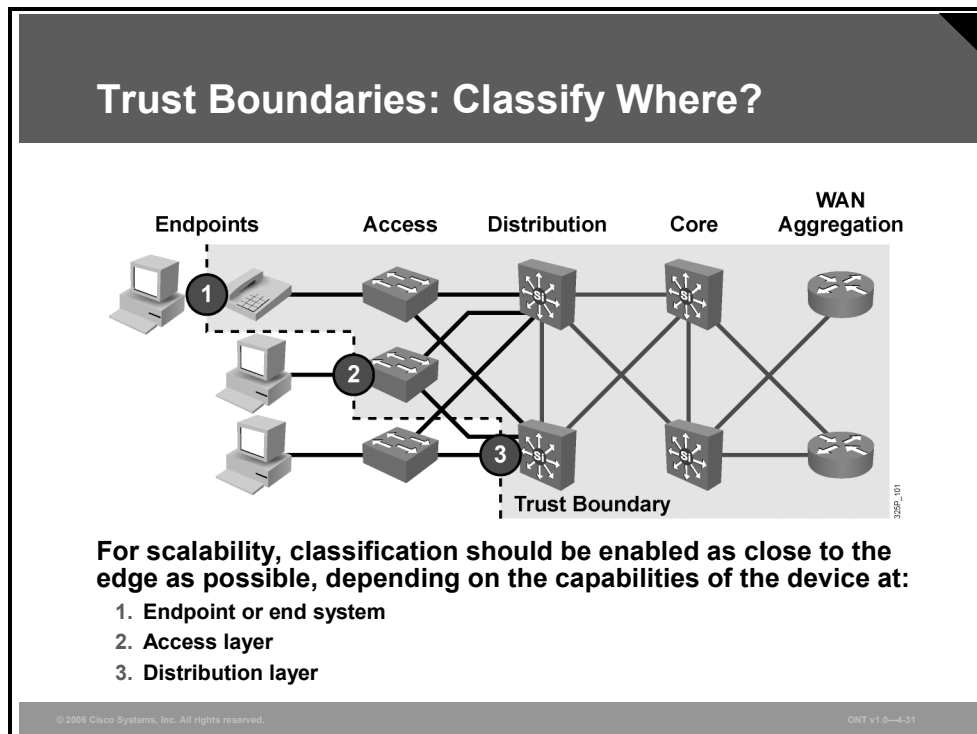
Cisco AutoQoS (10 Classes)	Layer 2 CoS	IPP	Binary DSCP	Decimal DSCP	Code Name	Traffic Type	Per Hop Behavior
Best Effort	0	CS0	000 00 0	0	CS0	Best effort	
			000 01 0	2			
			000 10 0	4			

Cisco AutoQoS (10 Classes)	Layer 2 CoS	IPP	Binary DSCP	Decimal DSCP	Code Name	Traffic Type	Per Hop Behavior
			000 11 0	6			
Scavenger	1	CS1	001 00 0	8	CS1	Scavenger	PHB
Bulk Data	1		001 01 0	10	AF11	Bulk transfers, web, general	
			001 10 0	12	AF12	Bulk transfers, web, general	
			001 11 0	14	AF13	Bulk transfers, web, general	
Network Management	2	CS2	010 00 0	16	CS2	Network management	PHB
Database App/ Transactional	2		010 01 0	18	AF21	Database applications, transactional	
			010 10 0	20	AF22	Database applications, transactional	
			010 11 0	22	AF23	Database applications, transactional	
Telephony Signaling & Control	3	CS3	011 00 0	24	CS3	Telephony signaling	PHB
Local Mission Critical	3	XXX	011 00 1	25	non- standard	Local mission critical	
Telephony Signaling & Control	3		011 01 0	26	AF31	Telephony signaling and control	
			011 10 0	28	AF32	Local mission critical	
			011 11 0	30	AF33	Local mission critical	
Streaming Media Traffic	4	CS4	100 00 0	32	CS4	Streaming video	PHB
Interactive Video- Data Traffic	4		100 01 0	34	AF41	Interactive video and voice	
			100 10 0	36	AF42	Interactive video and voice	
			100 11 0	38	AF43	Interactive video and voice	
		CS5	101 00 0	40	CS5	Voice	PHB
			101 01 0	42			
			101 10 0	44			

Cisco AutoQoS (10 Classes)	Layer 2 CoS	IPP	Binary DSCP	Decimal DSCP	Code Name	Traffic Type	Per Hop Behavior
Interactive Voice Bearer Traffic	5		101 11 0	46	EF	Interactive voice	
IP Routing	6	CS6	110 00 0	48	CS6	IP routing	
			110 01 0	50			
			110 10 0	52			
			110 11 0	54			
		CS7	111 00 0	56		Reserved	PHB
			111 01 0	58		Reserved	
			111 10 0	60		Reserved	
			111 11 0	62		Reserved	

# Trust Boundaries

This topic describes the concept of trust boundaries and how they are used with classification and marking.



A trust boundary is the point within the network where markings such as CoS or DSCP begin to be accepted. Previously set markings are overridden as required at the trust boundary.

The location of the trust boundary depends upon the capabilities of the devices connected to the access edge of the LAN. The trust boundary must be implemented at one of three locations in a network:

- Endpoint or end system
- Access layer
- Distribution layer

Trusted endpoints have the capabilities and intelligence to mark application traffic to the appropriate CoS and/or DSCP values. Trusted endpoints also have the ability to re-mark traffic that may have been previously marked by an untrusted device. Examples of trusted endpoints are these:

- Videoconferencing gateways and systems (Cisco IP/VC 3511 Multipoint Control Unit, 3521 BRI Videoconferencing Gateway, 3526 PRI Videoconferencing Gateway, and 3540 Enhanced Media Processor videoconferencing gateways and systems)
- IP conferencing stations (Cisco IP Conference Station 7935 and Unified IP Conference Station 7936)
- Wireless access points (Cisco Aironet 350, 1100 and 1200 Series Access Points)

If an endpoint is trusted, then the trust boundary should be at the endpoint. When trusted endpoints are connected to a switch port, all that is typically required is enabling the **mls qos trust dscp** interface command.

If the endpoint is not trusted and the switch in the wiring closet has QoS intelligence, then the trust boundary should be at the access layer—within the switch in the wiring closet.

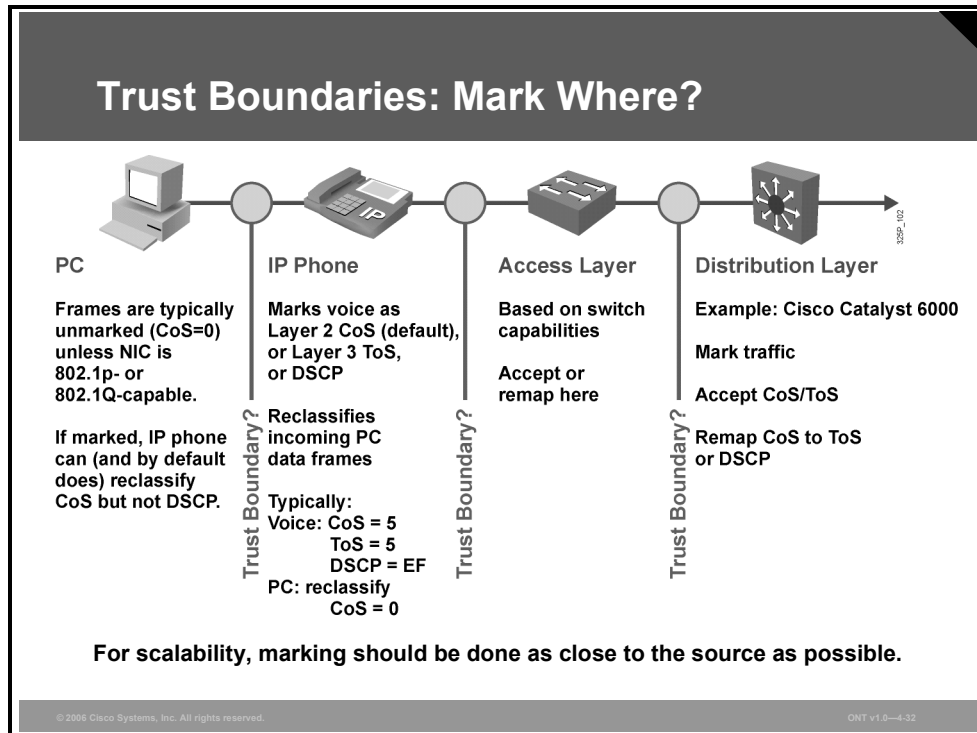
If the endpoint is not trusted and the switch in the wiring closet does not have QoS intelligence, then the trust boundary should be at the distribution layer—within the switch or router that is aggregating traffic from the access layer.

The concept of trusting or not trusting forms the basis for the trust boundary. Ideally, classification should be done as close to the source as possible.



# Trust Boundaries: IP Phones and PCs

Classification should take place at the network edge, typically in the wiring closet or within trusted endpoints (such as servers, trusted hosts, video endpoints, or IP telephony devices).



Trusting end users and their PCs is generally not recommended because newer operating systems like Windows XP and Linux make it relatively easy to set CoS or DSCP markings on PC network interface cards (NICs). Improperly set QoS markings can affect the service levels of users within an enterprise.

One of the main business advantages of IP telephony is the simplicity and related cost savings of adding, moving, or changing a user. To move, a user simply picks up the IP phone, plugs it in at his or her new location, and carries on business as usual. If the infrastructure supports inline power, it is literally a matter of unplugging a single RJ-45 cable and plugging it in at the new location.

IP phones are trusted devices, while PCs are not. This can be a problem when provisioning trust in a mobile environment. For example, port A is configured to trust the endpoint connected to it, which initially is an IP phone. Port B is configured not to trust the endpoint connected to it, which initially is a PC. Because of a move, these endpoints get plugged into the opposite ports. This change breaks the VoIP quality of calls made from the IP phone (now plugged into untrusted port B) and opens the network to unintentional or deliberate abuse of provisioned QoS by the PC (now plugged into the trusted port A).

Cisco switches with QoS intelligence use Cisco Discovery Protocol (CDP) to discover whether any devices plugged into its ports can be trusted. If the device can be trusted (it is a Cisco IP phone), the switch extends trust to the device dynamically. If CDP determines that the device cannot be trusted (it is a PC), the switch does not extend the trust boundary to the device.

The sequence is the following:

1. Switch and IP phone exchange CDP; trust boundary is extended to the IP phone.
2. IP phone sets CoS to 5 for VoIP and to 3 for call signaling traffic.
3. IP phone rewrites CoS from PC to 0.
4. Switch trusts CoS from IP phone and maps CoS to DSCP for output queuing.

It is generally recommended that end-user PC traffic not be trusted. However, some PCs may be running critical applications that require QoS treatment. A classic example is a PC running Cisco IP Communicator. In such a case, the critical application needs to be identified using access control lists (ACLs) and marked or remarked at the access edge. If the access layer switch is incapable of marking or re-marking the traffic, then marking or re-marking needs to take place at the distribution layer switch or router.

# Summary

This topic summarizes the key points that were discussed in this lesson.

## Summary

- **Packet classification** is a QoS mechanism responsible for distinguishing among traffic streams.
- **Packet marking** is a QoS mechanism that “identifies” a packet so it can be distinguished from other packets during the application of QoS.
- Packets can be classified and marked at the data link layer using many different mechanisms, including 802.1Q, ISL, and MPLS EXP bits.
- The DiffServ model describes services associated with traffic classes.
- Complex traffic classification and conditioning is performed at the network edge, resulting in a per-packet DSCP.
- A PHB is an externally observable forwarding behavior applied at a DiffServ-compliant node to a DiffServ BA.
- The EF PHB guarantees and polices bandwidth while ensuring a minimum departure rate.

© 2006 Cisco Systems, Inc. All rights reserved.

QNT v1.0—4-33

## Summary (Cont.)

- The AF PHB guarantees bandwidth while providing four classes, each having three DSCP values.
- The DSCP is backward-compatible with IP precedence and class-selector code point.
- The ability to map network layer QoS to link layer CoS allows service providers to offer a complete end-to-end QoS solution that does not depend on any specific link layer technology.
- A QoS service class is a logical grouping of packets that are to receive a similar level of applied quality, as defined in an administrative policy.
- An administrative policy for QoS requires that a specific set of service classes be defined. QoS mechanisms are uniformly applied to these individual service classes to meet the requirements of the administrative policy.
- It is important that a trust boundary be specified, allowing classification and marking as close to the source as possible.

© 2006 Cisco Systems, Inc. All rights reserved.

QNT v1.0—4-34



# Using NBAR for Classification

---

## Overview

Network-Based Application Recognition (NBAR), a feature in Cisco IOS software, provides intelligent network classification for the infrastructure. NBAR is a classification engine that can recognize a wide variety of applications, including web-based applications and client and server applications that dynamically assign TCP or User Datagram Protocol (UDP) port numbers. After the application is recognized, the network can invoke specific services for that particular application. NBAR currently works with quality of service (QoS) features to ensure that the network bandwidth is best used to fulfill company objectives. These features include the ability to guarantee bandwidth to critical applications, limit bandwidth to other applications, drop selected packets to avoid congestion, and mark packets appropriately so that the network and the service provider network can provide QoS from end to end.

This lesson describes NBAR, a Cisco IOS protocol discovery and classification mechanism. NBAR features covered in this lesson include applications that NBAR can support, Packet Description Language Modules (PDLs), and NBAR Protocol Discovery.

## Objectives

Upon completing this lesson, you will be able to explain Cisco MQC class-based classification and marking operations and configuration using NBAR. This ability includes being able to meet these objectives:

- Describe the Cisco IOS protocol discovery and classification mechanism known as NBAR
- Identify the types of applications supported by NBAR
- Explain the purpose of PDLs in NBAR
- Describe NBAR Protocol Discovery
- Identify the Cisco IOS commands required to configure and monitor NBAR Protocol Discovery
- Identify the Cisco IOS commands required to configure NBAR to recognize static port protocols
- Identify the Cisco IOS commands required to configure NBAR to recognize TCP and UDP stateful protocols

# Network-Based Application Recognition

This topic describes NBAR, a Cisco IOS protocol discovery and classification mechanism.

## Network-Based Application Recognition

- **NBAR classifies modern client-server and web-based applications.**
- **NBAR functions:**
  - **Performs identification of applications and protocols (Layer 4–7)**
  - **Performs protocol discovery**
  - **Provides traffic statistics**
- **NBAR enables downstream actions based on QoS policies via (RED), class-based queuing, and policing.**
- **New applications are easily supported by loading a PDLM.**

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—4.3

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

Classification of HTTP, by URL, or by Multipurpose Internet Mail Extensions (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.

The NBAR Protocol Discovery feature provides an easy way to discover application protocols that are transiting an interface. The feature discovers any protocol traffic supported by NBAR. NBAR Protocol Discovery can be applied to interfaces and can be used to monitor both input and output traffic. It maintains the following per-protocol statistics for enabled interfaces:

- Total number of input and output packets and bytes
- Input and output bit rates

An external Packet Description Language Module (PDLM) can be loaded at run time to extend the NBAR list of recognized protocols. PDLMs can also 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.

NBAR is not supported on these logical interfaces:

- Fast EtherChannel
- Interfaces configured to use tunneling or encryption

NBAR does not support the following:

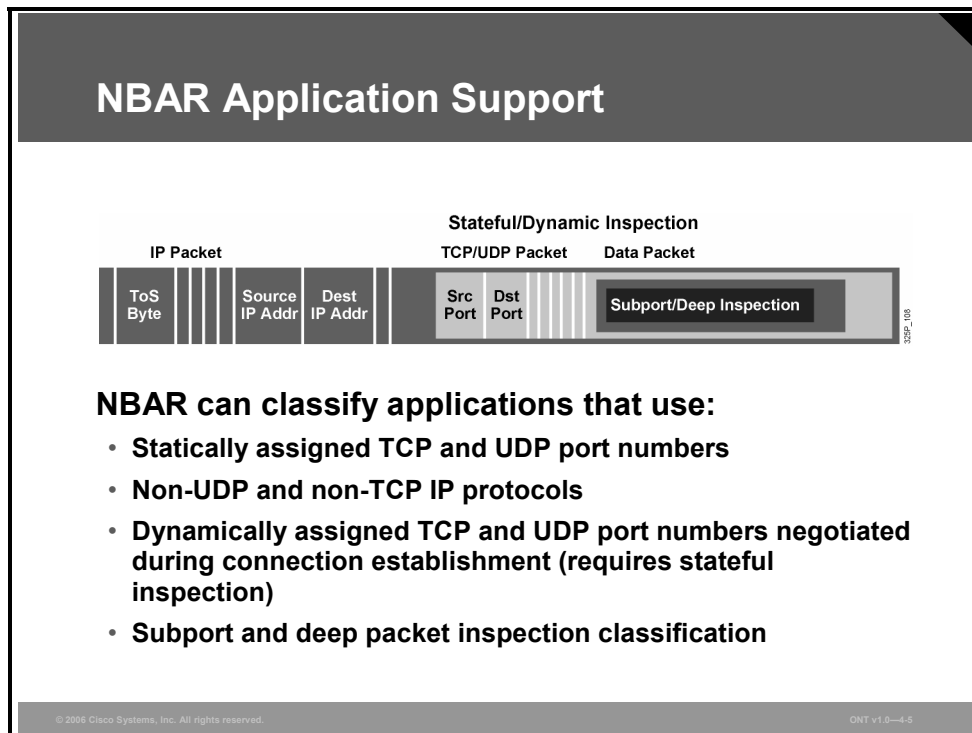
- More than 24 concurrent URLs, hosts, or MIME-type matches
- Matching beyond the first 400 bytes in a packet payload
- Multicast and switching modes other than Cisco Express Forwarding (CEF)
- Fragmented packets
- URL, host, or MIME classification with secure HTTP
- Packets originating from or destined to the router running NBAR

NBAR cannot be used to classify output traffic on a WAN link where tunneling or encryption is used. Therefore, NBAR should be configured on other interfaces on the router (such as a LAN link) to perform input classification before the traffic is switched to the WAN link for output. However, NBAR Protocol Discovery is supported on interfaces where tunneling or encryption is used. You can enable NBAR Protocol Discovery directly on the tunnel or on the interface where encryption is performed to gather key statistics on the various applications that are traversing the interface. The input statistics also show the total number of encrypted or tunneled packets received in addition to the per-protocol breakdowns.

NBAR introduces powerful application classification features into the network at a small-to-medium CPU overhead cost. The CPU utilization will vary based on factors such as the router processor speed and type and the traffic rate.

# NBAR Application Support

This topic describes the types of applications supported by NBAR.



NBAR supports simpler configuration than Modular QoS CLI (MQC) configuration, where classifications and protocol-related details need to be manually configured. NBAR supports stateful recognition of flows. The simpler configuration means that a protocol analyzer capture does not need to be examined to calculate ports and details. Stateful recognition means smarter, deeper packet recognition.

NBAR can be used to recognize packets belonging to different types of applications:

- Static applications establish sessions to well-known TCP or UDP destination port numbers. Such applications, such as Simple Mail Transfer Protocol (SMTP), could also be classified by using access control lists (ACLs).
- Dynamic applications use multiple sessions that use dynamic TCP or UDP port numbers. Typically, there is a control session to a well-known port number, and the other sessions are established to destination port numbers negotiated through the control sessions, such as used in FTP. NBAR inspects the port number exchange through the control session.
- Some non-IP protocols, such as Novell Internetwork Packet Exchange (IPX), can also be recognized by NBAR.
- NBAR also has the capability to inspect some applications for other information and to classify based on that information. For example, NBAR can classify HTTP sessions based on the requested URL, including MIME type or host name.



## NBAR Application Support (Cont.)

TCP and UDP Static Port Protocols				
BGP	IMAP	NNTP	RSVP	SNMP
BOOTP	IRC	Notes	SFTP	SOCKS
CU-SeeMe	Kerberos	Novadigm	SHTTP	SQL Server
DHCP/DNS	L2TP	NTP	SIMAP	SSH
Finger	LDAP	PCAnywhere	SIRC	STELNET
Gopher	MS-PPTP	POP3	SLDAP	Syslog
HTTP	NetBIOS	Printer	SMTP	Telnet
HTTPS	NFS	RIP	SNMP	X Window

TCP and UDP Stateful Protocols			
Citrix ICA	Gnutella	R-commands	StreamWorks
Exchange	HTTP	RealAudio	Sun RPC
FastTrack	Napster	RTP	TFTP
FTP	Netshow	SQL*NET	VDOLive

Non-UDP and Non-TCP Protocols	
EGP	ICMP
EIGRP	IPINIP
GRE	IPsec

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—4-6

Although ACLs can also be used to classify applications based on static port numbers, NBAR is easier to configure and can provide classification statistics.

The table contains the static IP protocols supported by NBAR.

### Static TCP and UDP NBAR Supported Protocols

Protocol	Network Protocol	Protocol ID	Description
BGP	TCP/UDP	179	Border Gateway Protocol
CU-SeeMe	TCP/UDP	7648, 7649	Desktop videoconferencing
CU-SeeMe	UDP	24032	Desktop videoconferencing
DHCP/ BOOTP	UDP	67, 68	DHCP, Bootstrap Protocol
DNS	TCP/UDP	53	Domain Name System
Finger	TCP	79	Finger user information protocol
Gopher	TCP/UDP	70	Internet Gopher protocol
HTTP	TCP	80	HTTP
HTTPS	TCP	443	Secure HTTP
IMAP	TCP/UDP	143, 220	Internet Message Access Protocol
IRC	TCP/UDP	194	Internet Relay Chat
Kerberos	TCP/UDP	88, 749	Kerberos Network Authentication Service
L2TP	UDP	1701	Layer 2 Tunneling Protocol
LDAP	TCP/UDP	389	Lightweight Directory Access Protocol
MS-PPTP	TCP	1723	Microsoft Point-to-Point Tunneling Protocol for VPN

Protocol	Network Protocol	Protocol ID	Description
MS-SQLServer	TCP	1433	Microsoft SQL Server Desktop Videoconferencing
NetBIOS	TCP	137, 139	NetBIOS over IP (Microsoft Windows)
NetBIOS	UDP	137, 138	NetBIOS over IP (Microsoft Windows)
NFS	TCP/UDP	2049	Network File System
NNTP	TCP/UDP	119	Network News Transfer Protocol
Notes	TCP/UDP	1352	Lotus Notes
Novadigm	TCP/UDP	3460–3465	Novadigm Enterprise Desktop Manager (EDM)
NTP	TCP/UDP	123	Network Time Protocol
PCAnywhere	TCP	5631, 65301	Symantec PCAnywhere
PCAnywhere	UDP	22, 5632	Symantec PCAnywhere
POP3	TCP/UDP	110	Post Office Protocol
Printer	TCP/UDP	515	Printer
RIP	UDP	520	Routing Information Protocol
RSVP	UDP	1698,17	Resource Reservation Protocol
SFTP	TCP	990	Secure FTP
SHTTP	TCP	443	Secure HTTP (see also HTTPS)
SIMAP	TCP/UDP	585, 993	Secure IMAP
SIRC	TCP/UDP	994	Secure IRC
SLDAP	TCP/UDP	636	Secure LDAP
SNNT	TCP/UDP	563	Secure NNTP
SMTP	TCP	25	Simple Mail Transfer Protocol
SNMP	TCP/UDP	161, 162	Simple Network Management Protocol
SOCKS	TCP	1080	Firewall security protocol
SPOP3	TCP/UDP	995	Secure POP3
SSH	TCP	22	Secured Shell Protocol
STELNET	TCP	992	Secure Telnet
Syslog	UDP	514	System logging utility
Telnet	TCP	23	Telnet protocol
X Window	TCP	6000-6003	X11, X Window

The table lists the non-TCP and non-UDP protocols supported by NBAR.

### Non-TCP and Non-UDP NBAR Supported Protocols

Protocol	Network Protocol	Protocol ID	Description
EGP	IP	8	Exterior Gateway Protocol
GRE	IP	47	Generic Routing Encapsulation
ICMP	IP	1	Internet Control Message Protocol
IPIP	IP	4	IP in IP
IPsec	IP	50, 51	IP Encapsulating Security Payload (ESP = 50) and Authentication Header (AH = 51)
EIGRP	IP	88	Enhanced Interior Gateway Routing Protocol

The table lists the dynamic (or stateful) protocols supported by NBAR.

### Stateful NBAR Supported Protocols

Stateful Protocol	Transport Protocol	Description
FTP	TCP	File Transfer Protocol
Exchange	TCP	MS-RPC for Microsoft Exchange
HTTP	TCP	HTTP with URL, MIME, or host classification
NetShow	TCP/UDP	Microsoft NetShow
RealAudio	TCP/UDP	RealAudio streaming protocol
r-commands	TCP	rsh, rlogin, rexec
StreamWorks	UDP	Xing Technology StreamWorks audio and video
SQL*NET	TCP/UDP	SQL*NET for Oracle
SunRPC	TCP/UDP	Sun Remote Procedure Call
TFTP	UDP	Trivial File Transfer Protocol
VDOLive	TCP/UDP	VDOLive streaming video

# Packet Description Language Module

This topic describes the purpose of PDLMs in NBAR.

## Packet Description Language Module

- PDLMs allow NBAR to recognize new protocols matching text patterns in data packets without requiring a new Cisco IOS software image or a router reload.
- An external PDLM can be loaded at run time to extend the NBAR list of recognized protocols.
- PDLMs can also be used to enhance an existing protocol recognition capability.
- PDLMs must be produced by Cisco engineers.

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—4.8

New features are usually added to new versions of Cisco IOS software. NBAR is the first mechanism that supports dynamic upgrades without having to change the Cisco IOS version or restart a router.

PDLMs contain the rules that are used by NBAR to recognize an application by matching text patterns in data packets, and they can be used to bring new or changed functionality to NBAR.

An external 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.

---

<b>Note</b>	New PDLMs are released only by Cisco Systems and are available from local Cisco representatives. The PDLMs can be loaded from flash memory. Registered users can find the PDLMs at <a href="http://www.cisco.com/cgi-bin/tablebuild.pl/pdlm">http://www.cisco.com/cgi-bin/tablebuild.pl/pdlm</a> .
-------------	--

---

## Packet Description Language Module (Cont.)

```
router(config)#
```

```
ip nbar pdlm pdlm-name
```

- Used to enhance the list of protocols recognized by NBAR through a PDLM.
- The filename is in the URL format (for example, flash://citrix.pdlm).

```
router(config)#
```

```
ip nbar port-map protocol-name [tcp | udp] port-number
```

- Configures NBAR to search for a protocol or protocol name using a port number other than the well-known port.
- Up to 16 additional port numbers can be specified.

© 2006 Cisco Systems, Inc. All rights reserved.ONT v1.0—4-8

To extend or enhance the list of protocols recognized by NBAR through a PDLM provided by Cisco, use the **ip nbar pdlm** global configuration command. The *pdlm-file* parameter should be in the URL format and can point to the flash where Cisco IOS software is stored (for example, flash://citrix.pdlm). The file can also be located on a TFTP server (for example, tftp://10.1.1.1/nbar.pdlm).

Use the **no** form of this command to unload a PDLM if it was previously loaded.

**ip nbar pdlm** *pdlm-name*

### ip nbar pdlm Parameter

Parameter	Description
<i>pdlm-name</i>	The URL where the PDLM can be found on the flash card

To configure NBAR to search for a protocol or protocol name using a port number other than the well-known port, use the **ip nbar port-map** global configuration command.

**ip nbar port-map** *protocol-name* [tcp | udp] *port-number*

## ip nbar port-map Parameter

Parameter	Description
<i>protocol-name</i>	Name of protocol known to NBAR.
<b>tcp</b>	(Optional) Specifies that a TCP port will be searched for the specified <i>protocol-name</i> argument.
<b>udp</b>	(Optional) Specifies that a UDP port will be searched for the specified <i>protocol-name</i> argument.
<i>port-number</i>	Assigned port for named protocol. The <i>port-number</i> argument is either a UDP or a TCP port number, depending on which protocol is specified in this command line. Up to 16 <i>port-number</i> arguments can be specified in one command line. Port number values can range from 0 to 65,535.

### Packet Description Language Module (Cont.)

router#

```
show ip nbar port-map [protocol-name]
```

- Displays the current NBAR protocol-to-port mappings

```
router#show ip nbar port-map
```

```
port-map bgp udp 179
port-map bgp tcp 179
port-map cuseeme udp 7648 7649
port-map cuseeme tcp 7648 7649
port-map dhcp udp 67 68
port-map dhcp tcp 67 68
port-map dns udp 53
port-map dns tcp 53
```

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—4-10

Use the **show ip nbar port-map** command to display the current protocol-to-port mappings in use by NBAR. The *protocol-name* argument can also be used to limit the display to a specific protocol.

After the **ip nbar port-map** command has been used, the **show ip nbar port-map** command displays the ports assigned by the administrator to the protocol. If no **ip nbar port-map** command has been used, the **show ip nbar port-map** command displays the default ports.

# Protocol Discovery

This topic describes NBAR Protocol Discovery.

## NBAR Protocol Discovery

- **Analyzes application traffic patterns in real time and discovers which traffic is running on the network**
- **Provides bidirectional, per-interface, and per-protocol statistics**
- **Important monitoring tool supported by Cisco QoS management tools:**
  - **Generates real-time application statistics**
  - **Provides traffic distribution information at key network locations**

© 2006 Cisco Systems, Inc. All rights reserved. OMT v1.0—4-12

To develop and apply QoS policies, NBAR includes a protocol-discovery feature that provides an easy way to discover application protocols that are transiting an interface. The feature discovers any protocol traffic that is supported by NBAR.

NBAR Protocol Discovery captures key statistics associated with each protocol in a network. These statistics can be used to define traffic classes and QoS policies for each traffic class.

NBAR Protocol Discovery can be applied to interfaces and can be used to monitor both input and output traffic. In addition, it shows the mix of applications currently running on the network. This information helps in defining QoS classes and policies, such as how much bandwidth to provide to mission-critical applications, and in determining which protocols should be policed.

# Configuring and Monitoring NBAR Protocol Discovery

This topic describes the Cisco IOS commands that are required to configure and monitor NBAR Protocol Discovery.

## Configuring and Monitoring Protocol Discovery

```
router(config-if)#  
ip nbar protocol-discovery
```

- Configures NBAR to discover traffic for all protocols known to NBAR on a particular interface
- Requires that CEF be enabled before protocol discovery
- Can be applied with or without a service policy enabled

```
router#  
show ip nbar protocol-discovery
```

- Displays the statistics for all interfaces on which protocol discovery is enabled

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—4-14

The NBAR feature has two components:

- One component monitors applications traversing a network.
- The other component classifies traffic by protocol.

To monitor applications traversing a network, NBAR Protocol Discovery must be enabled. The ability to classify traffic by protocol using NBAR and then to apply QoS to the classified traffic is configured using the MQC.

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

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 this feature is currently enabled. The default output of this command includes—in this order—input bit rate (bits per second), input byte count, input packet count, and protocol name. Output statistics include packet count, byte count, and the output bit rate in bits per second.



## Configuring and Monitoring Protocol Discovery (Cont.)

```
router#show ip nbar protocol-discovery

Ethernet0/0

  Protocol      Input          Output
                Packet Count   Packet Count
                Byte Count    Byte Count
                5 minute bit rate (bps)  5 minute bit rate (bps)
  -----
  realaudio     2911           3040
                1678304       198406
                19000         1000
  http          19624          13506
                14050949       2017293
                0           0
<...rest of the output omitted...>
```

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0-4.15

NBAR 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 (policing at the output interface, ACLs, or queue drops). The example displays partial output of the **show ip nbar protocol-discovery** command for an Ethernet interface.

# Configuring NBAR for Static Protocols

This topic describes the Cisco IOS commands that are required to configure NBAR to recognize static port protocols.

## Configuring NBAR for Static Protocols

### Required steps:

1. **Enable NBAR Protocol Discovery.**
2. **Configure a traffic class.**
3. **Configure a traffic policy.**
4. **Attach the traffic policy to an interface.**
5. **Enable PDLM if needed.**

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—4-17

The ability of NBAR to classify traffic by protocol and then apply QoS to that traffic uses the MQC class map match criteria. The following steps are required to successfully deploy NBAR for static protocols:

- Step 1**     Enable NBAR Protocol Discovery.
- Step 2**     Configure a traffic class.
- Step 3**     Configure a traffic policy.
- Step 4**     Attach the traffic policy to an interface.
- Step 5**     Enable PDLM if needed.

## Configuring NBAR for Static Protocols (Cont.)

```
router(config-cmap)#  
match protocol protocol
```

- Configures the match criteria for a class map on the basis of the specified protocol using the MQC configuration mode.
- Static protocols are recognized based on the well-known destination port number.
- A match not command can be used to specify a QoS policy value that is not used as a match criterion; in this case, all other values of that QoS policy become successful match criteria.

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—4.18

When configuring NBAR, the administrator does not need to understand how a certain protocol works. The configuration simply requires the administrator to enter the name of the protocol (static or stateful).

**match protocol** *protocol-name*

### match protocol Parameter

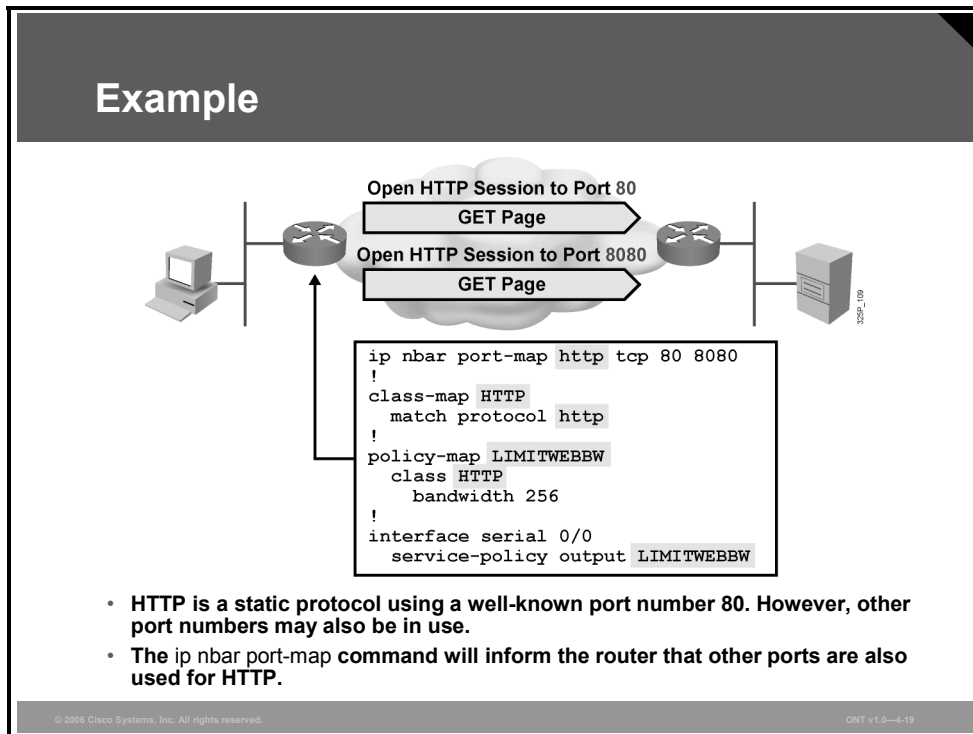
Parameter	Description
<i>protocol-name</i>	Name of the protocol used as a matching criterion. Supported protocols include the following (some protocols have been omitted; refer to Cisco IOS documentation for complete details): <ul style="list-style-type: none"><li>■ <b>aarp</b>—AppleTalk Address Resolution Protocol (ARP)</li><li>■ <b>arp</b>—IP ARP</li><li>■ <b>bridge</b>—bridging</li><li>■ <b>cdp</b>—Cisco Discovery Protocol</li><li>■ <b>compressedtcp</b>—compressed TCP</li><li>■ <b>dlsnw</b>—data-link switching</li><li>■ <b>ip</b>—IP</li><li>■ <b>ipx</b>—Novell IPX</li></ul>

Some protocols (static or stateful) can use additional TCP or UDP ports. Use the **ip nbar port-map** command to extend the NBAR functionality for well-known protocols to new port numbers.

To extend or enhance the list of protocols recognized by NBAR through a Cisco PDLM, use the **ip nbar pdlm** global configuration command.

## Example

HTTP is often used on other port numbers. The example shows the usage of the **ip nbar port-map** command to also enable HTTP recognition on TCP port 8080.



The NBAR port map is configured for HTTP for TCP ports 80 and 8080.

The class map called “HTTP” is used to match the HTTP protocol. The policy map called “LIMITWEBBW” will use the class map HTTP and set the bandwidth for HTTP traffic to 256 kbps.

The policy map is then applied as a service policy for outbound traffic on serial0/0.

# Configuring Stateful NBAR for Dynamic Protocols

This topic describes the Cisco IOS commands that are required to configure NBAR to recognize TCP and UDP stateful protocols.

## Configuring Stateful NBAR for Dynamic Protocols

### Required steps:

1. **Configure a traffic class.**
2. **Configure a traffic policy.**
3. **Attach the traffic policy to an interface.**

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—4-21

The ability to classify traffic by protocol using NBAR and then apply QoS to the classified traffic is configured using the MQC. The following steps are required to deploy NBAR for stateful protocols:

- Step 1**      Configure a traffic class.
- Step 2**      Configure a traffic policy.
- Step 3**      Attach the traffic policy to an interface.

## Configuring Stateful NBAR for Dynamic Protocols (Cont.)

```
router(config-cmap)#
```

```
match protocol http url url-string
```

- Recognizes the HTTP GET packets containing the URL, and then matches all packets that are part of the HTTP GET request
- Include only the portion of the URL following the address or host name in the match statement

```
router(config-cmap)#
```

```
match protocol http host hostname-string
```

- Performs a regular expression match on the host field content inside an HTTP GET packet and classifies all packets from that host

© 2006 Cisco Systems, Inc. All rights reserved.ONT v1.0-4.22

NBAR has enhanced classification capabilities for HTTP. It can classify packets belonging to HTTP flows based on the following:

- The URL portion after the host name, which appears in the GET request of the HTTP session
- The host name specified in the GET request

The following example classifies, within the class map called “class1,” HTTP packets based on any URL containing the string “whatsnew/latest” followed by zero or more characters:

```
class-map class1
  match protocol http url whatsnew/latest*
```

The next example classifies, within the class map called “class2,” packets based on any host name containing the string “cisco” followed by zero or more characters:

```
class-map class2
  match protocol http host cisco*
```

## Configuring Stateful NBAR for Dynamic Protocols (Cont.)

```
router(config-cmap)#
```

```
match protocol http mime MIME-type
```

- Matches a packet containing the MIME type and all subsequent packets until the next HTTP transaction for stateful protocol.

```
router(config-cmap)#
```

```
match protocol fasttrack file-transfer  
regular-expression
```

- Stateful mechanism to identify a group of peer-to-peer file-sharing applications.
- Applications that use FastTrack peer-to-peer protocol include Kazaa, Grokster, Gnutella, and Morpheus.
- A Cisco IOS regular expression is used to identify specific FastTrack traffic.
- To specify that all FastTrack traffic will be identified by the traffic class, use asterisk (\*) as the regular expression.

© 2006 Cisco Systems, Inc. All rights reserved.ONT v1.0--4.23

NBAR supports a wide range of network protocols, including the stateful protocols that were difficult to classify before. Stateful protocols such as HTTP or FastTrack applications need special configuration to use the NBAR feature.

NBAR offers the ability to match packets containing a specified MIME type.

The following example classifies, within the class map called “class3,” packets based on the JPEG MIME type:

```
class-map class3  
  match protocol http mime "*jpeg"
```

Applications that use the FastTrack peer-to-peer protocol include Kazaa, Grokster, and Morpheus (although newer versions of Morpheus use Gnutella).

A regular expression is used to identify specific FastTrack traffic. For instance, entering “cisco” as the regular expression would classify the FastTrack traffic containing the string “cisco” as a match for the traffic policy.

To specify that all FastTrack traffic be identified by the traffic class, use “\*” as the regular expression.

The following example configures NBAR to match all FastTrack traffic:

```
match protocol fasttrack file-transfer "*"
```

In the next example, all FastTrack files that have the .mpeg extension will be classified into class map nbar.

```
class-map match-all nbar  
match protocol fasttrack file-transfer "*.mpeg"
```

The following example configures NBAR to match FastTrack traffic that contains the string “cisco”:

```
match protocol fasttrack file-transfer "*cisco*"
```

## Configuring Stateful NBAR for Dynamic Protocols (Cont.)

```
router(config-cmap)#
```

```
match protocol rtp [audio | video | payload-type  
payload-string]
```

- **Identifies real-time audio and video traffic in the class-map mode of MQC.**
- **Differentiates on the basis of audio and video codecs.**
- **The match protocol rtp command has these options:**
  - **audio: Match by payload type values 0 to 23, reserved for audio traffic.**
  - **video: Match by payload type values 24 to 33, reserved for video traffic.**
  - **payload-type: Match by a specific payload type value; provides more granularity than the audio or video options.**

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0-4.24

Real-Time Transport Protocol (RTP) consists of a data part and a control part. The control part is called Real-Time Transport Control Protocol (RTCP). It is important to note that the NBAR RTP payload classification feature does not identify RTCP packets and that RTCP packets run on odd-numbered ports, while RTP packets run on even-numbered ports.

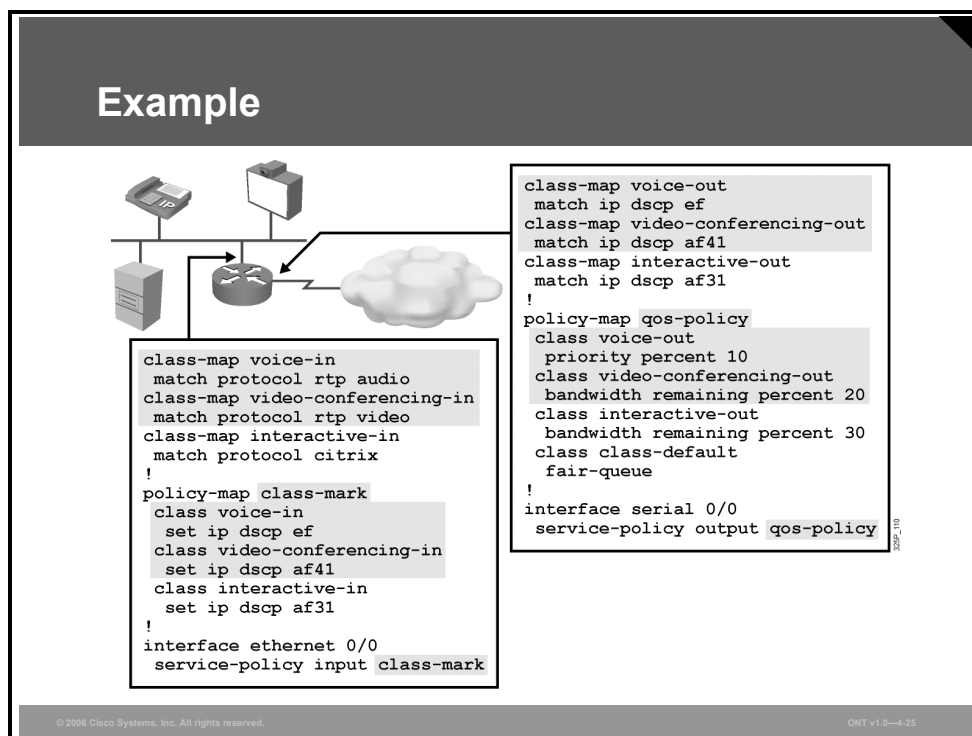
The data part of RTP is a thin protocol providing support for applications with real-time properties (such as continuous media [audio and video]), which includes timing reconstruction, loss detection, and security and content identification. The RTP payload type is the data transported by RTP in a packet (for example, audio samples or compressed video data).

NBAR RTP payload classification not only allows you to statefully identify real-time audio and video traffic, but it also can differentiate on the basis of audio and video codecs to provide more granular QoS. The RTP payload classification feature looks deep into the RTP header to classify RTP packets.



## Example: Classification of RTP Session

This example illustrates a simple classification of RTP sessions, both on the input interface and on the output interface of the router.



On the input interface, three class maps have been created: voice-in, videoconferencing-in, and interactive-in. The voice-in class map will match the RTP audio protocol, the videoconferencing-in class map will match the RTP video protocol, and the interactive-in class map will match the Citrix protocol.

The class-mark policy map will then do the following:

- If the packet matches the voice-in class map, the packet differentiated services code point (DSCP) field will be set to Expedited Forwarding (EF). If the packet matches the videoconferencing-in class map, the packet DSCP field will be set to Assured Forwarding (AF) 41. If the packet matches the interactive-in class map, the DSCP field will be set to AF 31.
- The class-mark policy map is applied to the input interface, Ethernet 0/0.

On the output interface, three class maps have been created: voice-out, videoconferencing-out, and interactive-out. The voice-out class map will match the DSCP field EF. The videoconferencing-out class map will match the DSCP field AF 41. The interactive-out class map will match the DSCP field AF 31.

As shown in the figure, the qos-policy policy map will then do the following:

- If the packet matches the voice-out class map, the packet priority will be set to 10 percent of the bandwidth. If the packet matches the videoconferencing-out class map, the packet priority will be set to 20 percent of the bandwidth. If the packet matches the interactive-out class map, the packet priority will be set to 30 percent of the bandwidth. All other packets will be classified as class-default and fair queuing will be performed on them.
- The class-mark policy map is applied to the output interface, serial 0/0.

# Summary

This topic summarizes the key points that were discussed in this lesson.

## Summary

- **NBAR identifies applications and protocols (Layer 4–7), and provides traffic statistics.**
- **NBAR supports both statically and dynamically assigned TCP and UDP port numbers along with other means to recognize applications.**
- **PDLs contain the rules that are used by NBAR to recognize an application and can be used to bring new or changed functionality to NBAR.**
- **NBAR Protocol Discovery analyzes application traffic patterns in real time and discovers which traffic is running on the network.**

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—4-26

## Summary (Cont.)

- **Use the `ip nbar protocol-discovery` command to configure NBAR to keep traffic statistics for all protocols known to NBAR. Use the `show ip nbar protocol-discovery` command to display statistics gathered by the NBAR Protocol Discovery feature.**
- **Use the `ip nbar port-map` command to extend the NBAR functionality for well-known protocols to new port numbers.**
- **Use the `match protocol` command to allow static protocols to be recognized based on well-known port numbers.**
- **The `match protocol rtp` command allows identification of real-time audio and video traffic.**

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—4-27

## References

For additional information, refer to this resource:

- Cisco Systems, Inc. NBAR Packet Description Language Modules software download at <http://www.cisco.com/cgi-bin/tablebuild.pl/pdlm>.

# Introducing Queuing Implementations

---

## Overview

Queuing algorithms are one of the primary ways to temporarily manage congestion in a network. Network devices handle an overflow of arriving traffic by using a queuing algorithm to sort traffic and determine a method of prioritizing the traffic onto an output link. Each queuing algorithm was designed to solve a specific network traffic problem and has a particular effect on network performance. Network administrators use queuing algorithms to solve certain network traffic congestion problems. This lesson describes the basic queuing algorithms.

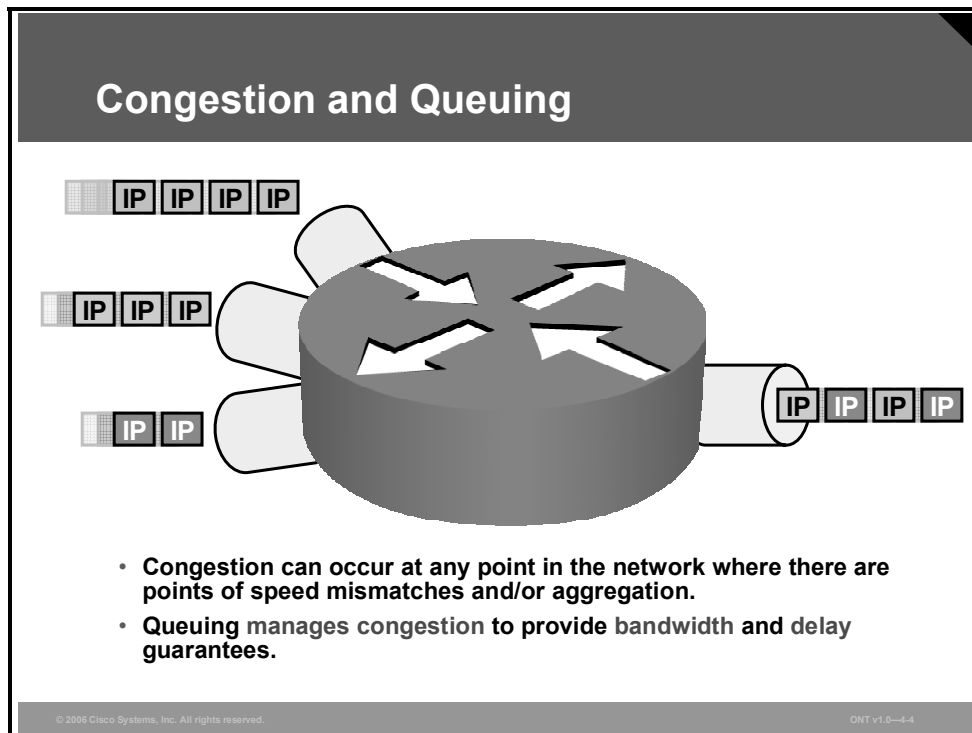
## Objectives

Upon completing this lesson, you will be able to explain Cisco queuing operations and basic configurations. This ability includes being able to meet these objectives:

- Explain the need for congestion-management mechanisms
- List the various queuing algorithms
- Describe the FIFO queuing algorithm
- Describe the PQ algorithm
- Describe the round-robin queuing algorithm and its variants
- Describe the primary components of a queuing mechanism

# Congestion and Queuing

This topic describes the need for congestion-management mechanisms.



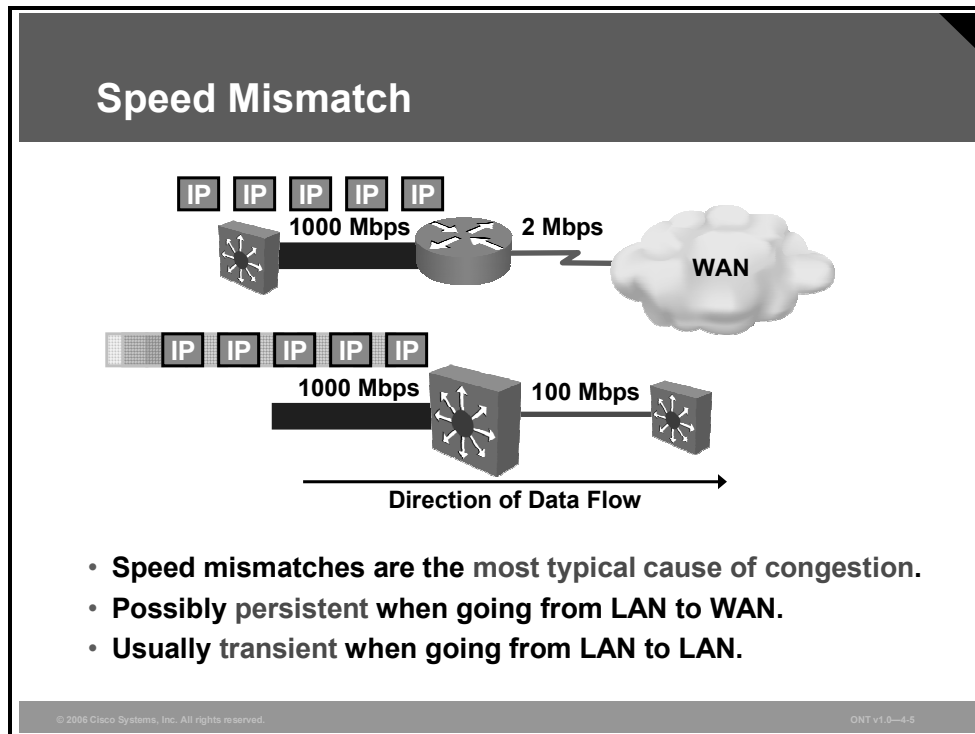
Congestion can occur anywhere within a network where speed mismatches (for example, a 1000-Mbps link feeding a 100-Mbps link), aggregation (for example, multiple 100-Mbps links feeding an upstream 100-Mbps link), or confluence (the joining of two or more traffic streams) accrue.

Congestion-management features control the congestion when it occurs. One way that network elements handle an overflow of arriving traffic is to use a queuing algorithm to sort the traffic and 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.

Many algorithms have been designed to serve different needs. A well-designed queuing algorithm provides some bandwidth and delay guarantees to priority traffic.

## Example: Congestion Caused by Speed Mismatch

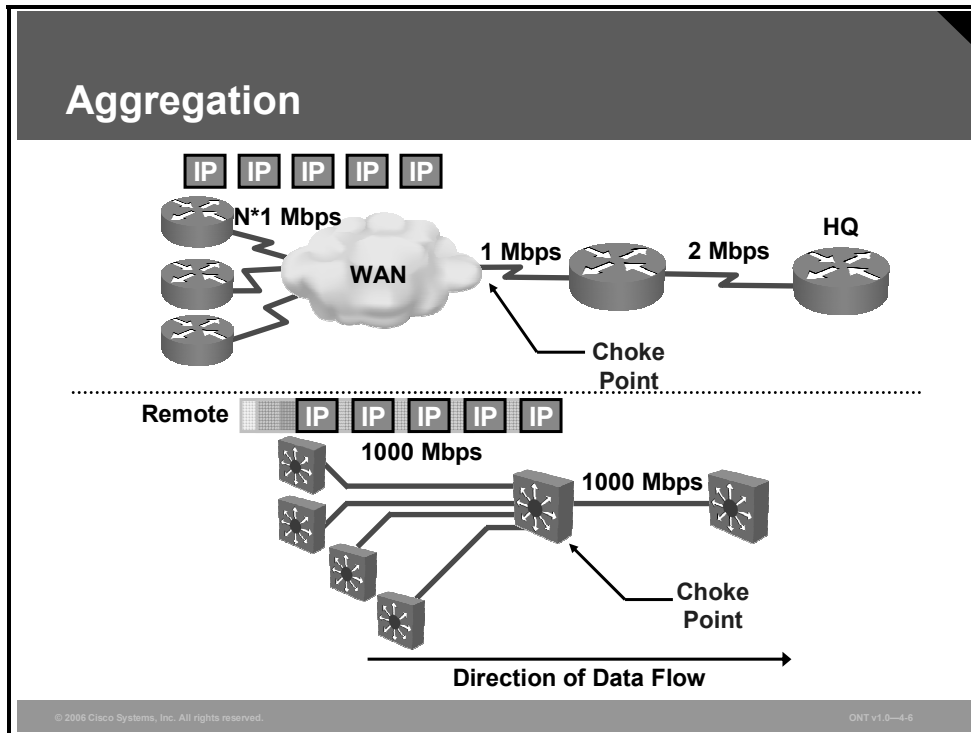
Speed mismatches are the most typical cause of congestion in a network.



Speed mismatches are most common reason for congestion. It is possible to have persistent congestion when traffic is moving from a LAN to a WAN, such as when traffic moves from a high-speed LAN environment (100 or 1000 Mbps) to lower-speed WAN links (1 or 2 Mbps). Speed mismatches are also common in LAN-to-LAN environments when, for example, a 1000-Mbps link feeds into a 100-Mbps link, but in those cases, they are transient.

## Example: Congestion Caused by Aggregation

The second most common source of congestion is points of aggregation in a network.



Typical points of aggregation occur in WANs when multiple remote sites feed into a central site.

In a LAN environment, congestion resulting from aggregation often occurs at the distribution layer of networks where the access layer devices feed traffic to the distribution layer switches.

# Queuing Algorithms

This topic describes queuing algorithms.

## Definition

- **Queuing is designed to accommodate temporary congestion on an interface of a network device by storing excess packets in buffers until bandwidth becomes available.**
- **Queuing is a congestion management mechanism that allows you to control congestion on interfaces.**

© 2006 Cisco Systems, Inc. All rights reserved.

QNT v1.0—4-8

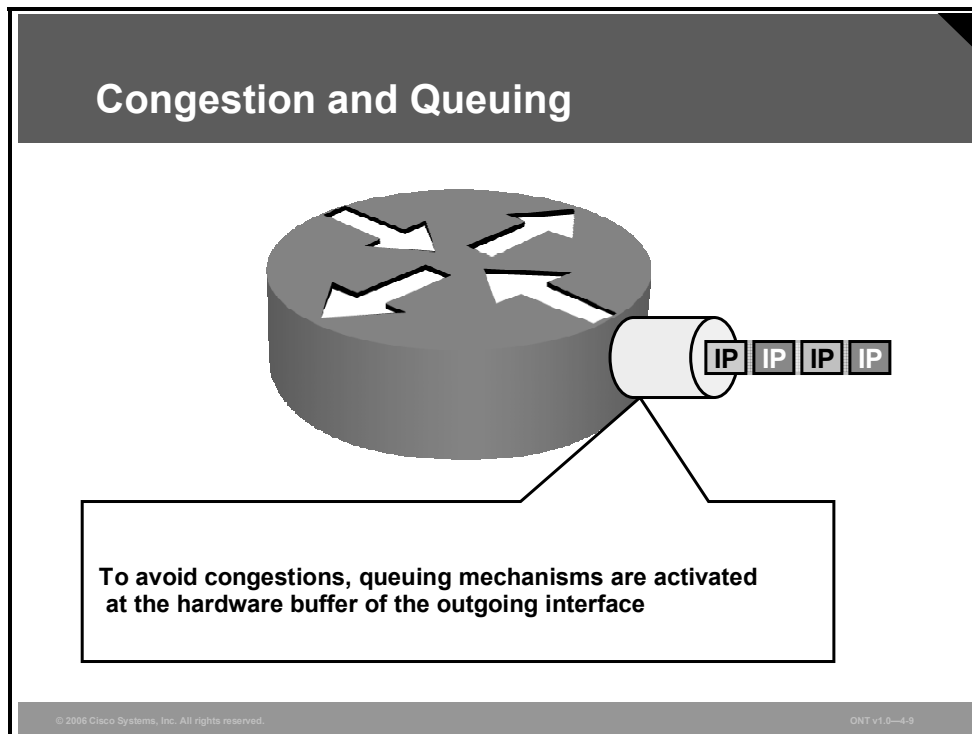
Queuing is designed to accommodate temporary congestion on an interface of a network device by storing excess packets in buffers until bandwidth becomes available or until the queue depth is exhausted and packets have to be dropped. Queuing is a congestion-management mechanism that allows you to control congestion by determining the order in which identified packets are sent out an interface based on priorities assigned to those packets. Congestion management entails creating queues, assigning packets to those queues based on the classification of the packet, and scheduling the packets in a queue for transmission. Cisco IOS routers support several queuing methods to meet the varying bandwidth, jitter, and delay requirements of different applications.

The default mechanism on most interfaces is the very simplistic FIFO queue. Some traffic types, such as voice and video, have very demanding delay and jitter requirements, so more sophisticated queuing mechanisms must be configured on interfaces used by voice and video traffic.

Complex queuing generally happens on outbound interfaces only. A router queues packets being transmitted out an interface.

## Congestion and Queuing

During periods with low traffic loads, when no congestion occurs, packets are sent out the interface as soon as they arrive. During periods of transmit congestion at the outgoing interface, packets arrive faster than the interface can send them.



When congestion-management features are being used, packets accumulating at an interface are placed in software queues according to their assigned priority and the queuing mechanism configured for the interface. They are then scheduled for transmission when the hardware buffer of the interface is free to send them. The router determines the order of packet transmission by controlling which packets are placed in each queue and how the queues are serviced with respect to each other.



# Queuing Algorithm Introduction

This subtopic describes specific queuing algorithms.

## Queuing Algorithms

- **First In First Out (FIFO)**
- **Priority Queuing (PQ)**
- **Round Robin (RR):**
  - **Weighted Round Robin (WRR)**

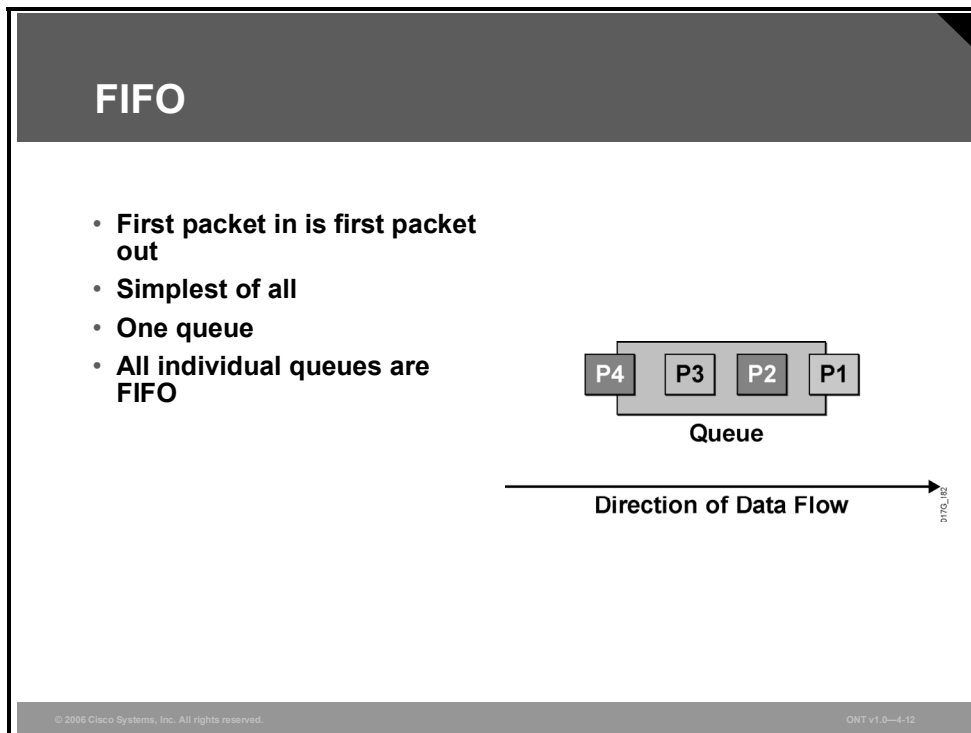
© 2006 Cisco Systems, Inc. All rights reserved. QNT v1.0—4-10

Key queuing algorithms include the following:

- **FIFO:** First in, first out; the simplest algorithm
- **Priority queuing (PQ):** Allows traffic to be prioritized
- **Round robin:** Allows several queues to share bandwidth
- **Weighted round robin (WRR):** Allows sharing of bandwidth with prioritization

# FIFO

This topic describes the FIFO queuing algorithm.



FIFO is the simplest queuing algorithm. FIFO provides basic store-and-forward capability. FIFO is the default queuing algorithm in some instances, thus requiring no configuration.

In its simplest form, FIFO queuing—also known as first-come, first-served (FCFS) queuing—involves storing packets when the network is congested and forwarding them in order of arrival when the network is no longer congested.

FIFO embodies no concept of priority or classes of traffic and consequently makes no decision about packet priority. There is only one queue, and all packets are treated equally. Packets are placed into a single queue and transmitted in the order in which they arrive. Higher-priority packets are not transmitted faster than lower-priority packets.

When FIFO is used, ill-behaved sources can consume all the bandwidth, and bursty sources can cause delays to time-sensitive or important traffic; also, important traffic can be dropped because less important traffic has filled the queue. When no other queuing strategies are configured, all interfaces except serial interfaces at E1 (2.048 Mbps) and below use FIFO by default. FIFO, which is the fastest method of queuing, is effective for large links that have little delay and minimal congestion. If your link has very little congestion, FIFO queuing may be the only queuing you need to use. All individual queues are, in fact, FIFO queues. Other queuing methods rely on FIFO as the underlying queuing mechanism for the discrete queues within the more complex queuing strategies that support advanced functions such as prioritization.

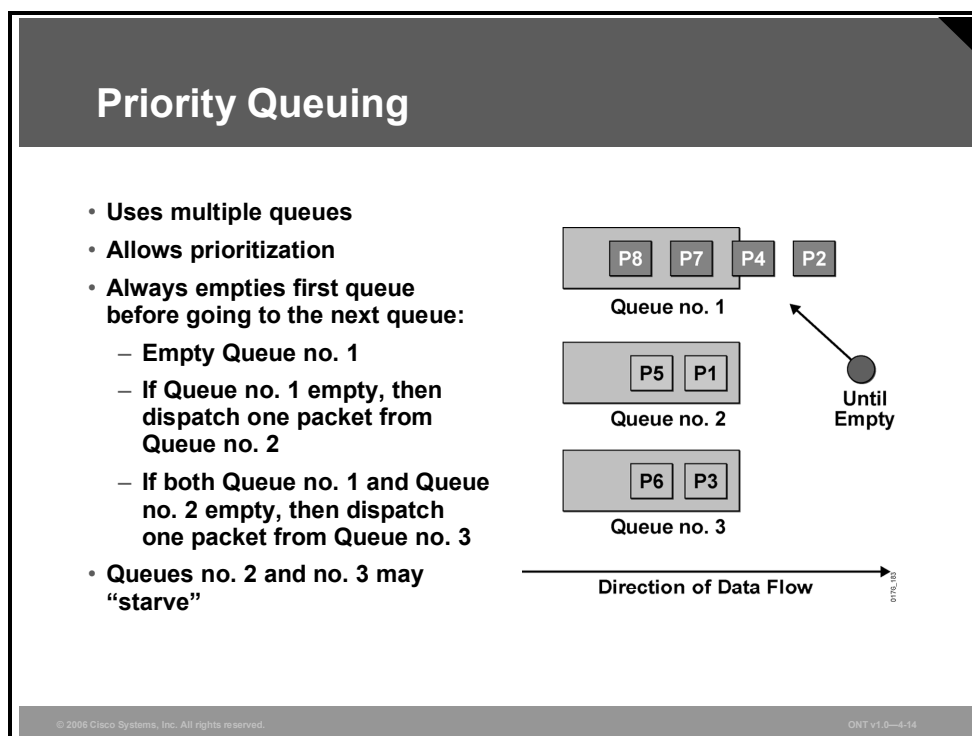
---

**Note** Serial interfaces at E1—2.048 Mbps—and below use weighted fair queuing (WFQ) by default.

---

# Priority Queuing

This topic describes the PQ algorithm.



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

During transmission, PQ gives priority queues absolute preferential treatment over low-priority queues; important traffic, given the highest priority, will always take precedence over less important traffic. Packets are classified based on user-specified criteria and placed in one of the four output queues—one, two, three, and four—based on the assigned priority. Packets that are not classified by priority fall into the normal queue.

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

Packets can be classified by the following:

- Protocol type
- Incoming interface
- Packet size
- Fragments
- Access control list (ACL)

Keepalives sourced by the network server are always assigned to the high-priority queue; all other management traffic (such as Enhanced Interior Gateway Routing Protocol [EIGRP] updates) must be configured. PQ provides absolute preferential treatment to high-priority traffic, ensuring that mission-critical traffic traversing various WAN links gets priority treatment. In addition, PQ provides a faster response time than do other methods of queuing.

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

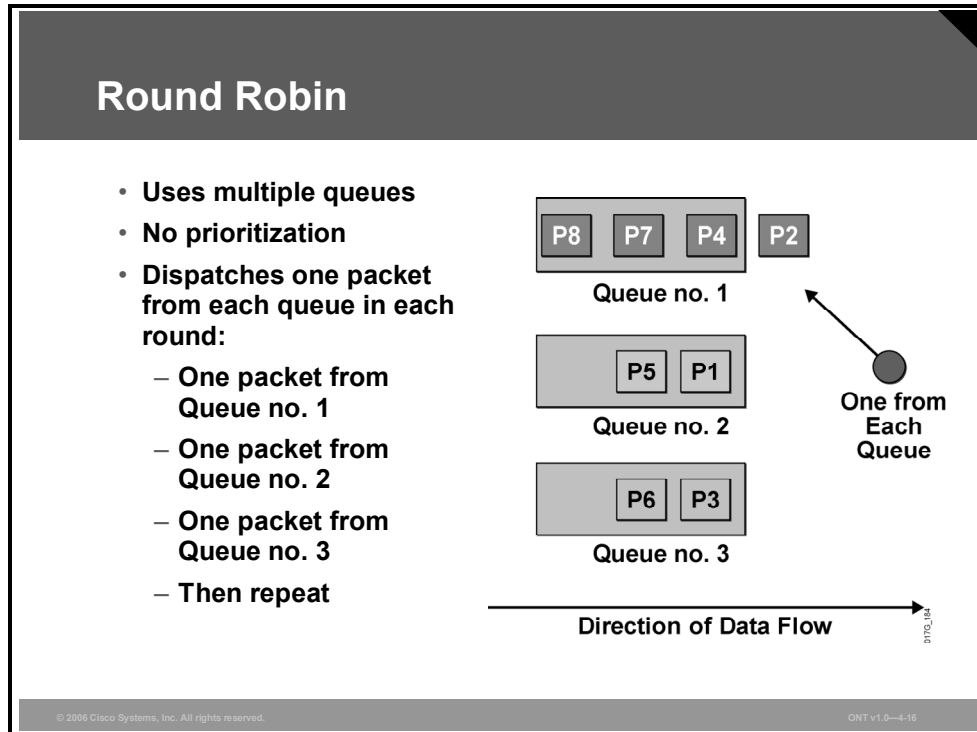
When you choose to use PQ, consider that, because lower-priority traffic is often denied bandwidth in favor of higher-priority traffic, the use of PQ could, in the worst case, result in lower-priority traffic never being transmitted (the lower-priority traffic class is “starved”). To avoid this problem, you can use traffic shaping to rate-limit the higher-priority traffic.

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

Furthermore, PQ uses a static configuration that does not adapt readily to changing network conditions.

# Round Robin

This topic describes the round-robin queuing algorithm and its variants.



A round robin is an arrangement of choosing all elements in a group equally in some rational order, usually starting from the top to the bottom of a list and then starting again at the top of the list and so on. A simple way to think of round robin is that it is about “taking turns.” In round-robin queuing, one packet is taken from each queue and then the process repeats.

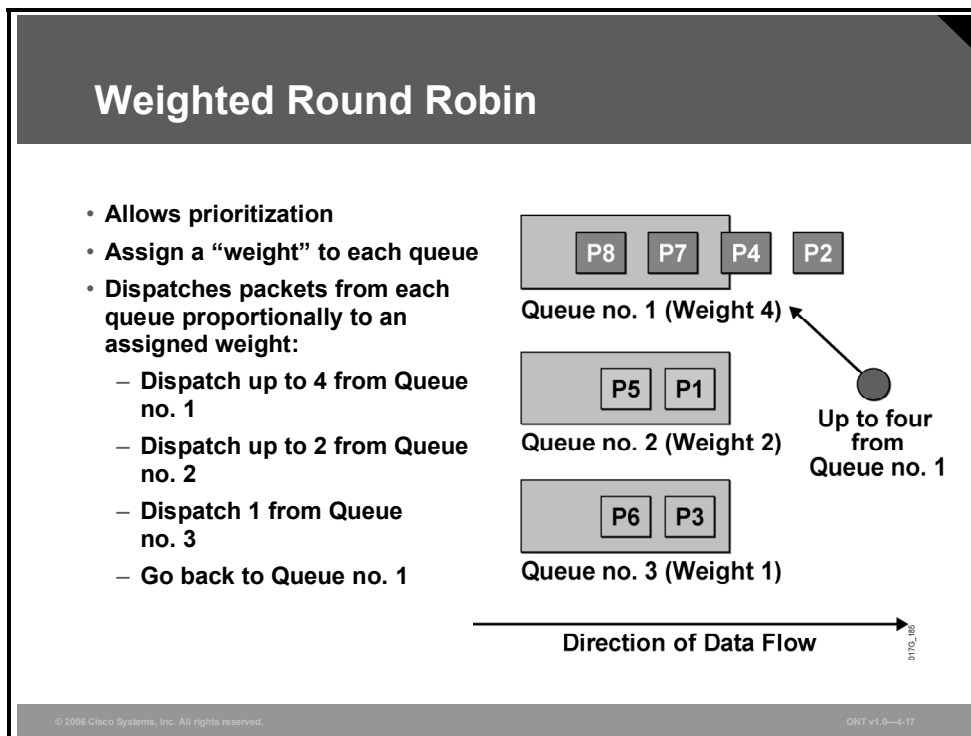
If all packets are the same size, all queues share the bandwidth equally. If packets being put into one queue are larger, that queue will receive a larger share of bandwidth.

No queue will “starve” with round-robin queuing because all queues receive an opportunity to dispatch a packet every round.

A limitation of round-robin queuing is the inability to prioritize traffic.

# Weighted Round Robin

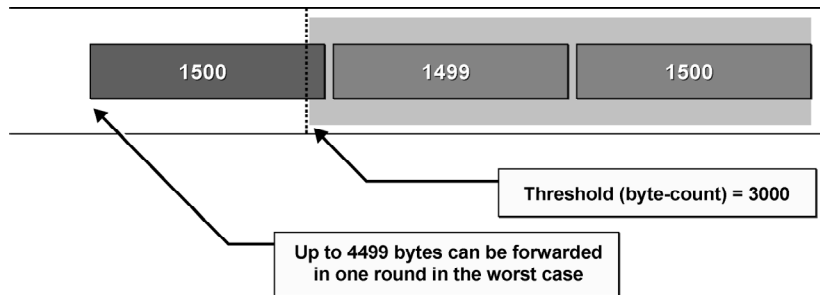
The WRR algorithm was developed to provide prioritization capabilities for round-robin queuing.



In WRR, packets are accessed round-robin style, but queues can be given priorities called “weights.” For example, in a single round, four packets from a high-priority class might be dispatched, followed by two from a middle-priority class, and then one from a low-priority class.

Some implementations of the WRR algorithm dispatch a configurable number of bytes during each round.

## Weighted Round Robin (Cont.)



### Problem with WRR:

- Some implementations of WRR dispatch a configurable number of bytes (threshold) from each queue for each round—several packets can be sent in each turn.
- The router is allowed to send the entire packet even if the sum of all bytes is more than the threshold.

Some implementations of the WRR algorithm provide prioritization by dispatching a configurable number of bytes each round rather than a number of packets. The Cisco custom queuing (CQ) mechanism is an example of this implementation.

This figure illustrates the worst-case scenario of the WRR algorithm, in which the following parameters were used to implement WRR queuing on an interface:

- The maximum transmission unit (MTU) of the interface is 1500 bytes.
- The byte count to be sent for the queue in each round is 3000 bytes (twice the MTU).

The example shows that the router first sent two packets with a total size of 2999 bytes. Because this size is within the limit (3000), the router can send the next packet (which is MTU sized). The result was that the queue received almost 50 percent more bandwidth in this round than it should have received.

This example shows one of the drawbacks of WRR queuing—it does not allocate bandwidth accurately.

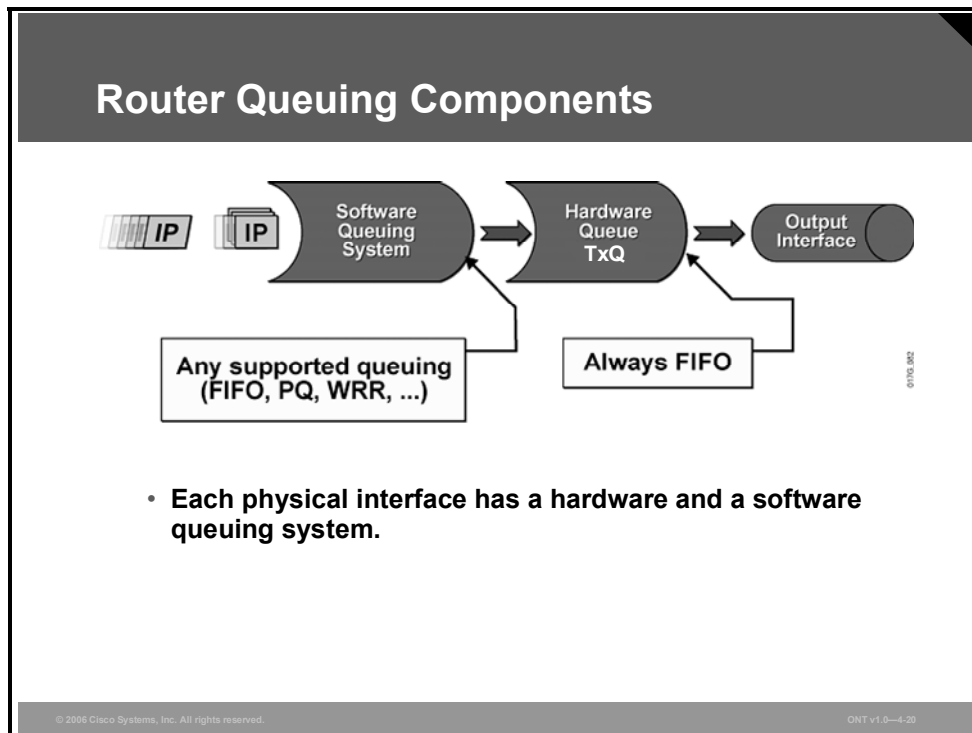
The limit or weight of the queue is configured in bytes. The accuracy of WRR queuing depends on the weight (byte count) and the MTU.

If the ratio between the byte count and the MTU is too small, WRR queuing will not allocate bandwidth accurately.

If the ratio between the byte count and the MTU is too large, WRR queuing will cause long delays.

# Router Queuing Components

This topic describes the primary components of a queuing mechanism.



Queuing on routers is necessary to accommodate bursts when the arrival rate of packets is greater than the departure rate, usually because of one of two reasons:

- The input interface is faster than the output interface.
- The output interface is receiving packets from multiple other interfaces.

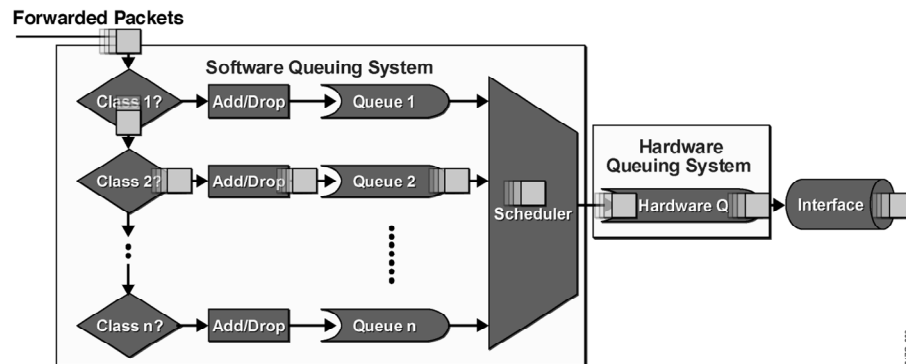
Initial implementations of queuing used a single FIFO strategy. More complex queuing mechanisms were introduced when special requirements required routers to differentiate among packets of different importance.

Queuing was split into two parts:

- **Hardware queue:** Uses FIFO strategy, which is necessary for the interface drivers to transmit packets one by one. The hardware queue is sometimes referred to as the transmit queue.
- **Software queuing system:** Schedules packets into the hardware queue based on the quality of service (QoS) requirements.



## Router Queuing Components (Cont.)



- The hardware queuing system always uses FIFO queuing.
- The software queuing system can be selected and configured depending on the platform and Cisco IOS version.

© 2006 Cisco Systems, Inc. All rights reserved.

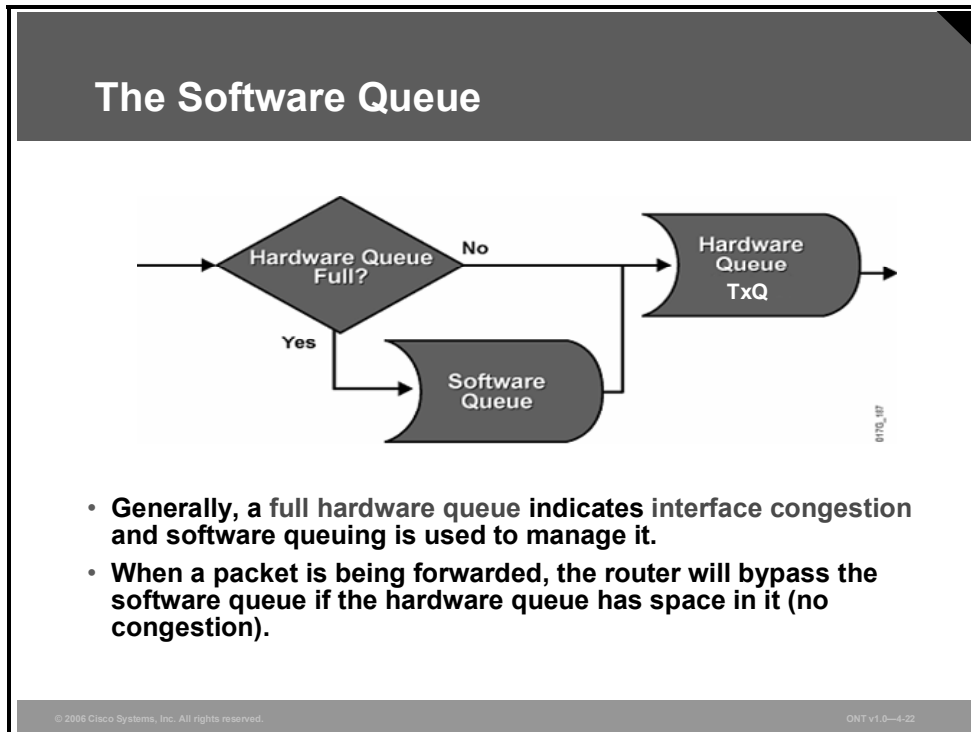
QNT v1.0—4-21

This figure illustrates the actions that must occur before a packet can be transmitted:

- Most queuing mechanisms include classification of packets.
- After a packet is classified, a router has to determine whether it can place the packet in the queue or drop the packet. Most queuing mechanisms will drop a packet only if the corresponding queue is full (tail drop). Some mechanisms use a more intelligent dropping scheme, such as WFQ, or a random dropping scheme, such as weighted random early detection (WRED).
- If the packet is allowed to be queued, it is put into the FIFO queue for that particular class.
- Packets are then taken from the individual per-class queues and put into the hardware queue.

# The Software Queue

The implementation of software queuing is optimized for periods when the interface is not congested. The software queuing system is bypassed whenever there is no packet in the software queue and there is room in the hardware queue.



The software queue is used only when data must wait to be placed into the hardware queue.

# The Hardware Queue

The double-queuing strategy (software and hardware queues) has its impacts on the results of overall queuing. Software queues serve a valuable purpose. If the hardware queue is too long, it will contain a large number of packets scheduled in the FIFO fashion. A long FIFO hardware queue most likely defeats the purpose of the QoS design requiring a certain complex software queuing system (for example, CQ).

## The Hardware Queue

- **Routers determine the length of the hardware queue based on the configured bandwidth of the interface.**
- **The length of the hardware queue can be adjusted with the `tx-ring-limit` command.**
- **Reducing the size of the hardware queue has two benefits:**
  - **It reduces the maximum amount of time packets wait in the FIFO queue before being transmitted.**
  - **It accelerates the use of QoS in the Cisco IOS software.**
- **Improper tuning of the hardware queue may produce undesirable results:**
  - **Long TxQ may result in poor performance of the software queuing system.**
  - **Short TxQ may result in a large number of interrupts, which causes high CPU utilization and low link utilization.**

© 2006 Cisco Systems, Inc. All rights reserved.

QNT v1.0—4-23

The hardware queue (transmit queue) is a final interface FIFO queue that holds frames to be immediately transmitted by the physical interface. The transmit queue ensures that a frame is always available when the interface is ready to transmit traffic, so that link utilization is driven to 100 percent of capacity.

Why use the hardware queue at all? Or why not just set its length to one? Doing so would force all packets to go through the software queue and be scheduled one by one to the interface for transmission. This approach has these drawbacks:

- Each time a packet is transmitted, the interface driver interrupts the CPU and requests more packets to be delivered to its hardware queue. Some queuing mechanisms have complex scheduling that takes time to deliver more packets. The interface does not send anything during that time (link utilization is decreased) if the hardware queue is empty, because its maximum size is one.
- The CPU schedules packets one by one instead of many at the same time (in the same interrupt interval). This process increases the CPU utilization.

The length of the transmit queue is dependant on the hardware, software, Layer 2 media, and queuing algorithm configured on the interface. The default transmit queue size is determined by Cisco IOS software, based on the bandwidth of the media, and should be fine for most queuing implementations. Some platforms and QoS mechanisms automatically adjust the transmit queue size to an appropriate value. Faster interfaces have longer hardware queues because they produce less delay. Slower interfaces have shorter hardware queues to prevent too much delay in the worst-case scenario in which the entire hardware queue is full of MTU-size packets.

## The Hardware Queue (Cont.)

- The **show controllers serial 0/1/0** command shows the length of the hardware queue.

```
R1#show controllers serial 0/1/0
Interface Serial0/1/0
Hardware is GT96K
DCE V.11 (X.21), clock rate 384000

<...part of the output omitted...>
1 sdma_rx_reserr, 0 sdma_tx_reserr
0 rx_bogus_pkts, rx_bogus_flag FALSE
0 sdma_tx_ur_processed

tx_limited = 1(2), errata19 count1 - 0, count2 - 0
Receive Ring
rxr head (27) (0x075BD090), rxr tail (0) (0x075BCEE0)
  rmd(75BCEE0): nbd 75BCEF0 cmd_sts 80800000 buf_sz 06000000 buf_ptr
75CB8E0
  rmd(75BCEF0): nbd 75BCF00 cmd_sts 80800000 buf_sz 06000000 buf_ptr
75CCC00
<...rest of the output omitted...>
```

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0-4.24

The length of the transmit queue depends on several factors and is adjusted automatically based on the configuration applied to the interface. To avoid too much delay on the slow interface and to avoid bad link utilization, check the length of the transmit queue and change it if needed. The default length works fine in almost all cases.

The **show controllers serial 0/1/0** command is used to see the length of the transmit queue. The transmit queue length is shown as the `tx_limited` or `tx_ring_limit` or `tx_ring` statement and varies depending on the platform.

In this example, the hardware queue length is defined by the `tx_limited` statement and equals two packets.

# Congestion on Software Interfaces

Subinterfaces and software interfaces do not have their own separate transmit queues; they use the main transmit queue.

## Congestion on Software Interfaces

- Subinterfaces and software interfaces (dialers, tunnels, Frame Relay subinterfaces) do not have their own separate TxQ.
- Subinterfaces and software interfaces congest when the TxQ of their main hardware interface congests.
- The tx-ring state (full, not-full) is an indication of hardware interface congestion.

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0--4.25

Software interface types include dialers, tunnels, and Frame Relay subinterfaces, and they will congest only when their main hardware interface transmit queue congests. The transmit (tx-ring) state is an indication of congestion of hardware interfaces caused by a congestion on the main hardware interface.

---

**Note**      The terms “TxQ” and “tx-ring” both describe the hardware queue and are interchangeable.

---

# Summary

This topic summarizes the key points that were discussed in this lesson.

## Summary

- Congestion can occur at any point in the network, but particularly at points of speed mismatches and traffic aggregation.
- Three basic queuing algorithms are used to manage congestion: FIFO, priority, and round-robin queuing.
- FIFO is the simplest queuing algorithm.
- PQ allows for the prioritization of traffic through the use of multiple queues but can starve lower-priority queues.
- RR queuing uses multiple queues to provide equal access to all queues.
- WRR offers priority access to multiple queues by assigning “weights” to queues, but some implementations may provide inaccurate access to some queues.

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—4-26

## Summary (Cont.)

- Each physical interface has a hardware and a software queuing system. The hardware queuing system uses FIFO, while the software queuing system can be configured depending on the platform and IOS version.
- The length of the hardware queue has a significant impact on performance and can be configured on a router with the `tx-ring-limit` command.
- Software interfaces have no queues; they congest only when their hardware interface congests.

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—4-27

# Configuring WFQ

---

## Overview

Weighted fair queuing (WFQ) is one of the primary default queuing mechanisms that are implemented on Cisco routers. Because WFQ is a basic queuing mechanism, its capabilities are limited, but WFQ is the basis for advanced queuing mechanisms like class-based weighted fair queuing (CBWFQ) and low latency queuing (LLQ).

## Objectives

Upon completing this lesson, you will be able to explain the procedure for configuring WFQ mechanisms. This ability includes being able to meet these objectives:

- Give a detailed explanation of WFQ
- Describe the architecture and benefits of WFQ
- Identify the Cisco IOS commands required to configure and monitor WFQ on a Cisco router

# Weighted Fair Queuing

This topic describes the purpose and features of weighted fair queuing (WFQ).

## Weighted Fair Queuing

- **A queuing algorithm should share the bandwidth fairly among flows by:**
  - **Reducing response time for interactive flows by scheduling them to the front of the queue**
  - **Preventing high-volume flows from monopolizing an interface**
- **In the WFQ implementation, conversations are sorted into flows and transmitted by the order of the last bit crossing its channel.**
- **Unfairness is reinstated by introducing weight to give proportionately more bandwidth to flows with higher IP precedence (lower weight).**

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—4.3

WFQ is one of the premier Cisco queuing techniques. It is a flow-based queuing algorithm that does two things simultaneously: It schedules interactive traffic to the front of the queue to reduce response time, and it fairly shares the remaining bandwidth among the various flows to prevent high-volume flows from monopolizing the outgoing interface.

The idea of WFQ is to have a dedicated queue for each flow without starvation, delay, or jitter within the queue. Furthermore, WFQ allows fair and accurate bandwidth allocation among all flows with minimum scheduling delay. WFQ makes use of the IP precedence bits as a weight when allocating bandwidth.

WFQ was introduced as a solution to the problems of the following queuing mechanisms:

- FIFO queuing causes starvation, delay, and jitter.
- Priority queuing (PQ) causes starvation of lower-priority classes and suffers from the FIFO problems within each of the four queues that it uses for prioritization.

---

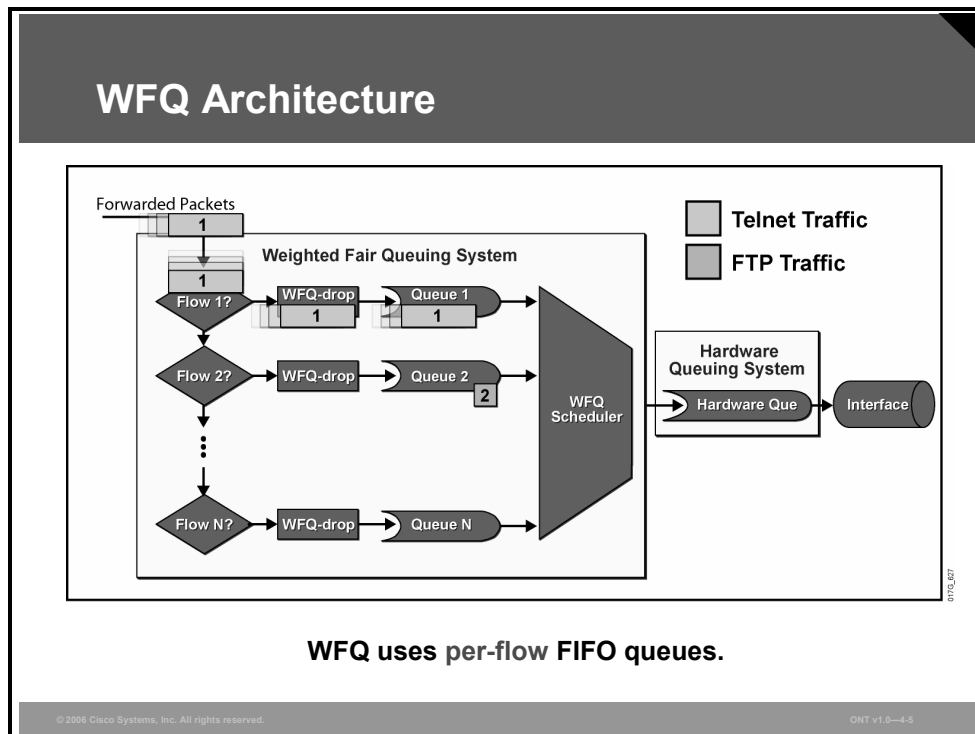
**Note** WFQ flows are also called “conversations.” These terms can be interchanged.

---



# WFQ Architecture and Benefits

This topic describes the WFQ architecture and its benefits compared to FIFO.



WFQ is a dynamic scheduling method that provides fair bandwidth allocation to all network traffic. WFQ applies weights to identified traffic, classifies traffic into flows, and determines how much bandwidth each flow is allowed, relative to other flows. WFQ is a flow-based algorithm that simultaneously schedules interactive traffic to the front of a hardware queue to reduce response time and fairly shares the remaining bandwidth among high-bandwidth flows. In other words, WFQ allows you to give low-volume traffic, such as Telnet sessions, priority over high-volume traffic, such as FTP sessions. WFQ gives concurrent file transfers balanced use of link capacity; that is, when multiple file transfers occur, the transfers are given comparable bandwidth.

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

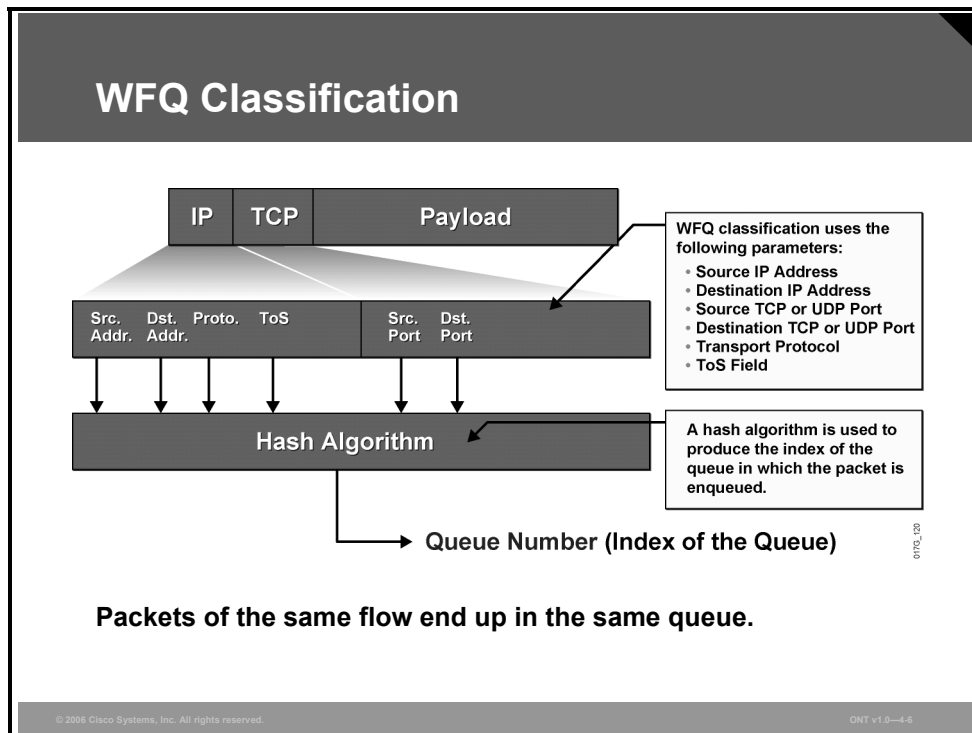
WFQ provides the solution for situations in which it is desirable to provide consistent response times to heavy and light network users alike, without adding excessive bandwidth. In addition, WFQ can manage duplex data flows, such as those between pairs of applications, and simplex data flows, such as voice or video.

Although WFQ automatically adapts to changing network traffic conditions, it does not offer the precise degree of control over bandwidth allocation that custom queuing (CQ) and class-based weighted fair queuing (CBWFQ) offer.

The significant limitation of WFQ is that it is not supported with tunneling and encryption because these features modify the packet content information required by WFQ for classification.

# WFQ Classification

WFQ classification has to identify individual flows.



A flow is identified based on the following information taken from the IP header and the TCP or User Datagram Protocol (UDP) headers:

- Source IP address
- Destination IP address
- Protocol number (identifying TCP or UDP)
- Type of service field
- Source TCP or UDP port number
- Destination TCP or UDP port number

These parameters are usually fixed for a single flow, although there are some exceptions. For example, a quality of service (QoS) design can mark packets with different IP precedence bit values even if they belong to the same flow. You should avoid such marking when using WFQ.

The parameters are used as input for a hash algorithm that produces a fixed-length number that is used as the index of the queue.

## WFQ Classification (Cont.)

- **A fixed number of per-flow queues is configured.**
- **A hash function is used to translate flow parameters into a queue number.**
- **System packets (eight queues) and RSVP flows (if configured) are mapped into separate queues.**
- **Two or more flows could map into the same queue, resulting in lower per-flow bandwidth.**
- **Important: The number of queues configured has to be significantly larger than the expected number of flows.**

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—4-7

WFQ uses a fixed number of queues. The hash function is used to assign a queue to a flow. There are eight additional queues for system packets and optionally up to 1000 queues for Resource Reservation Protocol (RSVP) flows. The number of dynamic queues that WFQ uses by default is based on the interface bandwidth. With the default interface bandwidth, WFQ uses 256 dynamic queues. The number of queues can be configured in the range between 16 and 4096 (the number must be a power of 2).

Should there be a large number of concurrent flows, it is likely that two flows could end up in the same queue. You should have several times as many queues as there are flows (on average). This design may not be possible in larger environments where concurrent flows number in the thousands.

# WFQ Insertion and Drop Policy

This section topic describes the WFQ insertion and drop policy.

## WFQ Insertion and Drop Policy

- **WFQ has two modes of dropping:**
  - **Early dropping when the congestive discard threshold is reached**
  - **Aggressive dropping when the hold-queue limit is reached**
- **WFQ always drops packets of the most aggressive flow.**
- **Drop mechanism exceptions:**
  - **A packet classified into an empty queue is never dropped.**
  - **The packet IP precedence has no effect on the dropping scheme.**

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—4-8

The WFQ system has a hold queue that represents the queue depth, which means the number of packets that can be held in the queue. WFQ uses the following two parameters that affect the dropping of packets:

- The congestive discard threshold (CDT) is used to start dropping packets of the most aggressive flow, even before the hold-queue limit is reached.
- The hold-queue limit defines the maximum number of packets that can be held in the WFQ system at any time.

There are two exceptions to the WFQ insertion and drop policy:

- If the WFQ system is above the CDT limit, the packet is still enqueued if the specific per-flow queue is empty.
- The dropping strategy is not directly influenced by IP precedence.

## Finish Time

The length of queues (for scheduling purposes) is determined not by the sum of the size in bytes of all the packets but by the time it would take to transmit all the packets in the queue. The end result is that WFQ adapts to the number of active flows (queues) and allocates equal amounts of bandwidth to each flow (queue).

The side effect is that flows with small packets (usually interactive flows) get much better service because they do not need a lot of bandwidth. They need low-delay handling, however, which they get because small packets have a low finish time.

## Benefits and Drawbacks of WFQ

The WFQ mechanism provides simple configuration (no manual classification is necessary) and guarantees throughput to all flows. It drops packets of the most aggressive flows. Because WFQ is a standard queuing mechanism, it is supported on most platforms and in most Cisco IOS versions.

Benefits and Drawbacks of WFQ	
Benefits	<ul style="list-style-type: none"><li>• Simple configuration (no need for classification to be configured)</li><li>• Guarantees throughput to all flows</li><li>• Drops packets of most aggressive flows</li><li>• Supported on most platforms</li><li>• Supported in most Cisco IOS versions</li></ul>
Drawbacks	<ul style="list-style-type: none"><li>• Possibility of multiple flows ending up in one queue</li><li>• Lack of control over classification</li><li>• Supported only on links less than or equal to 2 Mb</li><li>• Cannot provide fixed bandwidth guarantees</li></ul>

© 2006 Cisco Systems, Inc. All rights reserved. DNT v1.0—4-9

As good as WFQ is, it does have its drawbacks:

- Multiple flows can end up in a single queue.
- WFQ does not allow a network engineer to manually configure classification. Classification and scheduling are determined by the WFQ algorithm.
- WFQ is supported only on links with a bandwidth less than or equal to 2 Mb.
- WFQ cannot provide fixed guarantees to traffic flows.

# Configuring and Monitoring WFQ

This topic describes the Cisco IOS commands required to configure WFQ on a Cisco router.

## Configuring WFQ

```
router(config-if)#
```

```
fair-queue [cdt [dynamic-queues [reservable-  
queues]]]
```

- **cdt:** Number of messages allowed in each queue (a new threshold must be a power of 2 in the range from 16 to 4096; default is 64). When a conversation reaches this threshold, new message packets are discarded.
- **dynamic-queues:** Number of dynamic queues used for best-effort conversations (values are: 16, 32, 64, 128, 256, 512, 1024, 2048, and 4096; the default is 256).
- **reservable-queues:** Number of reservable queues used for reserved conversations in the range 0 to 1000 (used for interfaces configured for features such as RSVP—the default is 0).

© 2006 Cisco Systems, Inc. All rights reserved.ONT v1.0—4-11

WFQ is automatically enabled on all interfaces that have a default bandwidth of less than 2.048 Mbps. The **fair-queue** command is used to enable WFQ on interfaces where it is not enabled by default or was previously disabled.

**fair-queue** [*congestive-discard-threshold* [*dynamic-queues* [*reservable-queues*]]]

### fair-queue Parameters

Parameter	Description
<i>congestive-discard-threshold</i>	(Optional) Number of messages allowed in the WFQ system. The default is 64 messages, and a new threshold must be a power of 2 in the range from 16 to 4096. When a conversation reaches this threshold, new message packets are discarded.
<i>dynamic-queues</i>	(Optional) Number of dynamic queues used for best-effort conversations. Values are 16, 32, 64, 128, 256, 512, 1024, 2048, and 4096.
<i>reservable-queues</i>	(Optional) 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 RSVP.

## Additional WFQ Configuration Parameters

This subtopic describes additional WFQ configuration parameters.

### Additional WFQ Configuration Parameters

```
router(config-if)#  
hold-queue max-limit out
```

- Specifies the maximum number of packets that can be in all output queues on the interface at any time.
- The default value for WFQ is 1.000.
- Under special circumstances, WFQ can consume a lot of buffers, which may require lowering this limit.

© 2006 Cisco Systems, Inc. All rights reserved. ORT v1.0—4.12

The WFQ system will generally never reach the hold-queue limit because the CDT limit starts dropping the packets of aggressive flows in the software queue. Under special circumstances, it would be possible to fill the WFQ system. For example, a denial-of-service attack that floods the interface with a large number of packets (each different) could fill all queues at the same rate.

# Monitoring WFQ

The **show interface** command can be used to determine the queuing strategy. The summary statistics are also displayed.

## Monitoring WFQ

```
router>
```

```
show interface interface
```

- Displays interface delays including the activated queuing mechanism with the summary information

```
Router>show interface serial 1/0
Hardware is M4T
Internet address is 20.0.0.1/8
MTU 1500 bytes, BW 19 Kbit, DLY 20000 usec, rely 255/255, load
147/255
Encapsulation HDLC, crc 16, loopback not set
Keepalive set (10 sec)
Last input 00:00:00, output 00:00:00, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0 (size/max/drops); Total output drops: 0
Queueing strategy: weighted fair
Output queue: 0/1000/64/0 (size/max total/threshold/drops)
Conversations 0/4/256 (active/max active/max total)
Reserved Conversations 0/0 (allocated/max allocated)
5 minute input rate 18000 bits/sec, 8 packets/sec
5 minute output rate 11000 bits/sec, 9 packets/sec
... rest deleted ...
```

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—4-13

The sample output in this figure shows that there are currently no packets in the WFQ system. The system allows up to 1000 packets (hold-queue limit) with a CDT of 64. WFQ is using 256 queues. The maximum number of concurrent flows (conversations, or active queues) is four.



## Monitoring WFQ (Cont.)

router>

```
show queue interface-name interface-number
```

- Displays detailed information about the WFQ system of the selected interface

```
Router>show queue serial 1/0
Input queue: 0/75/0 (size/max/drops); Total output drops: 0
Queueing strategy: weighted fair
Output queue: 2/1000/64/0 (size/max total/threshold/drops)
Conversations 2/4/256 (active/max active/max total)
Reserved Conversations 0/0 (allocated/max allocated)

(depth/weight/discards/tail drops/interleaves) 1/4096/0/0/0
Conversation 124, linktype: ip, length: 580
source: 193.77.3.244, destination: 20.0.0.2, id: 0x0166, ttl: 254,
TOS: 0 prot: 6, source port 23, destination port 11033

(depth/weight/discards/tail drops/interleaves) 1/4096/0/0/0
Conversation 127, linktype: ip, length: 585
source: 193.77.4.111 destination: 40.0.0.2, id: 0x020D, ttl: 252,
TOS: 0 prot: 6, source port 23, destination port 11013
```

© 2006 Cisco Systems, Inc. All rights reserved.

QNT v1.0—4-14

The **show queue** command is used to display the contents of packets inside a queue for a particular interface, including flow (conversation) statistics:

- Queue depth is the number of packets in the queue.
- Weight is 4096 / (IP precedence + 1), or 32,384 / (IP precedence + 1), depending on the Cisco IOS version.
- In the command output, discards are used to represent the number of drops that are due to the CDT limit.
- In the command output, tail drops are used to represent the number of drops that are due to the hold-queue limit.

# Summary

This topic summarizes the key points that were discussed in this lesson.

## Summary

- **WFQ was developed to overcome the limitations of the more basic queuing methods. Traffic is sorted into flows and transmitted by the order of the last bit crossing its channel.**
- **WFQ is automatically enabled on serial interfaces that have a default bandwidth of less than 2 Mbps.**
- **WFQ benefits include simple configuration and the dropping of packets of the most aggressive flows. However, a flow can end up in a queue of a different flow, WFQ does not allow manual classification, and if cannot provide fixed guarantees.**
- **WFQ is the basis for advanced queuing mechanisms like CBWFQ and LLQ.**

# Configuring CBWFQ and LLQ

---

## Overview

Class-based weighted fair queuing (CBWFQ) extends the standard weighted fair queuing (WFQ) functionality to provide support for user-defined traffic classes. With CBWFQ, you define traffic classes based on match criteria, including protocols, access control lists (ACLs), and input interfaces. Packets satisfying the match criteria for a class constitute the traffic for that class. A queue is reserved for each class, and traffic belonging to a class is directed to the queue for that class. Low latency queuing (LLQ) extends the capabilities of CBWFQ with support for a single strict-priority queue. Strict-priority queuing gives delay-sensitive data, such as voice, preferential treatment over all the other queued traffic.

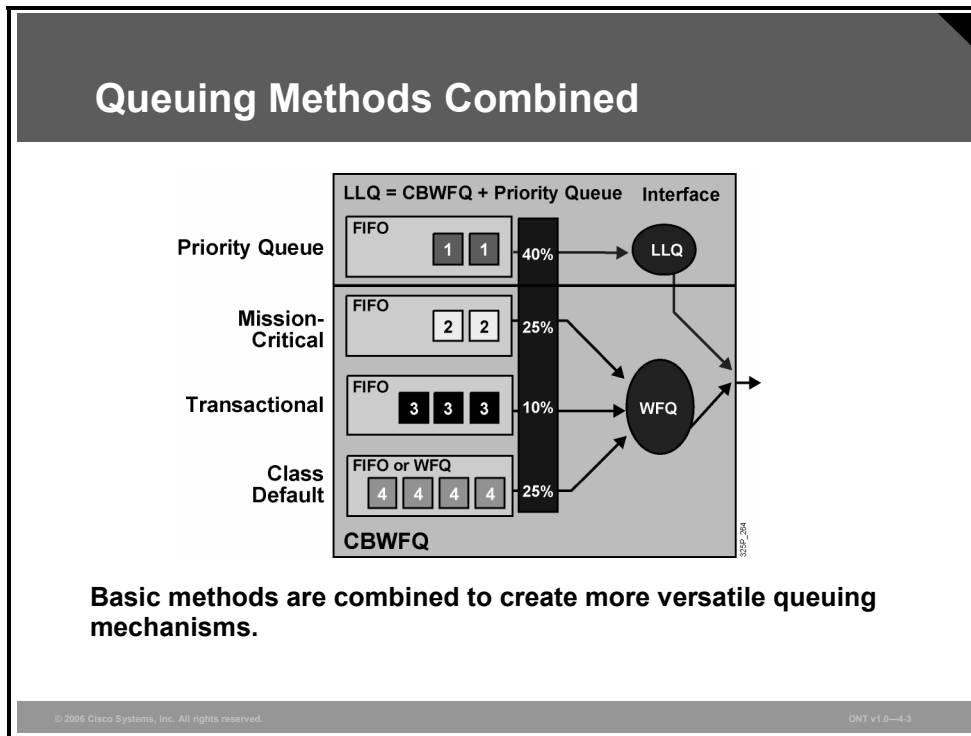
## Objectives

Upon completing this lesson, you will be able to explain the procedure for configuring queuing mechanisms, including CBWFQ and LLQ, on a router. This ability includes being able to meet these objectives:

- Describe the advanced queuing mechanisms of CBWFQ and LLQ
- Give a detailed explanation of CBWFQ
- Describe the architecture and benefits of CBWFQ
- Identify the Cisco IOS commands required to configure and monitor CBWFQ on a Cisco router
- Give a detailed explanation of LLQ
- Explain the architectures and benefits of LLQ
- Identify the Cisco IOS commands required to configure and monitor LLQ on a Cisco router

# Describing Advanced Queuing Mechanisms

This topic describes how basic queuing mechanisms can be used to build more advanced queuing mechanisms such as CBWFQ and LLQ.



Neither the basic queuing methods nor the more advanced weighted fair queuing (WFQ) method completely solves the quality of service (QoS) problems resulting from converged network traffic. The following problems remain:

- If only a priority queue is used for a voice-enabled network, voice gets the needed priority. However, data traffic would suffer.
- If only custom queuing (CQ) is used for a voice-enabled network, data traffic is assured of some bandwidth. However, voice traffic would suffer delays.
- If WFQ is used, voice still experiences delay even when treated “fairly” by WFQ.
- In PQ and CQ, all of the classification, marking, and queuing mechanisms are complicated to use and time-consuming when applied on an interface-by-interface basis.

Newer queuing mechanisms were developed to combine the best aspects of existing queuing methods. LLQ is a combination of CBWFQ, which assigns weights according to bandwidth, and a priority system based on class that gives voice the priority it requires while ensuring that data is serviced efficiently. The potential starvation problem of the priority queue is solved by including a policing function based on the configured bandwidth of the priority system.

# Class-Based Weighted Fair Queuing

This topic describes the purpose and features of CBWFQ.

## Class-Based Weighted Fair Queuing

- **CBWFQ is a mechanism that is used to guarantee bandwidth to classes.**
- **CBWFQ extends the standard WFQ functionality to provide support for user-defined traffic classes:**
  - **Classes are based on user-defined match criteria.**
  - **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 that class queue.**

© 2006 Cisco Systems, Inc. All rights reserved.

QNT v1.0—4.5

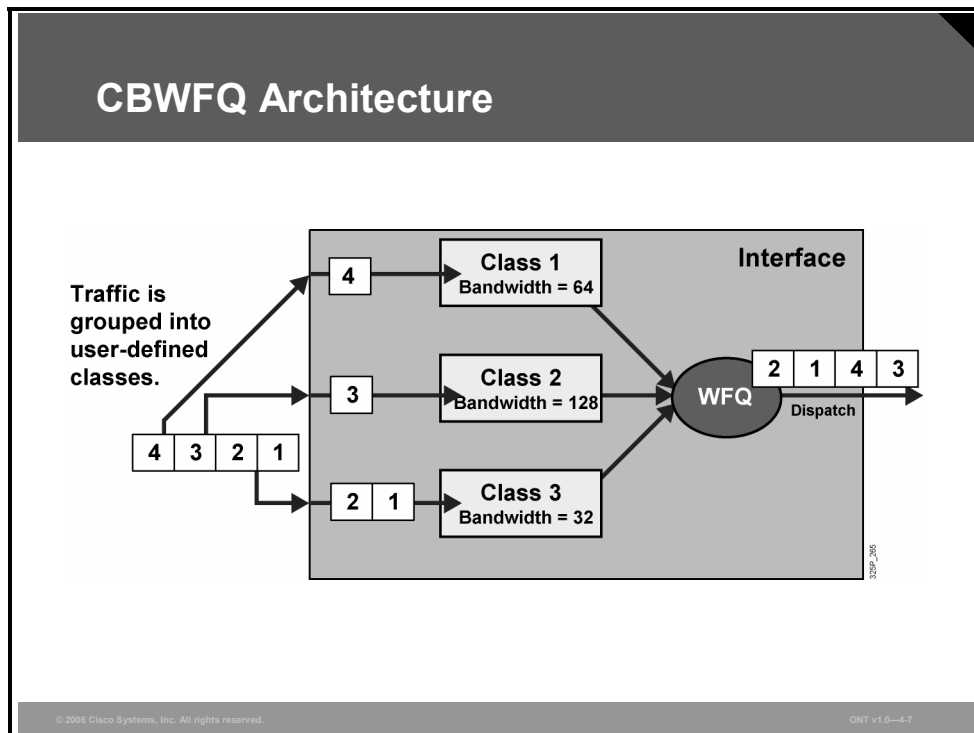
CBWFQ extends the standard WFQ functionality to provide support for user-defined traffic classes. With CBWFQ, the user defines the traffic classes based on match criteria, including protocols, ACLs, and input interfaces. Packets satisfying the match criteria for a class constitute the traffic for that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

After a class has been defined and its match criteria have been formulated, you can assign characteristics to the class. To characterize a class, you assign it bandwidth and maximum packet limit. The bandwidth assigned to a class is the minimum 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 class queue. The queuing guarantees the minimum bandwidth, but also gives a class unlimited access to more bandwidth if more is available. After a queue has reached its configured queue limit, enqueueing of additional packets to the class causes tail drop or random packet drop, depending on how the class policy is configured. You can configure up to 64 discrete classes in a service policy.

# CBWFQ Architecture and Benefits

This topic describes the architecture of CBWFQ and how CBWFQ works.



CBWFQ supports multiple class maps to classify traffic into its corresponding FIFO queues.

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

Serial interfaces at E1 (2.048 Mbps) and below use WFQ by default—other interfaces use FIFO by default. Enabling CBWFQ on a physical interface overrides the default interface queuing method.

Caution should be taken when configuring CBWFQ on ATM interfaces. Enabling CBWFQ on an ATM permanent virtual circuit (PVC) does not override the default queuing method. Also, CBWFQ is not supported on unspecified bit rate (UBR) connections.

# Classification

Any classification option for CBWFQ can be used, depending on availability in the Cisco IOS version, support on the selected interface, and encapsulation.

## Classification

- **Classification uses class maps.**
- **Availability of certain classification options depends on the Cisco IOS version.**
- **Some classification options depend on type of interface and encapsulation where service policy is used.**
- **For example:**
  - **Matching on Frame Relay discard-eligible bits can be used only on interfaces with Frame Relay encapsulation.**
  - **Matching on MPLS experimental bits has no effect if MPLS is not enabled.**
  - **Matching on ISL priority bits has no effect if ISL is not used.**

© 2006 Cisco Systems, Inc. All rights reserved. OMT v1.0—4-8

The following examples illustrate some of the limitations regarding classification options:

- Matching on Frame Relay discard-eligible bits can be used only on interfaces with Frame Relay encapsulation.
- Matching on Multiprotocol Label Switching (MPLS) experimental (EXP) bits has no effect if MPLS is not enabled.
- Matching on Inter-Switch Link (ISL) priority bits has no effect if ISL is not used.

It is important to note that CBWFQ is configured using Cisco Modular QoS CLI (MQC). The classifications that can be configured depend on the Cisco IOS version and the type of interface that is configured.

# Scheduling

This section describes the scheduling mechanism of CBWFQ.

## Scheduling

- **CBWFQ guarantees bandwidth according to weights assigned to traffic classes.**
- **Weights are internally calculated from bandwidth or its percentage.**
- **Bandwidth availability can be defined by specifying:**
  - **Bandwidth (in kbps)**
  - **Percentage of bandwidth (percentage of available interface bandwidth)**
  - **Percentage of remaining available bandwidth**
- **One service policy can not have mixed types of weights.**
- **The show interface command can be used to display the available bandwidth.**

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—4-9

The CBWFQ mechanism calculates weights based on the available bandwidth. These weights are then used by the CBWFQ scheduling mechanism to dispatch the packets.

You can configure bandwidth guarantees by using one of the following commands:

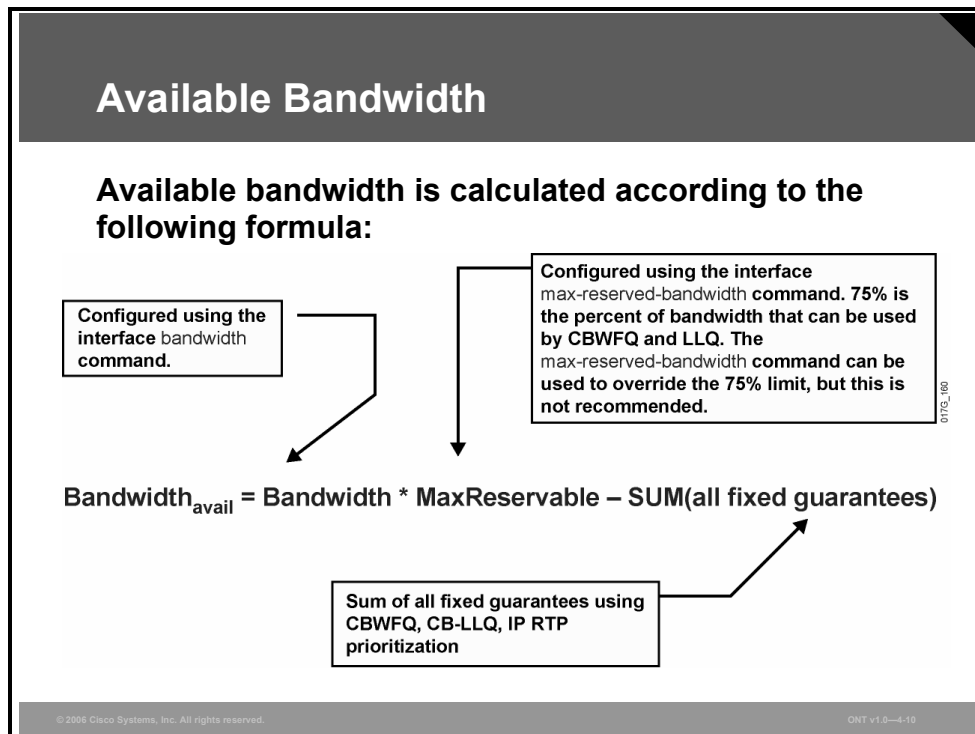
- The **bandwidth** command allocates a fixed amount of bandwidth by specifying the amount in kilobits per second. The reserved bandwidth is subtracted from the available bandwidth of the interface where the service policy is used. The allocated bandwidth must also be within the configured reservable limit (75 percent of interface bandwidth by default).
- The **bandwidth percent** command can be used to allocate a percentage of the total available bandwidth of an interface. The total interface bandwidth is defined by using the **bandwidth interface** command. It is recommended that the **bandwidth** command reflect the real speed of the link. The allocated bandwidth is subtracted from the available bandwidth of the interface where the service policy is used.
- The **bandwidth remaining percent** command is used to allocate the amount of guaranteed bandwidth based on a relative percentage of available bandwidth. When the **bandwidth remaining percent** command is configured, hard bandwidth guarantees may not be provided, and only relative per-class bandwidths are assured. That is, class bandwidths are always proportionate to the specified percentages of the interface bandwidth. When the link bandwidth is fixed, class bandwidth guarantees are in proportion to the configured percentages. If the link bandwidth is unknown or variable, class bandwidth guarantees in kilobits per second cannot be computed.

A single service policy cannot mix the **bandwidth** (fixed, in kilobits per second) and **bandwidth percent** commands (except with strict-priority queues). The weights needed for scheduling are calculated by the CBWFQ process based on the configured bandwidth or its percentage.



# Available Bandwidth

The available bandwidth displayed by the **show interface** command is calculated by subtracting all fixed bandwidth reservations from the default 75 percent of the configured bandwidth of an interface.



The available bandwidth is calculated with the following formula:

$$\text{Bandwidth}_{\text{avail}} = \text{Bandwidth} * \text{MaxReservable} - \text{SUM}(\text{all fixed guarantees})$$

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). The resulting sum represents the minimum bandwidth requirement for any given link, and it should not exceed 75 percent of the total available bandwidth for the link. The remaining 25 percent is used for other overhead, including Layer 2 overhead, routing traffic, and best-effort traffic. However, under aggressive circumstances in which you want to configure more than 75 percent of the interface bandwidth to classes, you can override the 75 percent maximum sum allocated to all classes or flows using the **max-reserved-bandwidth** command. If you want to override the default 75 percent, exercise caution and ensure that you allow enough remaining bandwidth to support best-effort and control traffic and Layer 2 overhead such as Layer 2 keepalive messages, as well as the class default traffic.

If all of the bandwidth is not allocated, the remaining bandwidth is proportionately allocated among the classes based on the configured bandwidth of the classes.

## CBWFQ Benefits and Drawbacks

CBWFQ allows you to define traffic classes based on custom-defined match criteria such as ACLs, input interfaces, and protocol type.

CBWFQ Benefits and Drawbacks	
Benefits	<ul style="list-style-type: none"><li>• Custom-defined classifications</li><li>• Minimum bandwidth allocation</li><li>• Finer granularity and scalability</li></ul>
Drawback	<ul style="list-style-type: none"><li>• Voice traffic can still suffer unacceptable delay.</li></ul>

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—4-11

The benefits of CBWFQ include these:

- **Classification:** CBWFQ allows custom-defined classifications based on many parameters, such as ACLs, input interfaces, byte count, and so on.
- **Bandwidth allocation:** CBWFQ allows you to specify the exact minimum 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 the flow-based WFQ. Flow-based WFQ applies weights to traffic to classify it into conversations and determines how much bandwidth each conversation is allowed, relative to other conversations. For flow-based WFQ, these weights, and traffic classification, are dependent on and limited to IP precedence levels.
- **Finer granularity and scalability:** CBWFQ allows you to define what constitutes a class based on criteria that exceed the confines of flow. You need not maintain traffic classification on a per-flow basis. Moreover, you can configure up to 64 discrete classes in a service policy.

The drawback is that voice traffic can still suffer from unacceptable delays if CBWFQ is used as the only queuing mechanism.

# Configuring and Monitoring CBWFQ

This topic describes the Cisco IOS commands that are used to configure and monitor CBWFQ on a Cisco router.

## Configuring CBWFQ

```
router(config-pmap-c) #
```

```
bandwidth bandwidth
```

- Allocates a fixed amount of bandwidth to a class.
- Sets the value in kilobits per second.

```
router(config-pmap-c) #
```

```
bandwidth percent percent
```

- Allocates a percentage of bandwidth to a class.
- The configured (or default) interface bandwidth is used to calculate the guaranteed bandwidth.

```
router(config-pmap-c) #
```

```
bandwidth remaining percent percent
```

- Allocates a percentage of available bandwidth to a class.

© 2006 Cisco Systems, Inc. All rights reserved.ONT v1.0—4-13

The **bandwidth** command within the **policy-map class** configuration command is used to specify or modify the bandwidth allocated for a class belonging to a policy map.

All classes belonging to one policy map should use the same type of bandwidth guarantee, in kilobits per second, percentage of interface bandwidth, or percentage of available bandwidth.

Configuring bandwidth in percentages is most useful when the underlying link bandwidth is unknown or the relative class bandwidth distributions are known.

**bandwidth** {*bandwidth* | **percent** *percent* | **remaining** *percent percent*}

### bandwidth Parameters

Parameter	Description
<i>bandwidth</i>	Amount of bandwidth, in kilobits per second, to be assigned to the class.
<i>percent percent</i>	Amount of guaranteed bandwidth, based on an absolute percentage of available bandwidth. The percentage can be a number from 1 to 100. (By default, only 75 percent can be reserved.)
<i>remaining percent percent</i>	Amount of guaranteed bandwidth, based on a relative percentage of available bandwidth. The percentage can be a number from 1 to 100.

These restrictions apply to the **bandwidth** command:

- If the **percent** keyword is used, the sum of the class bandwidth percentages cannot exceed 100 percent.
- The amount of bandwidth configured should be large enough to accommodate Layer 2 overhead.
- A policy map can have all the class bandwidths specified in kilobits per second or in percentages but not a mix of both. However, the unit for the **priority** command in the priority class can be different from the bandwidth unit of the low-priority class.

## Configuring CBWFQ (Cont.)

```
router(config-pmap-c) #
```

```
queue-limit queue-limit
```

- Sets the maximum number of packets that this queue can hold.
- The default maximum is 64.

```
router(config-pmap-c) #
```

```
fair-queue [number-of-dynamic-queues]
```

- The class-default class can be configured to use WFQ.
- The number of dynamic queues is a power of 2 in the range from 16 to 4096, specifying the number of dynamic queues.

© 2006 Cisco Systems, Inc. All rights reserved.

QNT v1.0—4-14

The default queue limit of 64 packets can be changed using the **queue-limit** command. It is recommended that you not change the default value.

The default class can be selected by specifying the **class-default** name of the class. The default class supports two types of queuing: a FIFO queue (default) or a flow-based WFQ system. Both types can be combined with WRED. A FIFO queue can also get a minimum bandwidth guarantee.

### Example: Configuration of FIFO Queuing for the Class-Default

This example shows the configuration of FIFO queuing within the default class. The default class is also guaranteed 1 Mbps of bandwidth, and the maximum queue size is limited to 40 packets.

```
policy-map A
class A
bandwidth 1000
class class-default
bandwidth 1000
queue-limit 40
```

### Example: Configuration of WFQ Queuing for the Class-Default

The following example shows the configuration of WFQ queuing within the default class. The number of dynamic queues is set to 1024, and the discard threshold is set to 50.

```
policy-map A
class A
bandwidth 1000
class class-default
fair-queue 1024
queue-limit 50
```

## Example of CBWFQ

The sample configuration shows how CBWFQ is used to guarantee bandwidth to each of the two classes.

### Example of CBWFQ

```
Router(config)#access-list 101 permit udp host 10.10.10.10 host
10.10.10.20 range 16384 20000
Router(config-if)#access-list 102 permit udp host 10.10.10.10 host
10.10.10.20 range 53000 56000
Router(config)#class-map class1
Router(config-cmap)#match access-group 101
Router(config-cmap)#exit
Router(config)#class-map class2
Router(config-cmap)#match access-group 102
Router(config-cmap)#exit
Router(config)#policy-map policy1
Router(config-pmap)#class class1
Router(config-pmap-c)#bandwidth 3000
Router(config-pmap-c)#queue-limit 30
Router(config-pmap-c)#exit
Router(config-pmap)#class class2
Router(config-pmap-c)#bandwidth 2000
Router(config-pmap-c)#exit
Router(config-pmap)#class class-default
Router(config-pmap-c)#fair-queue
Router(config-pmap-c)#exit
```

# Monitoring CBWFQ

The **show policy-map interface** command displays all service policies applied to the interface.

## Monitoring CBWFQ

```
router>
show policy-map interface [interface]
• Displays parameters and statistics of CBWFQ
Router#show policy-map interface
FastEthernet0/0

Service-policy output: policy1

Class-map: class1 (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: access-group 101
  Queueing
    Output Queue: Conversation 265
    Bandwidth 3000 (kbps) Max Threshold 30 (packets)
    (pkts matched/bytes matched) 0/0
    (depth/total drops/no-buffer drops) 0/0/0
<...part of the output omitted...>
Class-map: class-default (match-any)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: any
  Queueing
    Flow Based Fair Queueing
<...rest of the output omitted...>
```

© 2006 Cisco Systems, Inc. All rights reserved. CNT v1.0—4-16

In the classification, policing parameters, queuing mechanism, bandwidth and statistics are displayed. This policy is applied on the FastEthernet 0/0 interface. The class1 class map classifies any traffic matched by ACL 101 as CBWFQ. Also, bandwidth of 3000 kbps is displayed. Traffic not matching the configured classification (class1 and class2) will be placed in the class-default class map.

The complete output of this command is as follows:

```
Router#show policy-map interface
FastEthernet0/0

Service-policy output: policy1

Class-map: class1 (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: access-group 101
  Queueing
    Output Queue: Conversation 265
    Bandwidth 3000 (kbps) Max Threshold 30 (packets)
    (pkts matched/bytes matched) 0/0
    (depth/total drops/no-buffer drops) 0/0/0

Class-map: class2 (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: access-group 102
  Queueing
    Output Queue: Conversation 266
    Bandwidth 2000 (kbps) Max Threshold 64 (packets)
    (pkts matched/bytes matched) 0/0
    (depth/total drops/no-buffer drops) 0/0/0

Class-map: class-default (match-any)
```

0 packets, 0 bytes  
5 minute offered rate 0 bps, drop rate 0 bps  
Match: any  
Queueing  
  Flow Based Fair Queueing  
  Maximum Number of Hashed Queues 256  
  (total queued/total drops/no-buffer drops) 0/0/0



# Low Latency Queuing

This topic describes the purpose and features of LLQ.

## Low Latency Queuing

- A priority queue is added to CBWFQ for real-time traffic.
- High-priority classes are guaranteed:
  - Low-latency propagation of packets
  - Bandwidth
- High-priority classes are also policed when congestion occurs—they then cannot exceed their guaranteed bandwidth.
- Lower-priority classes use CBWFQ.

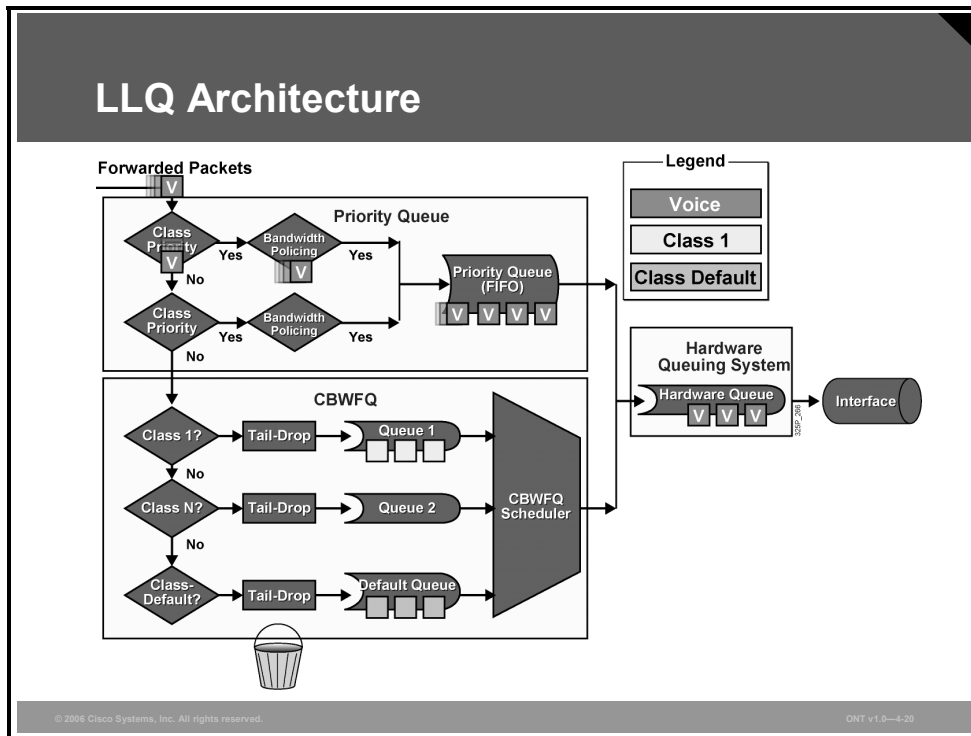
Although WFQ provides a fair share of bandwidth to every flow and provides fair scheduling of its queues, it cannot provide guaranteed bandwidth and low delay to selected applications. For example, voice traffic may still compete with other aggressive flows in the WFQ queuing system because the WFQ system lacks priority scheduling for time-critical traffic classes.

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

LLQ reduces the jitter in voice conversations. To enqueue real-time traffic to a strict-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, strict priority queue.

# LLQ Architecture and Benefits

This topic describes how LLQ works and identifies situations in which LLQ is most appropriate for providing quality of service (QoS).



LLQ extends CBWFQ by adding strict-priority queuing. Strict-priority queuing allows delay-sensitive data such as voice to be dequeued and sent first. Voice packets that enter the LLQ system are sent to the priority queue part of the LLQ system, where they have a fixed bandwidth allocation and where they are served first. Data packets enter the CBWFQ system directly, where they are treated according to the CBWFQ assigned weights.

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

# LLQ Benefits

One benefit of LLQ is having a consistent configuration across all media types, irrespective of the media used.

## LLQ Benefits

- **High-priority classes are guaranteed:**
  - **Low-latency propagation of packets**
  - **Bandwidth**
- **Configuration and operation are consistent across all media types.**
- **Entrance criteria to a class can be defined by an ACL.**
  - **Not limited to UDP ports as with IP RTP priority**
  - **Defines trust boundary to ensure simple classification and entry to a queue**

© 2006 Cisco Systems, Inc. All rights reserved. QRT v1.0-4.21

Also, with LLQ the entrance criteria for a class can be as granular as you like because you define it by an ACL. You are not limited, as with the IP RTP Priority feature, to a simple UDP port range. If the port range feature had not been changed, it is probable that every future voice application would take advantage of it, knowing that the application would get preferential treatment within the Cisco infrastructure.

# Configuring and Monitoring LLQ

This topic describes the Cisco IOS commands that are used to configure and monitor LLQ on a Cisco router.

## Configuring LLQ

```
router(config-pmap-c)#
```

```
priority bandwidth [burst]
```

- Allocates a fixed amount of bandwidth (in kilobits per second) to a class and ensures expedited forwarding.
- Traffic exceeding the specified bandwidth is dropped if congestion exists; otherwise, policing is not used.

```
router(config-pmap-c)#
```

```
priority percent percentage [burst]
```

- Allocates a percentage of configured or default interface bandwidth to a class and ensures expedited forwarding.
- Traffic exceeding the specified bandwidth is dropped if congestion exists.

© 2006 Cisco Systems, Inc. All rights reserved.ONT v1.0—4-23

When you specify the **priority** command for a class, you can use the *bandwidth* argument to specify the maximum bandwidth in kilobits per second. You use this parameter to specify the maximum amount of bandwidth allocated for packets belonging to the class configured with the **priority** command. The *bandwidth* parameter both guarantees bandwidth to the priority class and restrains the flow of packets from the priority class.

**priority** {*bandwidth* | **percent percentage**} [**burst**]

### priority Parameters

Parameter	Description
<i>bandwidth</i>	Guaranteed allowed bandwidth, in kilobits per second, for the priority traffic. The amount of guaranteed bandwidth varies according to the interface and platform in use. Beyond the guaranteed bandwidth, the priority traffic will be dropped in the event of congestion to ensure that the nonpriority traffic is not starved.
<i>percent</i>	Specifies that the amount of guaranteed bandwidth will be a percentage of available bandwidth.
<i>percentage</i>	Used in conjunction with the <b>percent</b> keyword; specifies the percentage of the total available bandwidth to be set aside for the priority class. The percentage can be a number from 1 to 100. (By default, only 75 percent can be reserved.)
<i>burst</i>	(Optional) Specifies the burst size, in bytes. The range of the burst size is 32 to 2,000,000 bytes. The burst size allows temporary bursting of traffic above the maximum limit after a period of inactivity. The default burst value is computed as 200 ms of traffic at the configured bandwidth rate.

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

Priority traffic metering has the following qualities:

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

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

Keep the following guidelines in mind when using the **priority** command:

- Layer 2 encapsulations are accounted for in the amount of bandwidth specified with the **priority** command. However, ensure that a bandwidth allocation is configured with room for the Layer 2 overhead.
- Use the **priority** command for VoIP on serial links and ATM PVCs.

## Configuring LLQ (Cont.)

```
class-map voip
  match ip precedence 5
!
class-map mission-critical
  match ip precedence 3 4
!
class-map transactional
  match ip precedence 1 2
!
policy-map Policy1
  class voip
    priority percent 10
  class mission-critical
    bandwidth percent 30
  class transactional
    bandwidth percent 20
  class class-default
    fair-queue
```

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—4-24

This figure shows configuration example where the VoIP traffic class, classified based on the IP precedence of 5, is queued in the LLQ priority queue. The priority class is guaranteed but is also limited to 10 percent of interface bandwidth.

# Monitoring LLQ

The **show policy-map interface** command displays the packet statistics of all classes that are configured for all service policies on the specified interface.

## Monitoring LLQ

```
router>
```

```
show policy-map interface interface
```

- Displays the packet statistics of all classes that are configured for all service policies on the specified interface or subinterface

```
router>show policy-map interface fastethernet 0/0
FastEthernet0/0

Service-policy output: LLQ

Class-map: LLQ (match-any)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: any
  Weighted Fair Queuing
  Strict Priority
  Output Queue: Conversation 264
  Bandwidth 1000 (kbps) Burst 25000 (Bytes)
  (pkts matched/bytes matched) 0/0
  (total drops/bytes drops) 0/0

Class-map: class-default (match-any)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: any
```

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0--4.25

Some of the key fields in the command output are described in the table.

## Key Fields in the show policy-map interface Command Output

Field	Description
Class-map	Class of traffic being displayed. Output is displayed for each configured class in the policy.
offered rate	Rate, in kilobits per second, of packets entering the class.
drop rate	Rate, in kilobits per second, at which packets are dropped from the class. The drop rate is calculated by subtracting the number of successfully transmitted packets from the offered rate.
Match	Match criteria specified for the class of traffic.
pkts matched/bytes matched	Number of packets (shown in bytes) matching this class that were placed in the queue.
depth/total drops/no-buffer drops	Number of packets, in bytes, discarded for this class. "No-buffer" indicates that no memory buffer exists to service the packet.

# Summary

This topic summarizes the key points that were discussed in this lesson.

## Summary

- **Basic queuing mechanisms can be used to build more advanced queuing mechanisms such as CBWFQ and LLQ.**
- **CBWFQ is a mechanism that is used to overcome the deficiencies of WFQ.**
- **CBWFQ extends the standard WFQ functionality to provide support for traffic classes. Classes are based on user-defined match criteria.**
- **CBWFQ provides a minimum bandwidth guarantee according to traffic classes.**

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—4-26

## Summary (Cont.)

- **LLQ is implemented within CBWFQ by the addition of a priority queue that is serviced using a strict-priority scheduler for time-sensitive traffic such as voice and video.**
- **The LLQ scheduler guarantees both low latency and bandwidth for the traffic in the priority queue.**
- **In the event of congestion, if the priority-queue traffic exceeds the bandwidth guarantee, a congestion-aware policer is used to drop the exceeds traffic.**

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—4-27



# Introducing Congestion Avoidance

---

## Overview

TCP supports traffic-management mechanisms such as slow start and fast retransmit. When congestion occurs, TCP traffic tail drop can cause TCP global synchronization, resulting in poor bandwidth use. This lesson will explain how TCP, in conjunction with random early detection (RED), weighted RED (WRED), and class-based WRED (CBWRED), manages the traffic flow between two hosts, and it will then explain the effects of tail drop on the TCP traffic.

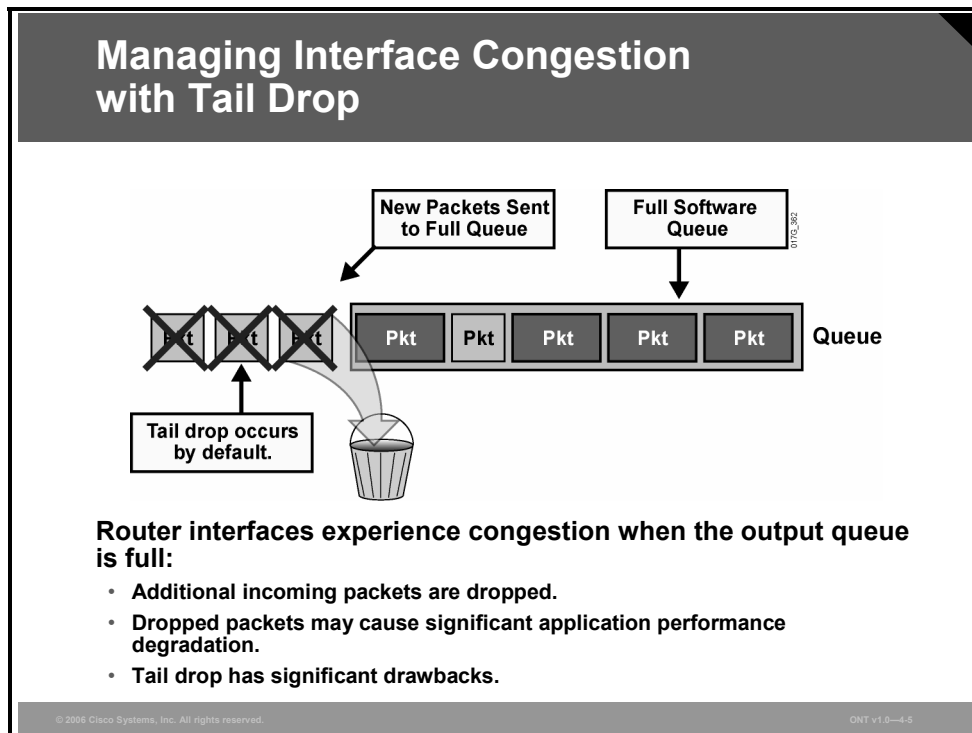
## Objectives

Upon completing this lesson, you will be able to explain Cisco CBWRED operations and basic configurations. This ability includes being able to meet these objectives:

- Describe the default mechanism for managing interface congestion with tail drop
- Describe the limitations of using tail drop as a congestion-management mechanism
- Describe RED and how it can be used to prevent congestion
- Describe WRED and how it can be used to prevent congestion
- Describe the traffic profiles that are used in WRED implementations
- Identify the Cisco IOS software commands that are required to configure CBWRED
- Identify the Cisco IOS software commands that are used to monitor CBWRED

# Managing Interface Congestion with Tail Drop

This topic describes the default mechanism for managing interface congestion, tail drop.



When an interface on a router cannot transmit a packet immediately, the packet is queued, either in an interface transmit (Tx) ring or the interface output hold queue, depending on the switching path that is used. Packets are then taken out of the queue and eventually transmitted on the interface.

If the arrival rate of packets to the output interface exceeds the ability of the router to buffer and forward traffic, the queues increase to their maximum length and the interface becomes congested. Tail drop is the default queuing response to congestion. Tail drop treats all traffic equally and does not differentiate among classes of service. Applications may suffer performance degradation stemming from packet loss caused by tail drop. 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 queue is no longer full.

Weighted fair queuing (WFQ), if configured on an interface, provides a more sophisticated scheme for dropping traffic. WFQ punishes the most aggressive flows using a congestive discard threshold (CDT)-based dropping algorithm.

# Tail Drop Limitations

This topic describes the limitations of using tail drop as a congestion-management mechanism.

## Tail Drop Limitations

**Tail drop should be avoided because it contains significant flaws:**

- **TCP synchronization**
- **TCP starvation**
- **No differentiated drop**

© 2006 Cisco Systems, Inc. All rights reserved. OMT v1.0—4-7

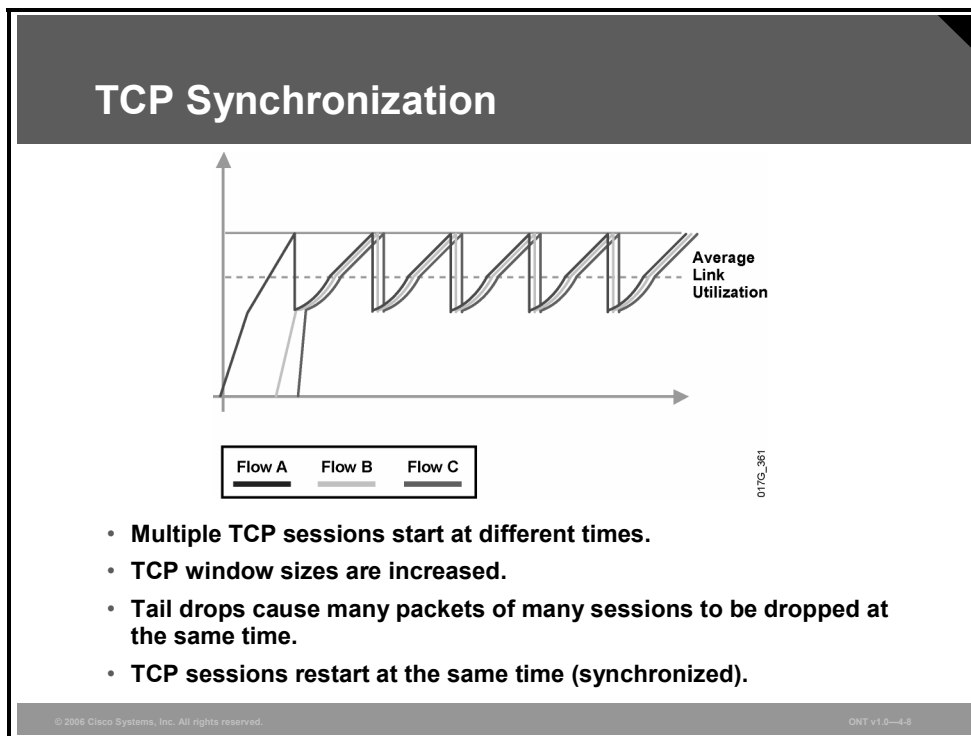
The simple tail-drop scheme does not work very well in environments with a large number of TCP flows or in environments in which selective dropping is required. Understanding the network interaction between TCP stack intelligence and dropping is required to implement a more efficient and fair dropping scheme, especially in service provider environments.

Tail drop has the following shortcomings:

- Normally, when congestion occurs, dropping affects most of the TCP sessions, which simultaneously back off and then restart again. This process causes inefficient link utilization at the congestion point (TCP global synchronization). To avoid dropping of TCP packets, TCP reduces the window size.
- TCP starvation occurs, in which all buffers are temporarily seized by aggressive flows, and normal TCP flows experience buffer starvation.
- There is no differentiated drop mechanism, and therefore higher-priority traffic is dropped in the same way as best-effort traffic.

# TCP Synchronization

A router can handle multiple concurrent TCP sessions. It is likely that when traffic exceeds the queue limit, it exceeds this limit because of the bursty nature of packet networks.

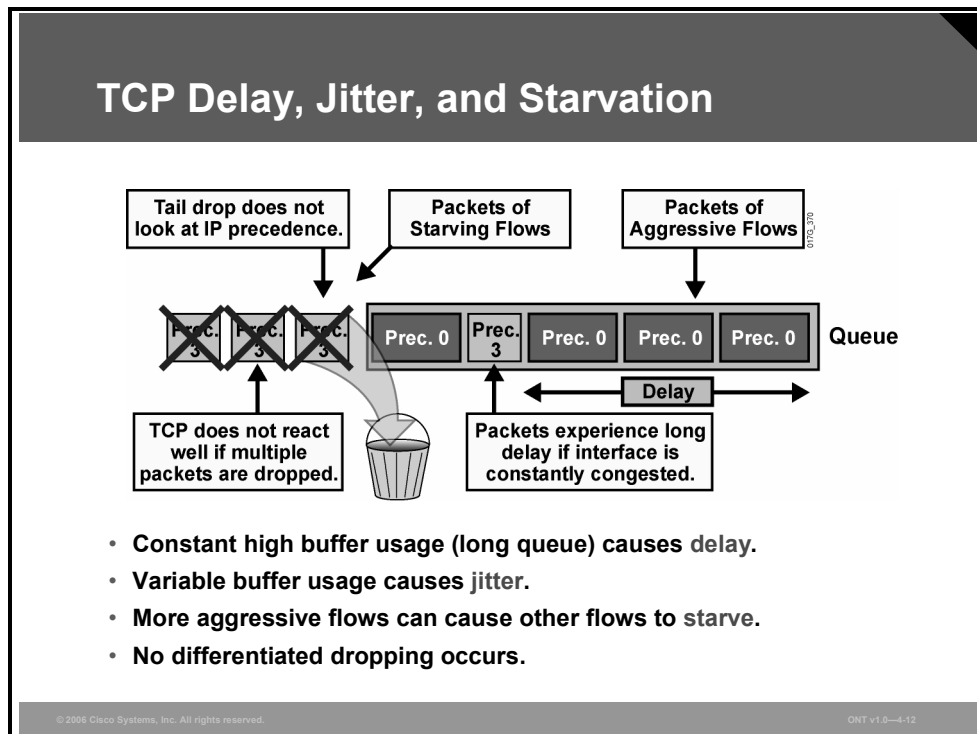


If the receiving router drops all traffic that exceeds the queue limit, as is done by default (with tail drop), many TCP sessions then simultaneously go into slow start. Traffic temporarily slows down to the extreme, and then all flows go into slow start again. This activity creates a condition called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

Global synchronization occurs as waves of congestion crest, only to be followed by troughs during which the transmission link is not fully used. Global synchronization of TCP hosts can occur because packets are dropped all at once. Global synchronization occurs when multiple TCP hosts reduce their transmission rates in response to packet dropping. When congestion is reduced, their transmission rates are increased. The waves of transmission known as global synchronization result in significant link underutilization.

# TCP Delay, Jitter, and Starvation

During periods of congestion, packets are queued up to the full queue length, which may cause increased delay for packets already in the queue. In addition, queuing introduces unequal delays for packets of the same flow, resulting in jitter.



Another TCP-related phenomenon that reduces optimal throughput of network applications is TCP starvation. When multiple flows are being transmitted through a router, some of these flows may be much more aggressive than other flows. For instance, when the TCP transmit window increases for file-transfer applications, the TCP session can send a number of large packets to its destination. These packets immediately fill the queue on the router, and other, less aggressive flows can be starved because there is no differentiated treatment indicating which packets should be dropped. As a result, less aggressive flows are dropped at the output interface.

Tail drop is not the optimal mechanism for congestion avoidance and therefore should not be used. Instead, more intelligent congestion avoidance mechanisms should be used that are capable of slowing traffic before congestion occurs.

# Random Early Detection

This topic describes RED and how it can be used to prevent congestion.

## Random Early Detection

- **Tail drop can be avoided if congestion is prevented.**
- **RED is a mechanism that randomly drops packets before a queue is full.**
- **RED increases drop rate as the average queue size increases.**
- **RED result:**
  - **TCP sessions slow to the approximate rate of output-link bandwidth.**
  - **Average queue size is small (much less than the maximum queue size).**
  - **TCP sessions are desynchronized by random drops.**

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—4-14

Random early detection (RED) is a dropping mechanism that randomly drops packets before a queue is full. The dropping strategy is based primarily on the average queue length—that is, when the average size of the queue increases, RED is more likely to drop an incoming packet than when the average queue length is shorter.

Because RED drops packets randomly, it has no per-flow intelligence. The rationale is that an aggressive flow will represent most of the arriving traffic, and it is likely that RED will drop a packet of an aggressive session. RED therefore punishes more aggressive sessions with a higher statistical probability and is able to somewhat selectively slow the most significant cause of congestion. Directing one TCP session at a time to slow down allows for full utilization of the bandwidth rather than utilization that manifests itself as crests and troughs of traffic.

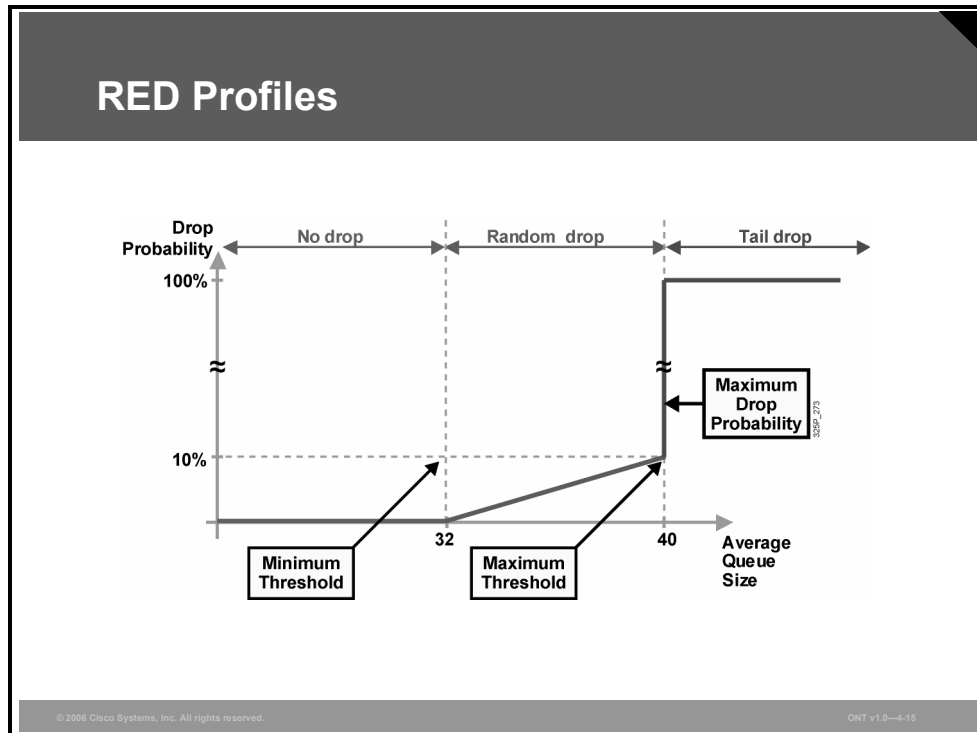
As a result of implementing RED, TCP global synchronization is much less likely to occur, and TCP can utilize link bandwidth more efficiently. In RED implementations, the average queue size also decreases significantly, because the possibility of the queue filling up is reduced. This is because of very aggressive dropping in the event of traffic bursts, when the queue is already quite full.

RED distributes losses over time and normally maintains a low queue depth while absorbing traffic spikes. RED can also utilize IP precedence or differentiated services code point (DSCP) bits in packets to establish different drop profiles for different classes of traffic.

RED is useful only when the bulk of the traffic is TCP traffic. With TCP, dropped packets indicate congestion, so the packet source reduces its transmission rate. With other protocols, packet sources might not respond or might re-send dropped packets at the same rate, and so dropping packets might not decrease congestion.

# RED Profiles

A RED traffic profile is used to determine the packet-dropping strategy and is based on the average queue length.



The probability of a packet being dropped is based on three configurable parameters contained within the RED profile:

- **Minimum threshold:** When the average queue length 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.
- **Mark probability denominator:** This number is the fraction of packets that are 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. The linear increase of packet drops from the minimum threshold (0 drops) to the maximum threshold is based on this parameter and the queue size between the minimum and maximum thresholds.

The minimum threshold value should be set high enough to maximize link utilization. If the minimum threshold is too low, packets may be dropped unnecessarily, and the transmission link will not be fully used.

The difference between the maximum threshold and the minimum threshold should be large enough to avoid global synchronization. If the difference is too small, many packets may be dropped at once, resulting in global synchronization.

# RED Modes

This section describes the three RED modes used in TCP.

## RED Modes

- **RED has three modes:**
  - **No drop:** When the average queue size is between 0 and the minimum threshold
  - **Random drop:** When the average queue size is between the minimum and the maximum threshold
  - **Full drop (tail drop):** When the average queue size is above the maximum threshold
- **Random drop should prevent congestion (prevent tail drops).**

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—4-16

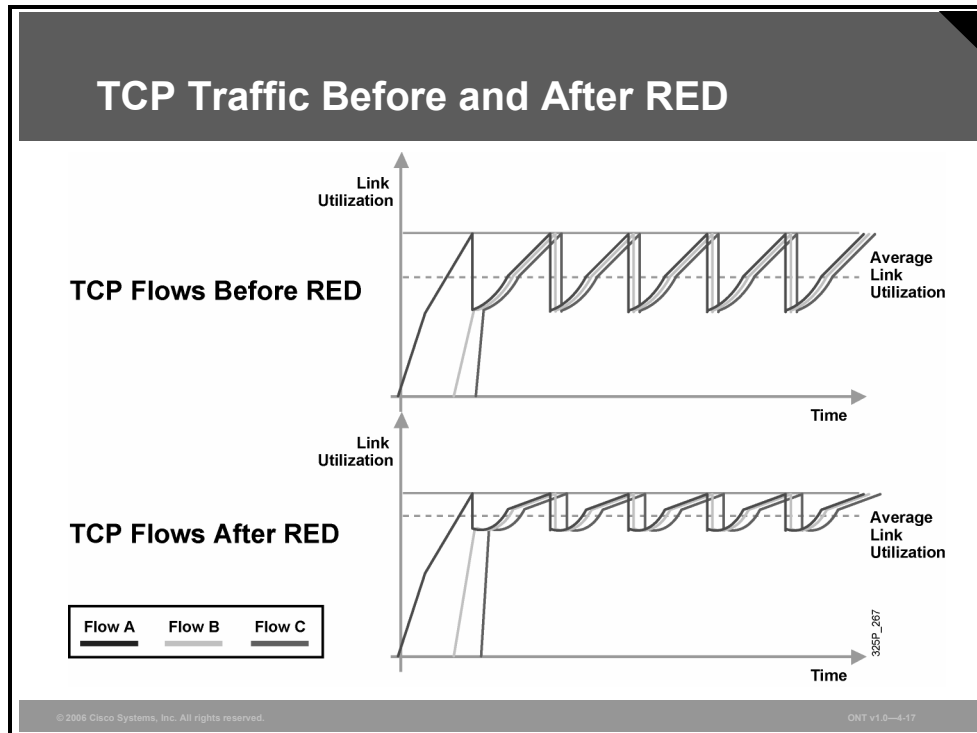
Based on the average queue size, RED has three dropping modes:

- When the average queue size is between 0 and the configured minimum threshold, no drops occur and all packets are queued.
- When the average queue size is between the configured minimum threshold and the configured maximum threshold, random drops occur, which is linearly proportional to the mark probability denominator and the average queue length.
- When the average queue size is at or higher than the maximum threshold, RED performs full (tail) drop in the queue. This situation is unlikely, because RED should slow down TCP traffic ahead of congestion. If a lot of non-TCP traffic is present, RED cannot effectively drop traffic to reduce congestion, and tail drops are likely to occur.



## TCP Traffic Before and After RED

This figure shows TCP throughput behavior compared to link bandwidth in a congested network scenario where the tail-drop and RED mechanisms are in use on the link.



Before RED, when all sessions slow down, congestion on the router interface is removed and all TCP sessions restart their transmission at about the same time. Again, the router interface quickly becomes congested, causing tail drop. As a result, all TCP sessions back off again. This behavior cycles constantly, resulting in a link that is generally underutilized.

After RED is applied, RED randomly drops packets, influencing a small number of sessions at a time, before the interface reaches congestion. Overall throughput of sessions is increased, as is average link utilization. Global synchronization is very unlikely to occur, because of selective, but random, dropping of adaptive traffic.

# Weighted Random Early Detection

This topic describes WRED and how it can be used to prevent congestion.

## Weighted Random Early Detection

- **WRED can use multiple different RED profiles.**
- **Each profile is identified by:**
  - Minimum threshold
  - Maximum threshold
  - Mark probability denominator
- **WRED profile selection is based on:**
  - IP precedence (8 profiles)
  - DSCP (64 profiles)
- **WRED drops less important packets more aggressively than more important packets.**
- **WRED can be applied at the interface, VC, or class level.**

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—4-19

Weighted random early detection (WRED) combines RED with IP precedence or DSCP and performs packet dropping based on IP precedence or DSCP markings.

As with RED, WRED monitors the average queue length in the router and determines when to begin discarding packets based on the length of the interface queue. When the average queue length exceeds the user-specified minimum threshold, WRED begins to randomly drop packets with a certain probability. If the average length of the queue continues to increase so that it becomes larger than the user-specified maximum threshold, WRED reverts to a tail-drop packet-discard strategy, in which all incoming packets are dropped.

The idea behind using WRED is to maintain the queue length at a level somewhere below the maximum threshold and to implement different drop policies for different classes of traffic. WRED can selectively discard lower-priority traffic when the interface becomes congested and can provide differentiated performance characteristics for different classes of service. WRED can also be configured to produce nonweighted RED behavior.

For interfaces configured to use Resource Reservation Protocol (RSVP), WRED chooses packets from other flows to drop rather than the RSVP flows. Also, IP precedence or DSCP helps determine which packets are dropped, because traffic at a lower priority has a higher drop rate than traffic at a higher priority (and, therefore, lower-priority traffic is more likely to be throttled back). In addition, WRED statistically drops more packets from large users than from small users. The traffic sources that generate the most traffic are more likely to be slowed down than traffic sources that generate little traffic.

WRED reduces the chances of tail drop by selectively dropping packets when the output interface begins to show signs of congestion. By dropping some packets early rather than waiting until the queue is full, WRED avoids dropping large numbers of packets at once and minimizes the chances of global synchronization. As a result, WRED helps maximize the utilization of transmission lines.

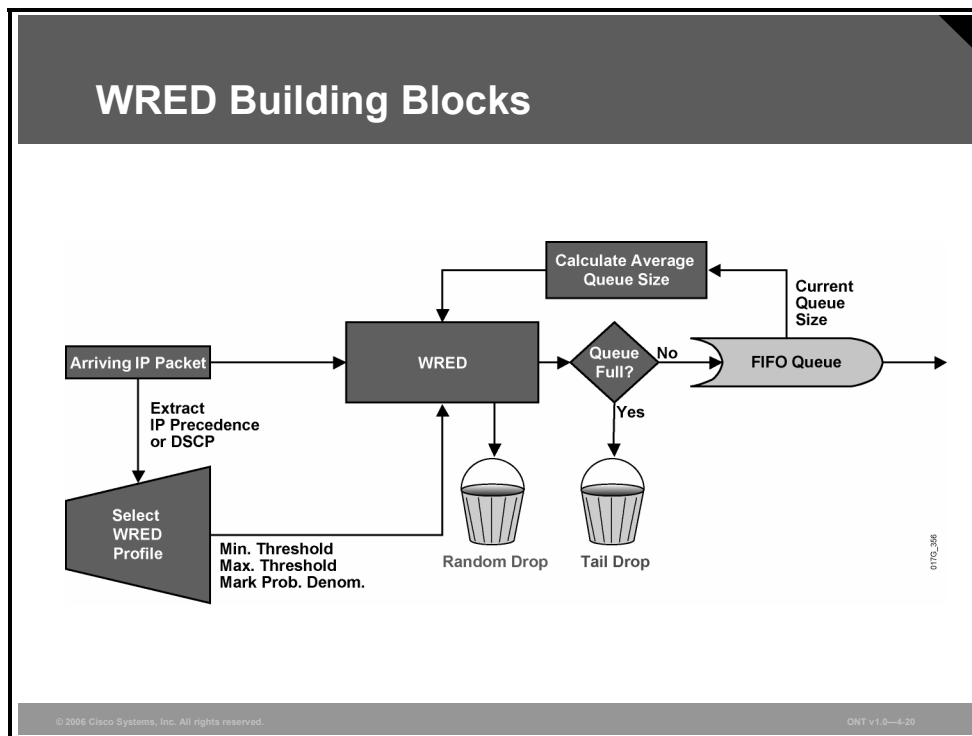
WRED treats non-IP traffic as precedence 0, the lowest precedence. Therefore, non-IP traffic, in general, is more likely to be dropped than IP traffic.

WRED should be used wherever there is a potential congested link (bottleneck), which could very well be an access or edge link. However, WRED is normally used in the core routers of a network rather than at the network edge. Edge routers assign IP precedence or DSCP to packets as they enter the network. WRED uses these assigned values to determine how to treat different types of traffic.

Note that WRED is not recommended for any voice queue, although WRED may be enabled on an interface carrying voice traffic. WRED will not throttle back voice traffic because voice traffic is User Datagram Protocol (UDP)-based. The network itself should be designed not to lose voice packets because lost voice packets result in reduced voice quality. WRED controls congestion by affecting prioritized traffic other than voice, and avoiding congestion helps to ensure voice quality.

# WRED Building Blocks

This figure shows how WRED is implemented and the parameters that are used by WRED to influence packet-drop decisions.



The router constantly updates the WRED algorithm with the calculated average queue length, which is based on the recent history of queue lengths.

Configured in the traffic profile are the parameters that define the drop characteristics used by WRED (minimum threshold, maximum threshold, and mark probability denominator). These parameters define the WRED probability slopes.

When a packet arrives at the output queue, the IP precedence or DSCP value is used to select the correct WRED profile for the packet. The packet is then passed to WRED for processing. Based on the selected traffic profile and the average queue length, WRED calculates the probability for dropping the current packet and either drops the packet or passes it to the output queue.

If the queue is already full, the packet is dropped. Otherwise, the packet is eventually transmitted out to the interface. If the average queue length is greater than the minimum threshold but less than the maximum threshold, based on the drop probability, WRED either queues the packet or performs a random drop.

# Class-Based WRED

Traditionally, Cisco IOS software used stand-alone RED and WRED mechanisms to avoid congestion on an interface. Those mechanisms can perform a differentiated drop based on the IP precedence or DSCP value.

## Class-Based WRED

- **Class-based WRED is available when configured in combination with CBWFQ.**
- **Using CBWFQ with WRED allows the implementation of DiffServ assured forwarding PHB.**
- **Class-based configuration of WRED is identical to stand-alone WRED.**

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—4-21

The class-based weighted fair queuing (CBWFQ) system supports the use of WRED inside the queuing system, thereby implementing class-based WRED (CBWRED). Each class is queued in its separate queue and has a queue limit, performing tail drop by default. WRED can be configured as the preferred dropping method in a queue, implementing a differentiated drop based on traffic class, and further, on the IP precedence or DSCP value.

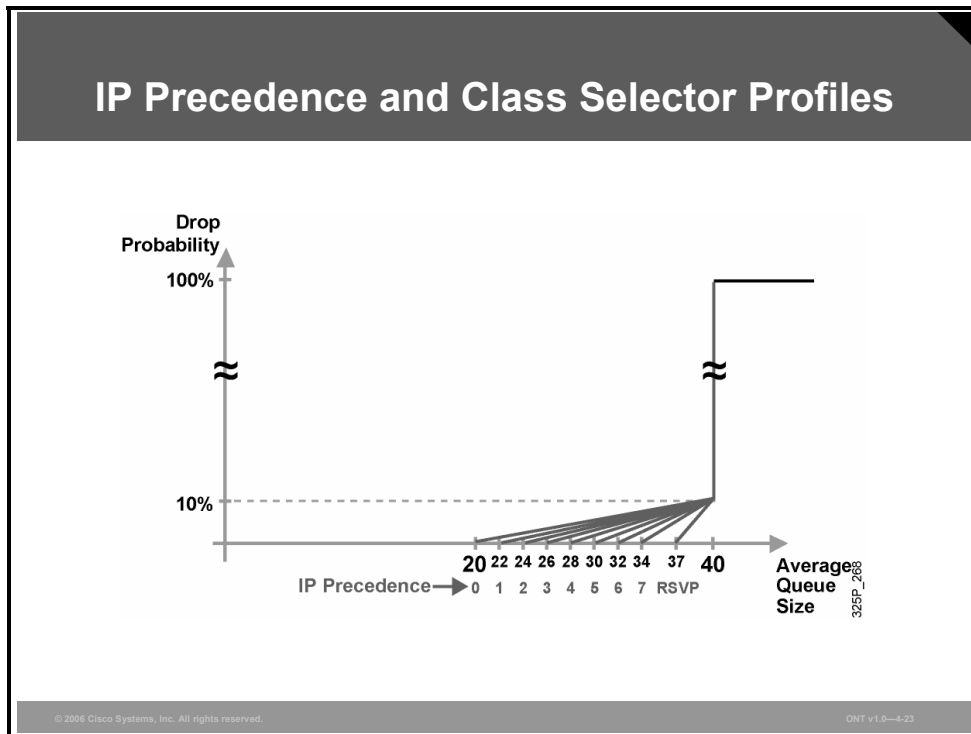
---

<b>Note</b>	The combination of CBWFQ and WRED on a single device is currently the only way to implement the Differentiated Services (DiffServ) assured forwarding (AF) per-hop behavior (PHB) using Cisco IOS software.
-------------	---

---

# WRED Profiles

This topic describes the traffic profiles that are used in WRED implementations.

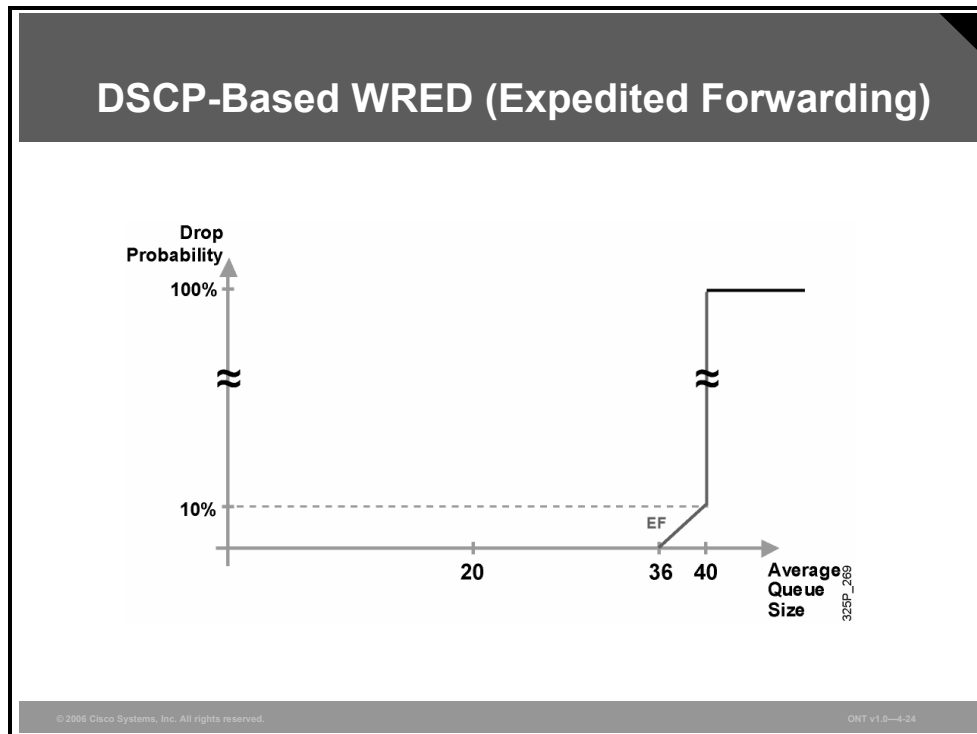


A PHB is the externally observable forwarding behavior applied at a DiffServ-compliant node to a DiffServ behavior aggregate (BA). With the ability of the system to mark packets according to DSCP setting, collections of packets (each with the same DSCP setting and sent in a particular direction) can be grouped into a DiffServ BA. Packets from multiple sources or applications can belong to the same DiffServ BA.

The class selector BA is used for backward compatibility with non-DiffServ-compliant devices (RFC 1812-compliant devices and, optionally, RFC 791-compliant devices). Therefore, the class selector range of DSCP values is used for backward compatibility with IP precedence.

# DSCP-Based WRED (Expedited Forwarding)

This section explains the DSCP-based WRED.



In DSCP, the Expedited Forwarding (EF) PHB is identified based on these parameters:

- A low departure rate is ensured to provide low delay to delay-sensitive applications.
- Bandwidth is guaranteed to prevent starvation of the application if there are multiple applications using EF PHB.
- Bandwidth is policed to prevent starvation of other applications or classes that are not using this PHB.
- Packets requiring EF should be marked with DSCP binary value 101110 (46).

For the EF DiffServ traffic class, WRED configures itself by default so that the minimum threshold is very high; increasing the probability of no drops being applied to that traffic class. It is expected that EF traffic will be dropped very late, compared to other traffic classes, and the EF traffic is therefore prioritized in the event of congestion.

# Configuring CBWRED

This topic describes the Cisco IOS software commands that are required to configure CBWRED.

## Configuring CBWRED

```
router(config-pmap-c)#  
random-detect
```

- Enables IP precedence-based WRED in the selected class within the service policy configuration mode.
- Default service profile is used.
- Command can be used at the interface, perVC (with random-detect-group), or at the class level (service policy).
- Precedence-based WRED is the default mode.
- WRED treats non-IP traffic as precedence 0.

```
policy-map Policy1  
  class mission-critical  
    bandwidth percent 30  
    random-detect  
  class transactional  
    bandwidth percent 20  
    random-detect  
  class class-default  
    fair-queue  
    random-detect
```

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0-4.26

The **random-detect** command is used to enable WRED on an interface. By default, WRED is IP precedence-based and uses eight default WRED profiles, one for each value of IP precedence.

Within the CBWFQ system, WRED is used to perform per-queue dropping within the class queues. Therefore, each class queue has its own WRED method, which can be further weighed based on the IP precedence or DSCP value. Each queue can therefore be configured with a separate drop policy to implement different drop policies for every class of traffic.

WRED treats all non-IP traffic as precedence 0. As a result, non-IP traffic is more likely to be dropped than IP traffic.

WRED cannot be configured on the same interface as custom queuing (CQ), priority queuing (PQ), or WFQ. However, CBWRED can be configured in conjunction with CBWFQ. Restricting nondistributed, non-class-based WRED to only FIFO queuing on an interface is typically not a major issue because WRED is usually applied in the network core, where advanced queuing mechanisms are not typically used. WRED is suited for the network core because WRED has a relatively low performance impact on routers. Furthermore, CBWRED can be used to overcome this limitation by combining WRED with WFQ.



## Changing the WRED Traffic Profile

When WRED is enabled, default values are selected for each traffic profile based on the weight used (IP precedence or DSCP). Network administrators can then modify these default values to match their specific administrative quality of service (QoS) policy goals.

### Changing the WRED Traffic Profile

```
router(config-pmap-c) #
```

```
random-detect precedence precedence min-threshold max-  
threshold mark-prob-denominator
```

- **Changes WRED profile for specified IP precedence value.**
- **Packet drop probability at maximum threshold is:**  
 $1 / \text{mark-prob-denominator}$
- **Nonweighted RED is achieved by using the same WRED profile for all precedence values.**

© 2006 Cisco Systems, Inc. All rights reserved.ONT v1.0—4-27

When you are modifying the default WRED profile for IP precedence, the following values are configurable:

- **Minimum threshold:** When the average queue depth is above the minimum threshold, WRED 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. The size of the hold queue is equivalent to the number of packets that can be held within a queue. The hold-queue length ranges from 0 to 4096, and, therefore, the minimum/maximum threshold range is 1 to 4096.
- **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. The default maximum threshold will reflect the defined hold-queue size. Thus, if the hold queue is changed, the maximum threshold will change.
- **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 10, one out of every 10 packets is dropped when the average queue is at the maximum threshold. The maximum probability of drop at the maximum threshold can be expressed as  $1 / \text{mark-prob-denominator}$ . The maximum drop probability is 10 percent if default settings are used that have a mark probability denominator value of 10. The value of the mark probability can range from 1 to 65,536.

If required, RED can be configured as a special case of WRED, by assigning the same profile to all eight IP precedence values. The default WRED parameter parameters are based on the best available data. It is recommended that these parameters not be changed from their default values unless you have determined that your applications will benefit from the changed values.

## CBWFQ Using IP Precedence with CBWRED: Example

This example shows IP precedence CBWRED in a CBWFQ deployment.

### CBWFQ Using IP Precedence with CBWRED: Example

- Enable CBWFQ to prioritize traffic according to the following requirements:
  - Class **mission-critical** is marked with IP precedence values 3 and 4 (3 is high drop, 4 is low drop) and should get 30% of interface bandwidth.
  - Class **bulk** is marked with IP precedence values 1 and 2 (1 is high drop, 2 is low drop) and should get 20% of interface bandwidth.
  - All other traffic should be per-flow fair-queued.
- Use differentiated WRED to prevent congestion in all three classes.

© 2006 Cisco Systems, Inc. All rights reserved.

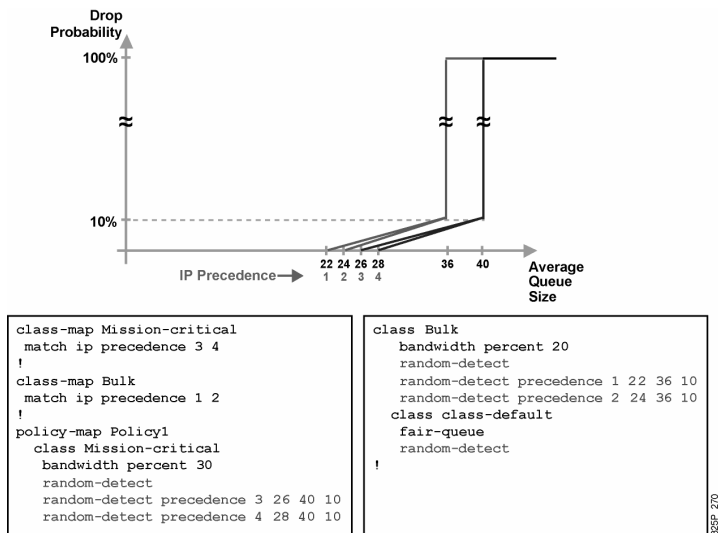
QRT v1.0—4.28

This example of CBWFQ with WRED focuses on a network that provides the following three different service levels for three traffic classes:

- **Mission-critical class:** This class is marked with IP precedence values 3 and 4 (3 is used for high-drop service, and 4 is used for low-drop service within the service class) and should get 30 percent of an interface bandwidth.
- **Bulk class:** This class is marked with IP precedence values 1 and 2 (1 is used for high-drop service, and 2 is used for low-drop service) should get 20 percent of the interface bandwidth.
- **Best-effort class:** This class should get the remaining bandwidth share and should be fair-queued.

To enforce this service policy, a router uses CBWFQ to perform bandwidth sharing and WRED within service classes to perform differentiated drop.

## CBWFQ Using IP Precedence with CBWRED: Example (Cont.)

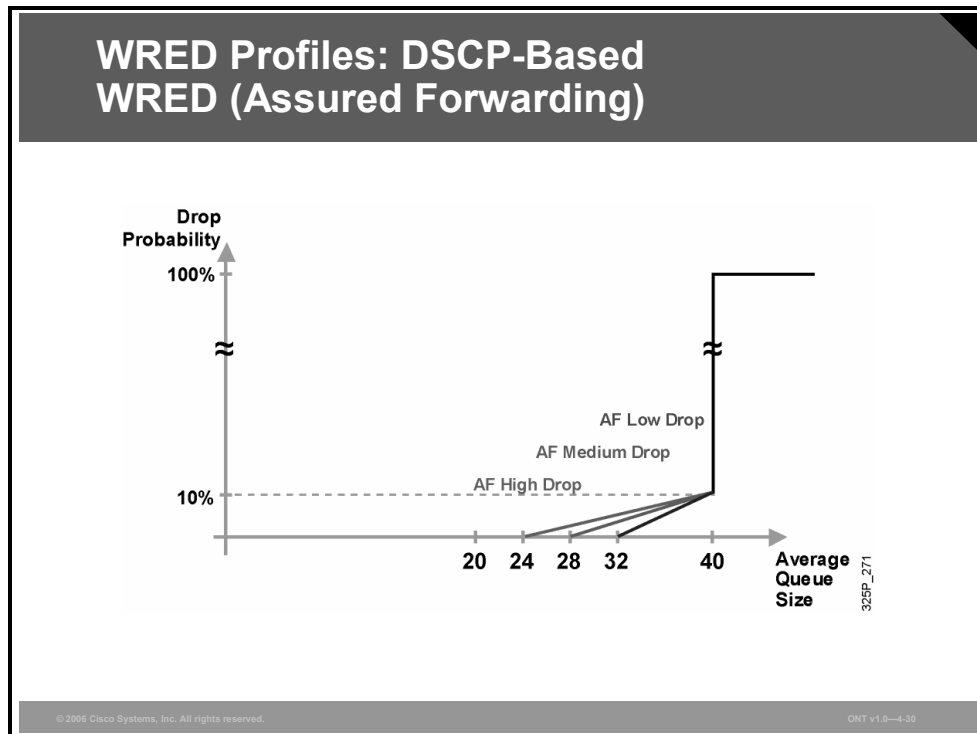


This figure shows the WRED traffic profile representing the QoS service policy and the configuration that is used to implement the example service policy. The traffic is classified based on the IP precedence bits, and all noncontract traffic is classified into the default class:

- The mission-critical class is guaranteed at least 30 percent of bandwidth with a custom WRED profile that establishes a low-drop and a high-drop PHB.
- The bulk class is guaranteed at least 20 percent of bandwidth, is configured with somewhat lower WRED drop thresholds, and is therefore more likely to be dropped than the mission-critical class if interface congestion occurs.
- All other traffic is part of the default class and is fair-queued with default WRED parameters.

## WRED Profiles: DSCP-Based WRED (AF)

This section describes the DSCP-based WRED Assured Forwarding (AF) profiles.



In DSCP, the AF PHB is identified based on the following parameters:

- Guarantees a certain amount of bandwidth to an AF class.
- Allows access to extra bandwidth, if available.

Packets requiring AF PHB should be marked with DSCP value *aaadd0*, where *aaa* is the number of the class and *dd* is the drop probability, or drop preference, of the traffic class.

There are four defined AF classes. Each class should be treated independently and have bandwidth allocated that is based on the QoS policy. For the AF DiffServ traffic class, WRED configures itself by default for three different profiles, depending on the drop preference DSCP marking bits. Therefore, AF traffic should be classified into the three possible classes, such as AF high drop, AF medium drop, and AF low drop. These three classes are based on the sensitivity to packet drops of the application or applications represented by the class. This would mean that the mission-critical class would have an AF low drop, an AF medium drop, and an AF high drop.

## Configuring DSCP-Based CBWRED

The **random-detect dscp-based** command is used to enable DSCP-based WRED on an interface. Changing WRED weighting to values based on DSCP increases the number of WRED traffic profiles to 64.

### Configuring DSCP-Based CBWRED

```
router(config-pmap-c) #  
random-detect dscp-based
```

- **Enables DSCP-based WRED.**
- **Command can be used at the interface, perVC (with random detect group), or at the class level (service policy).**
- **Default service profile is used.**
- **The WRED random-detect command and the WFQ queue-limit command are mutually exclusive for class policy.**

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—4-31

You can configure WRED as part of the policy for a standard class or the default class. The WRED **random-detect** command and the WFQ **queue-limit** command are mutually exclusive for class policy. If you configure WRED, its packet-drop capability is used to manage the queue when packets exceeding the configured maximum count are enqueued. If you configure the WFQ **queue-limit** command for class policy, tail drop is used.

WRED cannot be configured on the same interface as CQ, PQ, or WFQ. However, CB-WRED can be configured in conjunction with CBWFQ. Restricting nondistributed, non-class-based WRED only to FIFO queuing on an interface is not a major issue because WRED is usually applied in the network core, where advanced queuing mechanisms are not typically deployed. WRED is suited for the network core because it has a relatively low performance impact on routers. Further, CBWRED can be used to overcome this limitation by combining WRED with WFQ.

## Changing the WRED Traffic Profile

When DSCP-based WRED is enabled, default values are selected for each traffic profile based on DSCP. Network administrators can then modify these default values to match their specific administrative QoS policy goals.

### Changing the WRED Traffic Profile

```
router(config-pmap-c)#
```

```
random-detect dscp dscpvalue min-threshold max-threshold  
mark-prob-denominator
```

- **Changes WRED profile for specified DSCP value**
- **Packet drop probability at maximum threshold is:**  
 $1 / \text{mark-prob-denominator}$

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—4-32

When you are modifying the default WRED profile for DSCP, the same values described previously for IP precedence are configurable, except that the minimum threshold, the maximum threshold and the mark probability denominator will be applied to DSCP classified packets.

## CBWRED Using DSCP with CBWFQ: Example

This example shows DSCP-based CBWRED in a CBWFQ deployment.

### CBWRED Using DSCP with CBWFQ: Example

- **Enable CBWFQ to prioritize traffic according to the following requirements:**
  - **Class mission-critical is marked using DSCP AF2 and should get 30% of interface bandwidth.**
  - **Class bulk is marked using DSCP AF1 and should get 20% of interface bandwidth.**
  - **All other traffic should be per-flow fair-queued.**
- **Use differentiated WRED to prevent congestion in all three classes.**
- **Make sure that the new configurations still conform to the design and implementation from the previous example.**

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—4-33

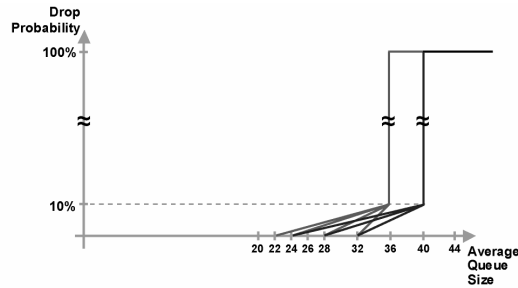
Remember that the DiffServ model itself provides defined traffic classes and their associated PHBs. DiffServ-based classification is used in this example as follows:

- **Mission-critical class:** This class is marked using DSCP AF class 2 and should get 30 percent of an interface bandwidth.
- **Bulk class:** This class is marked using DSCP AF class 1 and should get 20 percent of the interface bandwidth.
- **Best-effort class:** This traffic should get the remaining bandwidth share and should be fair-queued.

To enforce this service policy, a router uses CBWFQ to perform bandwidth sharing and uses WRED within service classes to perform differentiated drop.



## CBWRED Using DSCP with CBWFQ: Example (Cont.)



```
class-map Mission-critical
match ip dscp af21 af22 af23 cs2
!
class-map Bulk
match ip dscp af11 af12 af13 cs1
!
policy-map Policy1
class Mission-critical
bandwidth percent 30
random-detect dscp-based
random-detect dscp af21 32 40 10
random-detect dscp af22 28 40 10
random-detect dscp af23 24 40 10
random-detect dscp cs2 24 40 10
!
class Bulk
bandwidth percent 20
random-detect dscp-based
random-detect dscp af11 32 36 10
random-detect dscp af12 28 36 10
random-detect dscp af13 24 36 10
random-detect dscp cs1 22 36 10
!
class class-default
fair-queue
random-detect dscp-based
```

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—4-34

The configuration example shows how traffic classification is performed using DSCP-based classes, representing the mission-critical class as the AF1 class, and the bulk class as the AF2 class. WRED DSCP-based parameters are set reflecting the class-dependent drop strategy:

- The mission-critical class is guaranteed at least 30 percent of bandwidth, with a custom WRED profile that establishes three different drop probabilities for AF class 2.
- The bulk class is guaranteed at least 20 percent of bandwidth, is configured with three different drop probabilities for AF class 1, and has a somewhat lower WRED maximum threshold. As a result, bulk-class traffic is more likely to be dropped than the mission-critical class if interface congestion occurs.

All other traffic is part of the default class and is fair-queued, with default WRED parameters.

**Note** When you enable WRED with the **random-detect** command, the parameters are set to their default values. The weight factor is 9. For all precedences, the mark probability denominator is 10, and maximum threshold is based on the output buffering capacity and the transmission speed for the interface.

# Monitoring CBWRED

This topic describes the Cisco IOS software commands that are required to monitor CBWRED.

## Monitoring CBWRED

```
router#  
show policy-map interface interface-name
```

- Displays the configuration of all classes configured for all service policies on the specified interface

```
router#show policy-map interface Ethernet 0/0  
Ethernet0/0  
Service-policy output: Policy1  
Class-map: Mission-critical (match-all)  
  0 packets, 0 bytes  5 minute offered rate 0 bps, drop rate 0 bps  
  Match: ip precedence 2  Match: ip dscp 18  20  22  
  Weighted Fair Queueing  
    Output Queue: Conversation 265  
    Bandwidth 30 (%)  Bandwidth 3000 (kbps)  
    (pkts matched/bytes matched) 0/0  
    (depth/total drops/no-buffer drops) 0/0/0  
    exponential weight: 9  
    mean queue depth: 0  
Dscp      Transmitted Random drop Tail drop  Minimum  Maximum  Mark  
(Prec)    pkts/bytes  pkts/bytes  pkts/bytes  threshold threshold probability  
0 (0)      0/0          0/0         0/0         20        40       1/10  
1          0/0          0/0         0/0         22        40       1/10  
2          0/0          0/0         0/0         24        40       1/10
```

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—4-38

The **show policy-map interface** command displays the configuration of all classes configured for all service policies on the specified interface. This includes all WRED parameters implementing the drop policy on the specified interface.

The table explains some of the key fields of the output of the **show policy-map interface** command.

## Key Fields in the show policy-map interface Command Output

Field	Description
Service-policy output	Name of the output service policy applied to the specified interface or virtual circuit (VC).
Class-map	Class of traffic being displayed. Output is displayed for each configured class in the policy. The choice for implementing class matches (for example, match-all or match-any) can also appear next to the traffic class.
Match	Match criteria specified for the class of traffic. Choices include criteria such as IP precedence, IP DSCP value, Multiprotocol Label Switching (MPLS) experimental (EXP) value, access groups, and QoS groups.
exponential weight	Exponent used in the average queue size calculation for a WRED parameter group.
mean queue depth	Average queue depth based on the actual queue depth on the interface and the exponential weighting constant. It is a fluctuating average. The minimum and maximum thresholds are compared against this value to determine drop decisions.

# Summary

This topic summarizes the key points that were discussed in this lesson.

## Summary

- **TCP uses windowing and the TCP slow-start mechanism as its means of controlling congestion.**
- **Tail drop causes significant issues, including TCP synchronization, starvation, and delay. TCP synchronization decreases the average utilization of network links.**
- **RED is a mechanism that randomly drops packets before a queue is full, preventing congestion and avoiding tail drop.**
- **RED operates by increasing the rate at which packets are dropped from queues as the average queue size increases.**

© 2006 Cisco Systems, Inc. All rights reserved.

QNT v1.0—4-37

## Summary (Cont.)

- **RED has three modes of operation: no drop, random drop, and full drop (tail drop).**
- **With RED, TCP global synchronization is eliminated and the average link utilization increases.**
- **WRED combines RED with IP precedence or DSCP and performs packet dropping based on IP precedence or DSCP markings.**
- **Each WRED profile defines the minimum and maximum threshold and the maximum drop probability. Profiles are already defined by default for IP precedence and DSCP.**

© 2006 Cisco Systems, Inc. All rights reserved.

QNT v1.0—4-38



# Introducing Traffic Policing and Shaping

---

## Overview

Traffic policing can be used to control the maximum rate of traffic sent or received on an interface. Traffic policing is often configured on interfaces at the edge of a network to limit traffic into or out of the network. Traffic shaping can be used to control the traffic going out an interface to match its flow to the speed of the remote target interface and to ensure that the traffic conforms to policies contracted for it. Traffic policing and traffic shaping differ in the way that they respond to traffic violations. Policing typically drops excess traffic, while shaping typically queues excess traffic. This lesson describes the traffic-policing and traffic-shaping quality of service (QoS) mechanisms that are used to limit the available bandwidth to traffic classes.

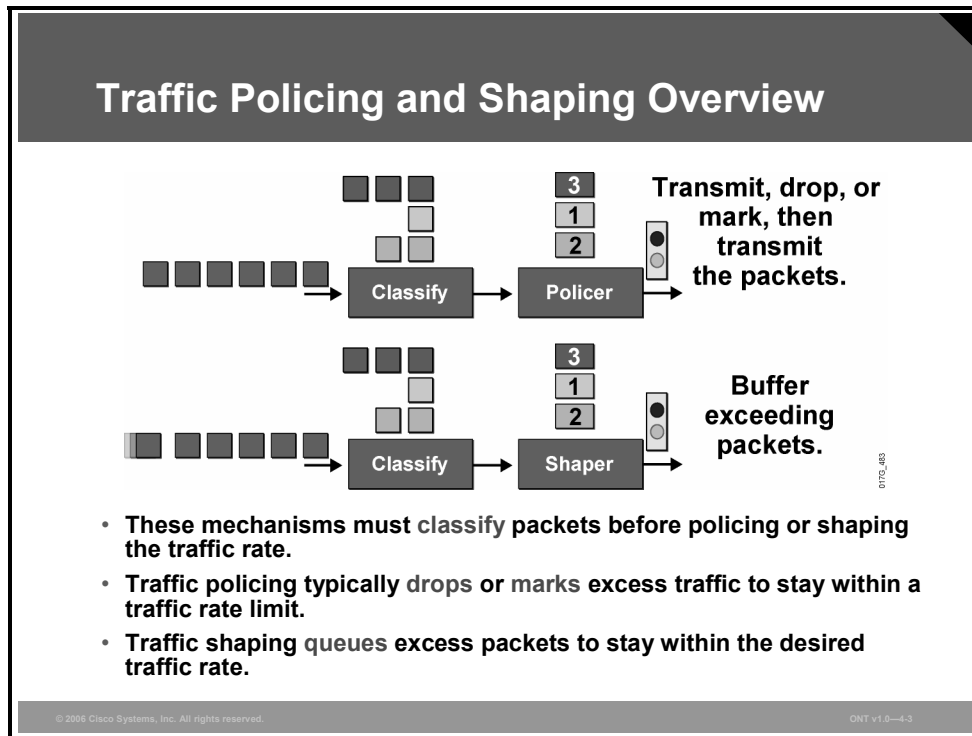
## Objectives

Upon completing this lesson, you will be able to explain Cisco class-based traffic-policing and class-based traffic-shaping operations and basic configurations. This ability includes being able to meet these objectives:

- Describe the purpose of traffic conditioning using traffic policing and traffic shaping
- List key benefits of traffic conditioning using traffic policing and traffic shaping
- Differentiate among the features of traffic policing and traffic shaping
- Describe how a token bucket can be used by network devices to measure traffic rates
- Describe how traffic can be policed using a single token bucket scheme
- Describe the key traffic policing and shaping mechanisms available in Cisco IOS software and how each compares to the others
- Identify the points in a network where rate-limiting can most effectively be employed

# Traffic Policing and Shaping Overview

This topic describes the purpose of traffic conditioning using traffic policing and traffic shaping.



Both traffic shaping and policing mechanisms are traffic-conditioning mechanisms that are used in a network to control the traffic rate. Both mechanisms use classification so that they can differentiate traffic. They both measure the rate of traffic and compare that rate to the configured traffic-shaping or traffic-policing policy.

The difference between traffic shaping and policing can be described in terms of their implementation:

- Traffic policing drops excess traffic to control traffic flow within specified rate limits. Traffic policing does not introduce any delay to traffic that conforms to traffic policies. Traffic policing can cause more TCP retransmissions, because traffic in excess of specified limits is dropped.

Traffic-policing mechanisms such as class-based policing or committed access rate (CAR) also have marking capabilities in addition to rate-limiting capabilities. Instead of dropping the excess traffic, traffic policing can mark and then send the excess traffic. This feature allows the excess traffic to be re-marked with a lower priority before the excess traffic is sent out.

- Traffic shaping buffers excessive traffic so that the traffic stays within the desired rate. With traffic shaping, traffic bursts are smoothed out by queuing the excess traffic to produce a steadier flow of data. Reducing traffic bursts helps reduce congestion in the network. Traffic shapers such as class-based shaping, Frame Relay traffic shaping (FRTS), or virtual IP (VIP)-based distributed traffic shaping (DTS) in Cisco IOS software do not have the ability to mark traffic.

# Why Use Policing?

This subtopic describes the purpose of policing.

## Why Use Policing?

- **To limit access to resources when high-speed access is used but not desired (subrate access)**
- **To limit the traffic rate of certain applications or traffic classes**
- **To mark down (recolor) exceeding traffic at Layer 2 or Layer 3**

© 2006 Cisco Systems, Inc. All rights reserved. OMT v1.0—4-4

Traffic policing is typically used to satisfy one of these requirements:

- Limiting the access rate on an interface when high-speed physical infrastructure is used in transport. Rate limiting is typically used by service providers to offer customers subrate access. For example, a customer may have an Optical Carrier-3 (OC-3) connection to the service provider but pay only for a T1 access rate. The service provider can rate-limit the customer traffic to T1 speed.
- Engineering bandwidth so that traffic rates of certain applications or classes of traffic follow a specified traffic-rate policy. For example, traffic from file-sharing applications may be rate-limited to 64 kbps maximum.
- Re-marking excess traffic with a lower priority at Layer 2 and Layer 3 or both before sending the excess traffic out. Cisco class-based traffic policing can be configured to mark packets at both Layer 2 and Layer 3. For example, excess traffic can be re-marked to a lower differentiated services code point (DSCP) value and also have the Frame Relay discard eligible (DE) bit set before the packet is sent out.

# Why Use Shaping?

Traffic shaping is typically used to prevent and manage congestion in ATM, Frame Relay, or Metro Ethernet networks, where asymmetric bandwidths are used along the traffic path. If shaping is not used, then buffering can occur at the slow (usually the remote) end, which can lead to queuing and cause delays, and overflow, which can cause drops.

## Why Use Shaping?

- **To prevent and manage congestion in ATM, Frame Relay, and Metro Ethernet networks, where asymmetric bandwidths are used along the traffic path**
- **To regulate the sending traffic rate to match the subscribed (committed) rate in ATM, Frame Relay, or Metro Ethernet networks**
- **To implement shaping at the network edge**

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—4-5

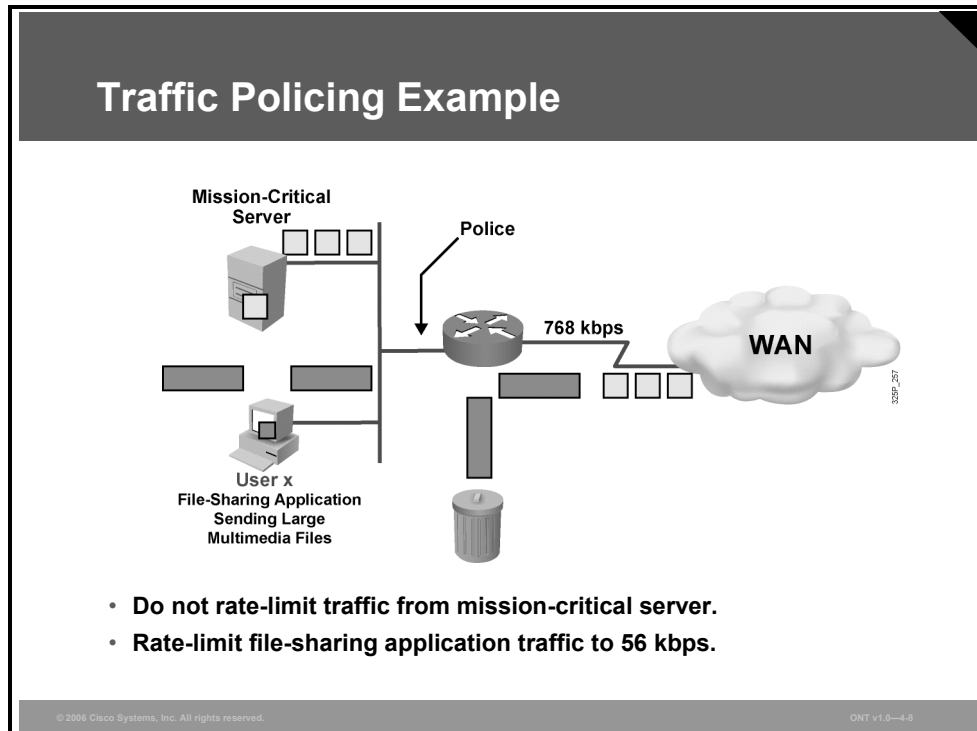
Traffic shaping is an attempt to control traffic in ATM, Frame Relay, or Metro Ethernet networks to optimize or guarantee performance, low latency, or bandwidth. Traffic shaping deals with concepts of classification, queue disciplines, enforcing policies, congestion management, quality of service (QoS), and fairness.

Traffic shaping provides a mechanism to control the volume of traffic being sent into a network (bandwidth throttling) by not allowing the traffic to burst above the subscribed (committed) rate. For this reason, traffic-shaping schemes need to be implemented at the network edges like ATM, Frame Relay, or Metro Ethernet to control the traffic entering the network. It also may be necessary to identify traffic with a granularity that allows the traffic-shaping control mechanism to separate traffic into individual flows and shape them differently.



# Why Use Traffic Conditioners?

This topic gives examples of scenarios where traffic conditioning using traffic policing and traffic shaping may be used.

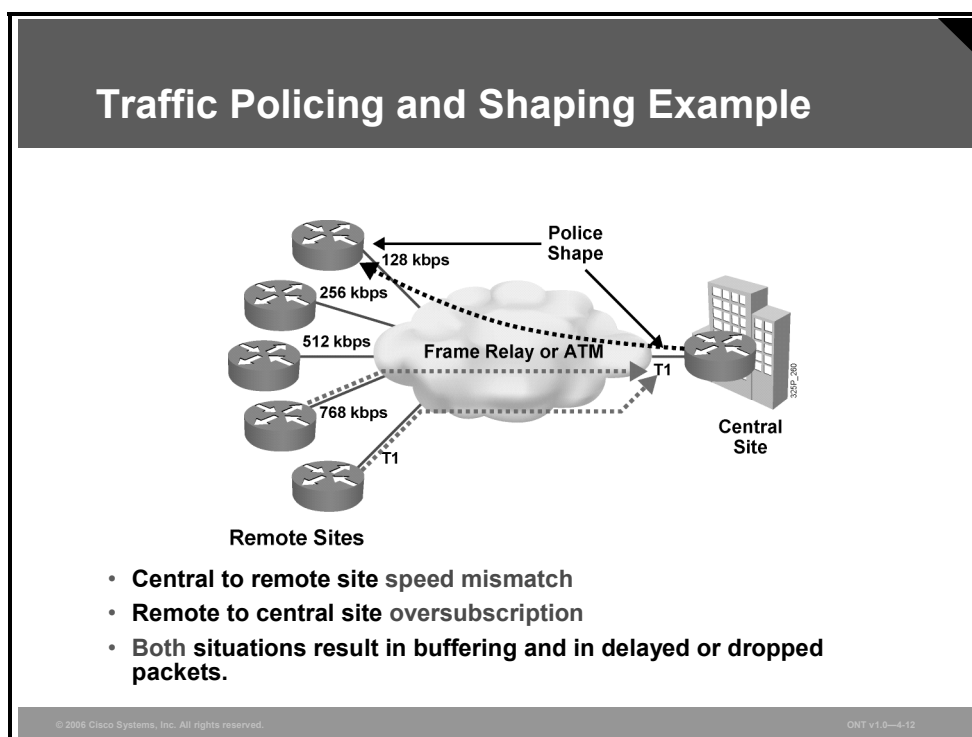


Traffic policing can be used to divide the shared resource (the upstream WAN link) among many flows. In this example, the router Fast Ethernet interface has an input traffic-policing policy applied to it in which the mission-critical server traffic rate is not limited but the user x file-sharing application traffic rate is limited to 56 kbps. All file-sharing application traffic from user x that exceeds the rate limit of 56 kbps will be dropped.

## Traffic Policing and Shaping: Example

Traffic policing tools are often configured on interfaces at the edge of a network to limit the rate of traffic entering or leaving the network. In the most common traffic-policing configurations, traffic that conforms is transmitted and traffic that exceeds is sent with a decreased priority or is dropped. Such priorities can be based on IP precedence or DSCP. Network administrators can change these configuration options to suit their network needs.

Traffic-shaping tools limit the transmit rate from a source by queuing the excess traffic. This limit is typically a value lower than the line rate of the transmitting interface. Traffic shaping can be used to account for speed mismatches, which are common in nonbroadcast multiaccess (NBMA) networks such as Frame Relay and ATM.

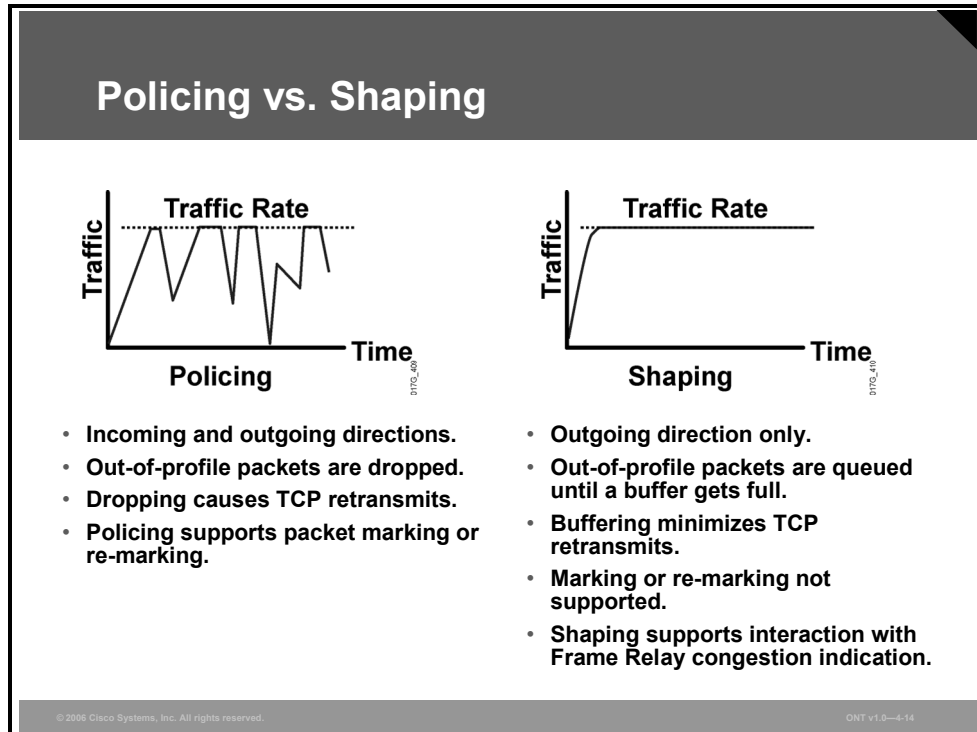


In the figure, two types of speed mismatches are shown:

- The central site can have a higher-speed link than the remote site. Thus, traffic shaping can be deployed at the central site router to shape the traffic rate out of the central site router to match the link speed of the remote site. For example, the central router can shape the permanent virtual circuit (PVC) outgoing traffic rate (going to the top remote-site router) to 128 kbps to match that remote-site link speed. At each remote-site router, traffic shaping is also implemented to shape the remote-site outgoing traffic rate to 128 kbps to match the committed information rate (CIR).
- The aggregate link speed of all the remote sites can be higher than the central site link speed (oversubscribing the central site link speed). In this case, the remote-site routers can be configured for traffic shaping to avoid oversubscription at the central site. For example, the bottom two remote-site routers can be configured to shape the PVC outgoing traffic rate to 256 kbps to keep the central-site router from being oversubscribed.
- On all the links between the central site and the remote sites, traffic policing can be used to prioritize packets and to decide, for instance, whether packets that conform can be configured to be transmitted, packets that exceed can be configured to be sent with a decreased priority, and packets that violate can be configured to be dropped. For example, packets with an IP precedence of 4 that do not exceed 128 kbps are transmitted.

# Policing vs. Shaping

This topic describes the difference between the features of traffic policing and traffic shaping.



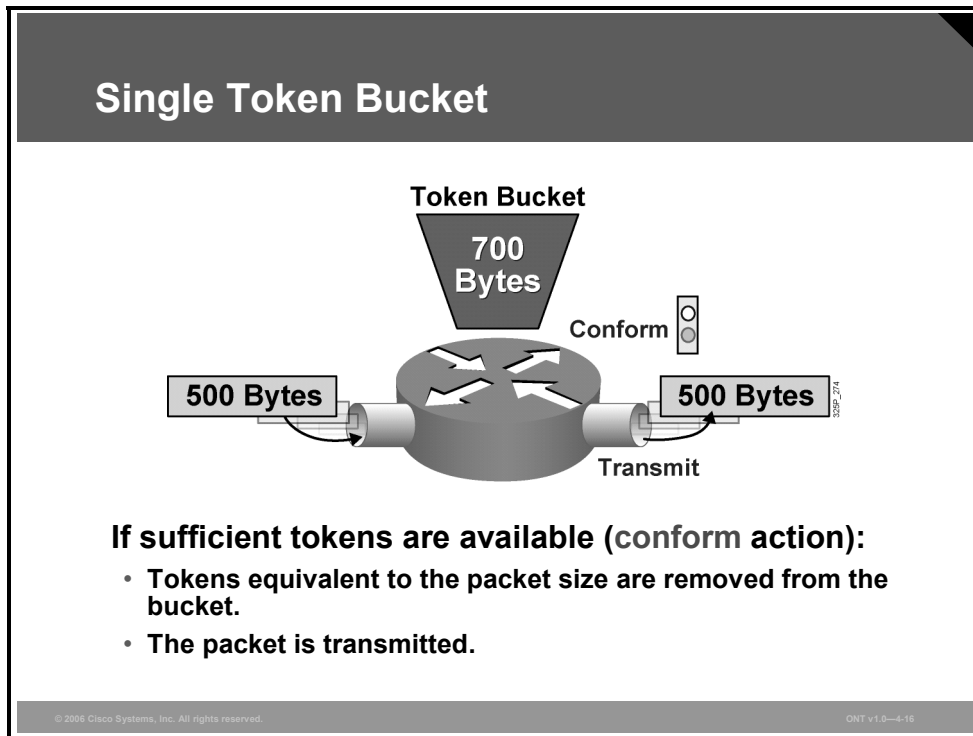
Policing can be applied to either the inbound or outbound direction, while shaping can be applied only in the outbound direction. Policing drops nonconforming traffic instead of queuing the traffic like shaping. Policing also supports marking of traffic. Traffic policing is more efficient in terms of memory utilization than traffic shaping because no additional queuing of packets is needed.

Both traffic policing and shaping ensure that traffic does not exceed a bandwidth limit, but each mechanism has different impacts on the traffic:

- Policing drops packets more often, generally causing more retransmissions of connection-oriented protocols, such as TCP.
- Shaping adds variable delay to traffic, possibly causing jitter. Shaping queues excess traffic by holding packets in a shaping queue. Traffic shaping is used to shape the outbound traffic flow when the outbound traffic rate is higher than a configured rate. Traffic shaping smoothes traffic by storing traffic above the configured rate in a shaping queue. Therefore, shaping increases buffer utilization on a router and causes unpredictable packet delays. Traffic shaping can also interact with a Frame Relay network, adapting to indications of Layer 2 congestion in the WAN. For example, if the backward explicit congestion notification (BECN) bit is received, the router can lower the rate limit to help reduce congestion in the Frame Relay network.

# Measuring Traffic Rates

This topic describes how a token bucket can be used by network devices to measure traffic rates.



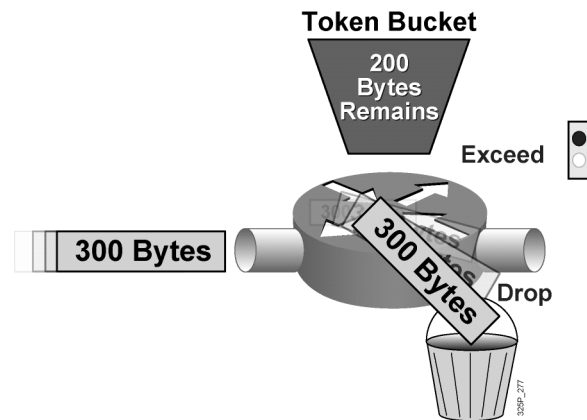
The token bucket is a mathematical model that is used by routers and switches to regulate traffic flow. The model has two basic components:

- **Tokens:** Each token represents permission to send a fixed number of bits into the network. Tokens are put into a token bucket at a certain rate by Cisco IOS software.
- **Token bucket:** A token bucket has the capacity to hold a specified number of tokens. Each incoming packet, if forwarded, takes tokens from the bucket representing the packet size. If the bucket fills to capacity, newly arriving tokens are discarded. Discarded tokens are not available to future packets. If there are not enough tokens in the token bucket to send the packet, the traffic-conditioning mechanisms may take the following actions:
  - Wait for enough tokens to accumulate in the bucket (traffic shaping)
  - Discard the packet (traffic policing)

Using a single token bucket model, the measured traffic rate can conform to or exceed the specified traffic rate. The measured traffic rate is conforming if there are enough tokens in the token bucket to transmit the traffic. The measured traffic rate is exceeding if there are not enough tokens in the token bucket to transmit the traffic.

The figure shows a single token bucket traffic policing implementation. Starting with a current capacity of 700 bytes worth of tokens accumulated in the token bucket, when a 500-byte packet arrives at the interface, its size is compared to the token bucket capacity (in bytes). The 500-byte packet conforms to the rate limit (500 bytes is less than 700 bytes), and the packet is forwarded: 500 bytes worth of tokens are taken out of the token bucket, leaving 200 bytes worth of tokens for the next packet.

## Single Token Bucket (Cont.)



**If sufficient tokens are not available (exceed action):**

- Drop (or mark) the packet.

Continuing with the single token bucket example from the previous figure, when the next 300-byte packet arrives immediately after the first packet, no new tokens have been added to the bucket (which is done periodically). This packet exceeds the rate limit. The current packet size (300 bytes) is greater than the current capacity of the token bucket (200 bytes), and the exceed action is performed. In traffic policing, the exceed action can be to drop or mark the packet.

### Example: Token Bucket as a Piggy Bank

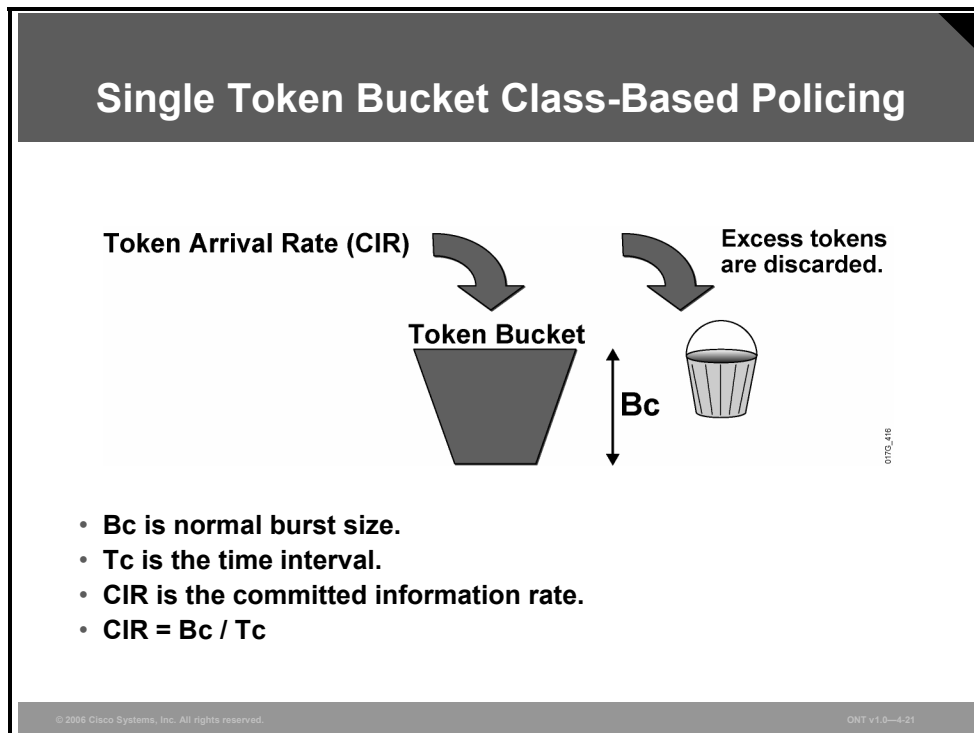
Think of a token bucket as a piggy bank. Every day you can insert one dollar into the piggy bank (the token bucket). At any given time, you can spend only what you have saved in the piggy bank. If your saving rate is one dollar per day, your long-term average spending rate will be one dollar per day if you constantly spend what you saved. However, if you do not spend any money on a given day, then you can build up your savings to the maximum that the piggy bank can hold. For example, if the piggy bank is limited to holding five dollars, and if you save and do not spend for five straight days, the piggy bank will contain five dollars. When the piggy bank fills to its capacity, you will not be able to put any more money in it. Then, at any time, you can spend up to five dollars (bursting above the long-term average rate of one dollar per day).

To define a conforming rate, using the piggy bank example: If you have two dollars in the piggy bank and spend one dollar, you are spending at a conforming rate because you are not spending more than you have saved.

To define an exceeding rate, using the piggy bank example: If you have two dollars in the piggy bank and try to spend three dollars, you are spending at an exceeding rate because you are trying to spend more than you have saved.

# Single Token Bucket Class-Based Policing

This topic describes how traffic can be policed using a single token bucket scheme.



Token bucket operations rely on parameters such as CIR, committed burst (Bc), and committed time interval (Tc). Bc is known as the normal burst size. The mathematical relationship between CIR, Bc, and Tc is as follows:

$$CIR \text{ (bps)} = Bc \text{ (bits)} / Tc \text{ (sec)}$$

With traffic policing, new tokens are added into the token bucket based on the interpacket arrival rate and the CIR. Every time a packet is policed, new tokens are added back into the token bucket. The number of tokens added back into the token bucket is calculated as follows:

$$(\text{Current Packet Arrival Time} - \text{Previous Packet Arrival Time}) * CIR$$

An amount (Bc) of tokens is forwarded without constraint in every time interval (Tc). For example, if 8000 bits (Bc) worth of tokens are placed in the bucket every 250 ms (Tc), the router can steadily transmit 8000 bits every 250 ms if traffic arrives constantly at the router.

$$CIR \text{ (normal burst rate)} = 8,000 \text{ bits (Bc)} / 0.25 \text{ seconds (Tc)} = 32 \text{ kbps}$$

When configuring Cisco IOS class-based traffic policing, it is recommended that you allow Cisco IOS software to automatically calculate the optimal Bc and Tc value based on the configured CIR.

Without any excess bursting capability, if the token bucket fills to capacity (Bc of tokens), the token bucket overflows and newly arriving tokens are discarded. Using the example, in which the CIR is 32 kbps (Bc = 8000 bits and Tc = 0.25 seconds), the maximum traffic rate can never exceed a hard rate limit of 32 kbps.

# Cisco IOS Traffic Policing and Shaping Mechanisms

This topic describes the key traffic policing and shaping mechanisms available in Cisco IOS software and how each compares to the others.

Cisco IOS Traffic Policing Mechanism	
	Class-Based Policing
Enable method	Enabled in policy map
Conditions	Conform, exceed, violate
Actions	Drop, set, transmit
Implementations	Single or dual token bucket, single- or dual-rate policing, multiactions

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—4-23

The table lists the characteristics of the class-based traffic-policing mechanism that is available in Cisco IOS software. Class-based policing is also available on some Cisco Catalyst switches.

Class-based policing supports a single or dual token bucket. Class-based policing also supports single-rate or dual-rate metering and multiaction policing. Multiaction policing allows more than one action to be applied; for example, marking the Frame Relay DE bit and also the DSCP value before sending the exceeding traffic.

Class-based policing is configured using the Cisco Modular QoS CLI (MQC), using the **police** command under the policy map configuration.

---

**Note** A token bucket is a formal definition of a rate of transfer. It has three components: a burst size, a mean rate, and a time interval (Tc). Although the mean rate is generally represented as bits per second, any two values may be derived from the third by the relation shown as follows: *mean rate = burst size / time interval*. A token bucket is used to manage a device that regulates the data in a flow. The single token bucket is used for single-rate metering. To provide more metering granularity, the single-rate token bucket function is doubled, resulting in the dual token bucket that is used in the dual-rate metering.

---

# Cisco IOS Traffic-Shaping Mechanisms

The table lists two of the traffic-shaping mechanisms available in Cisco IOS software: class-based traffic shaping and FRTS.

Cisco IOS Traffic-Shaping Mechanisms		
	Class-Based Shaping	FRTS
Restriction	Shaper for any subinterface	Shaper for Frame Relay only
Classification	Class-based	Per DLCI or subinterface
Link fragmentation and interleaving	No support for FRF.12	Supports FRF.12
Frame Relay Support	Understands BECN and FECN	Understands BECN and FECN
Configuration	Supported via MQC	Supported via MQC

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—4-24

Class-based traffic shaping uses the MQC to allow traffic to be shaped per traffic class as defined by the class map. Class-based traffic shaping can be used in combination with class-based weighted fair queuing (CBWFQ), in which the shaped rate is used to define an upper rate limit while the bandwidth statement within the CBWFQ configuration is used to define a minimum rate limit.

FRTS is used to shape Frame Relay traffic only. FRTS allows an individual PVC (data-link connection identifier [DLCI]) to be shaped. FRTS can use priority queuing (PQ), custom queuing (CQ), or weighted fair queuing (WFQ) as the shaping queue and supports only FIFO as the software queue.

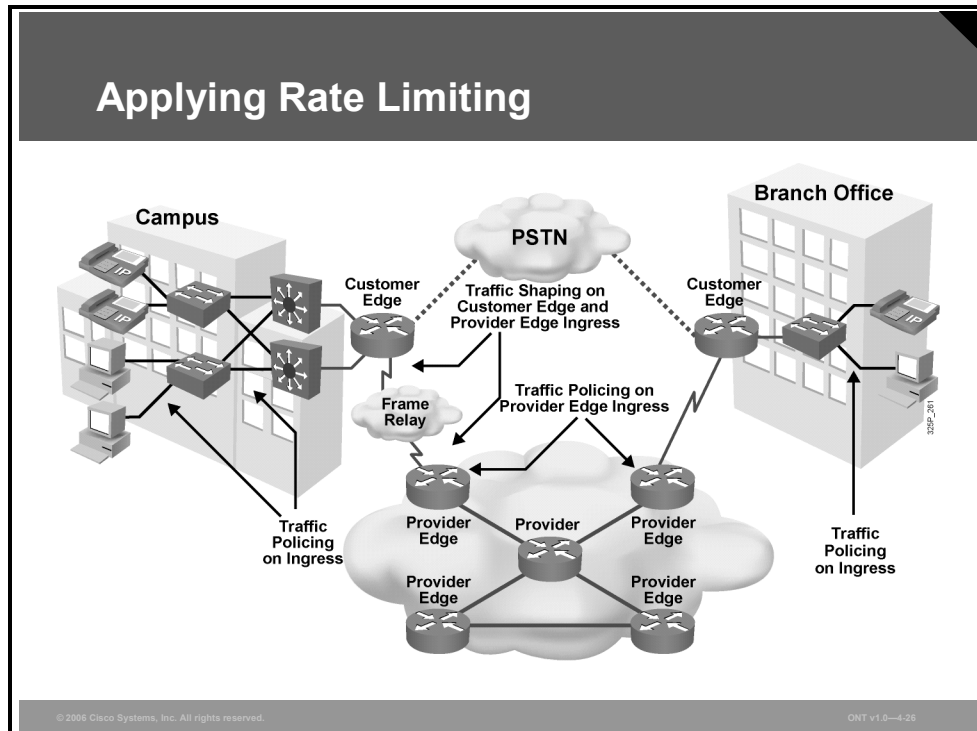
FRTS supports FRF.12 Frame Relay fragmentation, while class-based shaping does not support FRF.12 fragmentation for Frame Relay.

Traffic-shaping mechanisms can interact with a Frame Relay network, adapting to indications of Layer 2 congestion in the WAN. For example, if the backward explicit congestion notification (BECN) bit is received, the router can lower the rate limit to help reduce congestion in the Frame Relay network. And if the forward explicit congestion notification (FECN) bit is received, the router can generate a test frame with the BECN bit set. This enables the sender to notice congestion even if there is no data traffic flowing back from the receiver to the sender.



# Applying Traffic Conditioners

This topic describes the points in a network where rate limiting can most effectively be employed.



In a typical enterprise network, traffic policing is often implemented at the access or distribution layer to limit certain traffic classes before that traffic exits the campus onto the WAN. Traffic shaping is often implemented at the WAN edge when there are speed mismatches or oversubscription.

In a typical service provider network, traffic policing is often implemented inbound at the provider edge router to rate-limit incoming traffic from the customer edge router to ensure that the customer traffic rate is not exceeding the contractual rate. Traffic shaping is often implemented outbound at the provider edge and at the customer edge to limit the traffic rate between the provider edge and customer edge and to allow for FRF.12 fragmentation on Frame Relay connections between the customer edge and provider edge.

# Summary

This topic summarizes the key points that were discussed in this lesson.

## Summary

- **Traffic shaping and policing are mechanisms that use classification to limit traffic rate.**
- **Traffic shaping queues excess packets to stay within the contractual rate. Traffic policing typically drops excess traffic to stay within the limit; alternatively, it can re-mark, then send excess traffic.**
- **Both traffic policing and shaping ensure that traffic does not exceed a bandwidth limit, but they have different impacts on the traffic.**
- **The token bucket is a mathematical model that is used by routers and switches to regulate traffic flow.**
- **With a single token bucket model, the measured traffic rate can conform to or exceed the specified traffic rate.**
- **Class-based policing is the latest Cisco IOS traffic-policing mechanism. Class-based shaping and FRTS are two Cisco IOS traffic-shaping mechanisms.**
- **Policing is often implemented at the access or distribution layer, shaping is implemented at the WAN edge.**

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—4-27

# Understanding WAN Link Efficiency Mechanisms

---

## Overview

Interactive traffic such as Telnet and VoIP is susceptible to increased latency when network processes using large packets, such as bulk FTP, traverse WAN links. Packet delay is especially significant when FTP packets are queued on slower links within the WAN. To solve delay problems on slow links, a method for fragmenting larger frames and then queuing the smaller frames between fragments of the larger frames is required. To meet this requirement, tools such as header and payload compression and link fragmentation and interleaving (LFI) can be used to reduce the size of frames that are sent on WAN links.

This lesson describes various approaches for improving the efficiency of WAN links. It describes link efficiency mechanisms that either compress the payload or reduce packet headers. It also describes the various Layer 2 LFI mechanisms and Frame Relay fragmentation.

## Objectives

Upon completing this lesson, you will be able to explain Cisco class-based header compression operations and basic configurations. This ability includes being able to meet these objectives:

- Explain the various link efficiency mechanisms and their functions
- Describe the purpose of Layer 2 payload compression and how Layer 2 payload compression affects throughput and delay
- Describe the purpose of header compression and how header compression affects throughput and delay
- Explain how VoIP packets are susceptible to increased latency when large packets, such as FTP transfers, traverse slow WAN links
- Explain LFI operation and how LFI reduces delay and jitter for VoIP packets
- Identify the points in a network where link efficiency mechanisms can most effectively be employed

# Link Efficiency Mechanisms Overview

This topic describes link efficiency mechanisms and their functions.

## Compression

- **Data compression works by the identification of patterns in a stream of data.**
- **Basic elements of compression:**
  - Remove redundancy as much as possible.
  - There is a theoretical limit, known as **Shannon's limit**.
- **Many compression algorithms exist, for different purposes:**
  - **MPEG compression for video**
  - **Huffmann compression for text and software**
  - **LZ compression, used in Stacker compression**
- **Two methods of compression are used:**
  - **Hardware compression**
  - **Software compression**

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—4.3

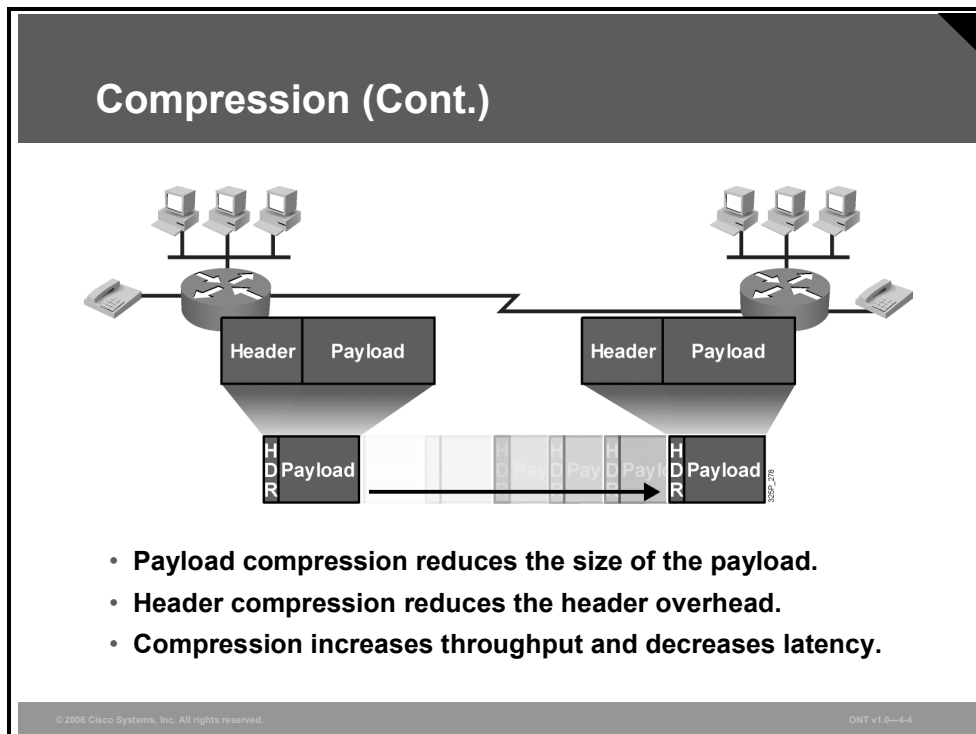
## Compression

Data compression works by the identification of patterns in a stream of data. Data compression chooses a more efficient method to represent the same information. Essentially, an algorithm is applied to the data in order to remove as much redundancy as possible. The efficiency and effectiveness of a compression scheme is measured by its compression ratio, the ratio of the size of uncompressed data to compressed data. A compression ratio of 2:1 (which is relatively common) means that the compressed data is half the size of the original data.

Several compression algorithms exist. Some algorithms are designed to take advantage of a specific medium and the redundancies found in it. However, they do a poor job when applied to other sources of data. For example, the MPEG standard is designed to take advantage of the relatively small difference between one frame and another in video data. It does an excellent job in compression of motion pictures, but does not compress text well. For text compression, the Huffmann compression algorithm would be used. Lempel-Ziv (LZ) compression is used in Stacker compression. One of the most important concepts in compression theory is that there is a theoretical limit, known as Shannon's limit, that describes how far a given source of data can be compressed. Modern compression algorithms coupled with the fast processors available today allow compression to approach Shannon's limit.

Hardware compression and software compression refer to the site in the router to which the compression algorithm is applied. In software compression, compression is implemented in the main CPU as a software process. In hardware compression, the compression computations are off-loaded to a secondary hardware module. This frees the central CPU from the computationally intensive task of calculating compression.

If you assume that the router has the clock cycles available to perform the compression calculations—for example, CPU utilization remains at a reasonable level—then there is no difference between the efficiency of hardware compression and software compression. The achieved compression ratio is a function of the compression algorithm selected and the amount of redundancy in the data to be compressed, not where the compression calculations take place.



Payload compression squeezes payloads, either the Layer 2 payload or the Layer 3 payload. With Layer 2 payload compression, the Layer 2 header remains intact, but its payload (Layer 3 and above) is compressed. With Layer 3 payload compression, Layer 2 and 3 headers remain intact. Payload compression increases the throughput and decreases the latency in transmission, because smaller packets (with compressed payloads) take less time to transmit than the larger, uncompressed packets. Layer 2 payload header compression is performed on a link-by-link basis, whereas Layer 3 payload compression is generally used on a session-by-session basis.

Header compression methods work by *not* transmitting repeated information in packet headers throughout a session. The two peers on a PPP Layer 2 connection (a dial-up link) agree on session indices that index a dictionary of packet headers. The dictionary is built at the start of every session and is used for all subsequent (noninitial) packets. Only changing, or nonconstant, parameters in the headers are actually sent along with the session index.

Header compression cannot be performed across multiple routers because routers need full Layer 3 header information to be able to route packets to the next hop.

# Link Efficiency Mechanisms

Although there are many quality of service (QoS) mechanisms for optimizing throughput and reducing delay in network traffic, QoS mechanisms do not create bandwidth. QoS mechanisms optimize the use of existing resources and enable differentiation of traffic according to a policy. Link efficiency QoS mechanisms such as payload compression, header compression, and link fragmentation and interleaving (LFI) are deployed on WAN links to optimize the use of WAN links.

## Link Efficiency Mechanisms

- **Link efficiency mechanisms are often deployed on WAN links to increase the throughput and to decrease delay and jitter.**
- **Cisco IOS link efficiency mechanisms include:**
  - **Layer 2 payload compression (Stacker, Predictor, MPPC)**
  - **Header compression (TCP, RTP, class-based TCP, and class-based RTP)**
  - **LFI (MLP, FRF.12, and FRF.11.C)**

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—4.5

Compression increases the amount of data that can be sent through a transmission resource. Payload compression is primarily performed on Layer 2 frames and therefore compresses the entire Layer 3 packet. The Layer 2 payload compression methods include these:

- Stacker
- Predictor
- Microsoft Point-to-Point Compression (MPPC)

These algorithms differ vastly in their compression efficiency and in their use of router resources.

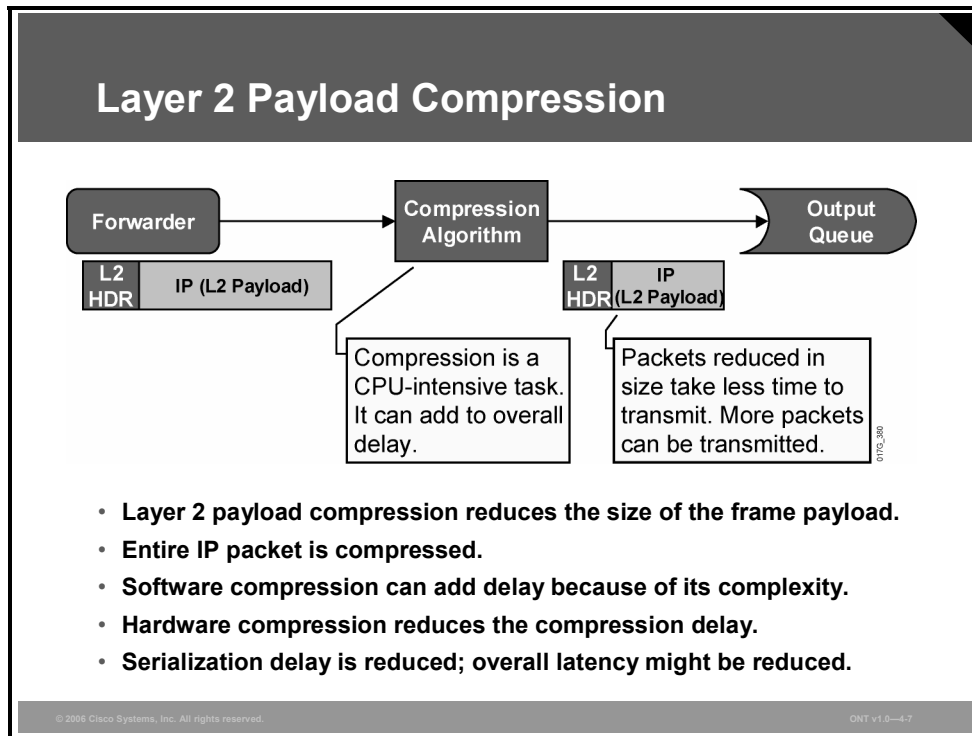
Compression methods are based on eliminating redundancy. The protocol header is an item of repeated data. The protocol header information in each packet in the same flow does not change much over the lifetime of that flow. Using header-compression mechanisms, most header information can be sent only at the beginning of the session, stored in a dictionary, and then referenced in later packets by a short dictionary index. The header-compression methods include:

- TCP
- Real-Time Transport Protocol (RTP)
- Class-based TCP
- Class-based RTP

LFI is a Layer 2 technique in which large frames are broken into small, equal-sized fragments and transmitted over the link in an interleaved fashion. Using LFI, smaller frames are prioritized, and a mixture of fragments is sent over the link. LFI reduces the queuing delay of small frames because the small frames are sent almost immediately. LFI, therefore, reduces delay and jitter by expediting the transfer of smaller frames through the hardware transmit (Tx) queue. The LFI methods available include Multilink PPP (MLP), FRF.12, and FRF.11 Annex C.

# Layer 2 Payload Compression

This topic describes the purpose of Layer 2 payload compression and how Layer 2 payload compression affects throughput and delay.



When a router forwards a packet, the packet is subjected to the Layer 2 compression method after it has been encapsulated at the output. The compression method squeezes the payload of the Layer 2 frame (the entire Layer 3 packet), and transmits the packet on the interface.

Layer 2 payload compression is a CPU-intensive task and can add per-packet compression delay because of the application of the compression method to each frame. The serialization delay, however, is reduced, because the resulting frame is smaller. Serialization delay is the fixed delay that is required to clock the frame onto the network interface. Depending on the complexity of the Layer 2 payload compression algorithm, overall latency might be reduced, especially on low-speed links.

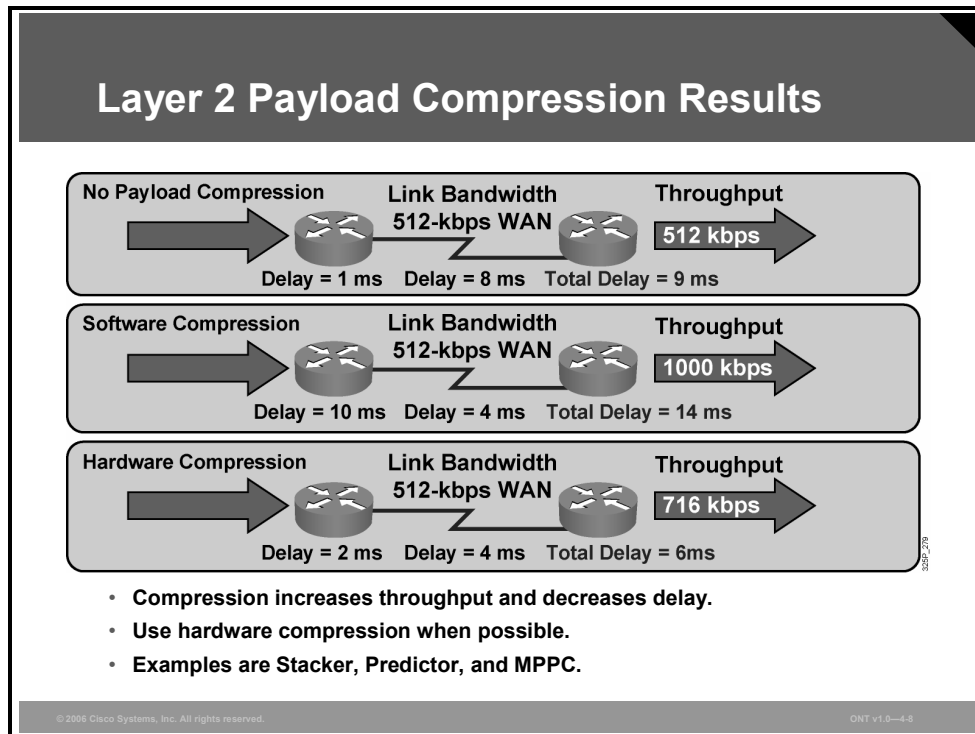
Cisco routers support hardware-assisted compression to reduce the CPU load and the Layer 2 payload compression delay.

Layer 2 payload compression involves the compression of the payload of a Layer 2 WAN protocol, such as PPP, Frame Relay, High-Level Data Link Control (HDLC), X.25, and Link Access Procedure, Balanced (LAPB). The Layer 2 header is untouched by the act of compression. However, the entire contents of the payload (which include higher-layer protocol headers) are compressed. They are compressed using either a form of the Stacker algorithm (based on the industry standard LZ algorithm) or the Predictor algorithm, which is an older algorithm that is mostly used in legacy configurations.



## Layer 2 Payload Compression Results

This figure compares three throughput and latency scenarios on a WAN link.



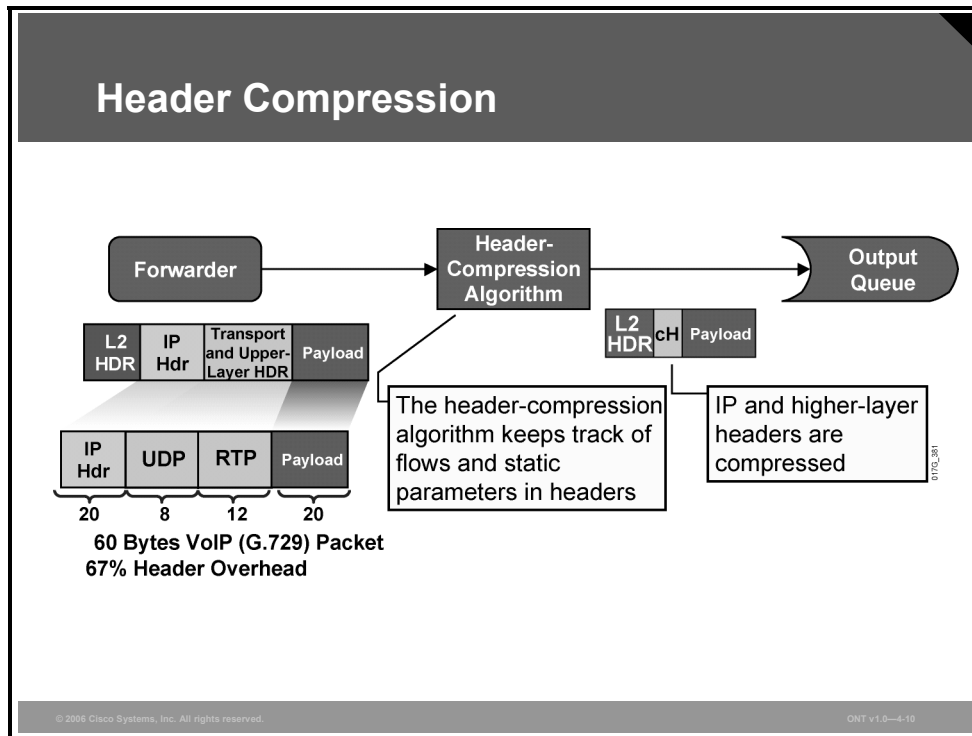
If no compression is used, throughput is limited by the link bandwidth, and the average delay is influenced by forwarding or buffering delay, serialization, and propagation delay.

If compression is enabled—even if the serialization delay is now shorter because the frame is smaller—the compression or decompression delay may increase the overall latency between the two hops. The perceived throughput is generally increased because the size of the Layer 2 payload is reduced, allowing more Layer 2 frames to be sent through a transmission resource in a given time period. Throughput is limited by the effectiveness of the Layer 2 payload-compression algorithm and may be significantly higher than the link bandwidth limit.

If hardware-assisted Layer 2 payload compression is used, compression or decompression delays may become insignificant compared to forwarding and serialization delays, and overall latency may decrease. Throughput is again limited by the effectiveness of the Layer 2 payload compression method and may be significantly higher than the link bandwidth limit.

# Header Compression

This topic describes the purpose of header compression and how header compression affects throughput and delay.



Header compression increases throughput and reduces delay by compressing the protocol headers. Header compression is most useful for applications that generate small payloads because the protocol headers of such applications use a significant percentage of bandwidth on a link relative to their payload. Header compression based on session dictionary techniques works by replacing phrases in the input string with indexes into some dictionary table.

When header compression is applied on a TCP/IP header, some of the redundant fields in the header of a TCP/IP connection are removed. Header compression keeps a copy of the original header on either side of the link, removes the entirely redundant fields, and differentially codes the remaining fields in order to allow the compression of 40 bytes of header to an average of 5 bytes. This process uses a very specific algorithm designed around the constant structure of the TCP/IP header. It does not touch the payload of the TCP packet in any way.

TCP and RTP header compression applies to all TCP and RTP flows. For example, if TCP compression is enabled on a link, there is no mechanism to restrict its function to specific application types. TCP header compression for bulk data transfer yields little bandwidth savings. Class-based TCP header compression can be performed on specific traffic classes, such as the Telnet traffic class.

Class-based TCP header compression allows configuring RTP or TCP IP header compression on a per-class basis, when a class is configured within a policy map. Policy maps are created using the Cisco Modular QoS CLI (MQC).

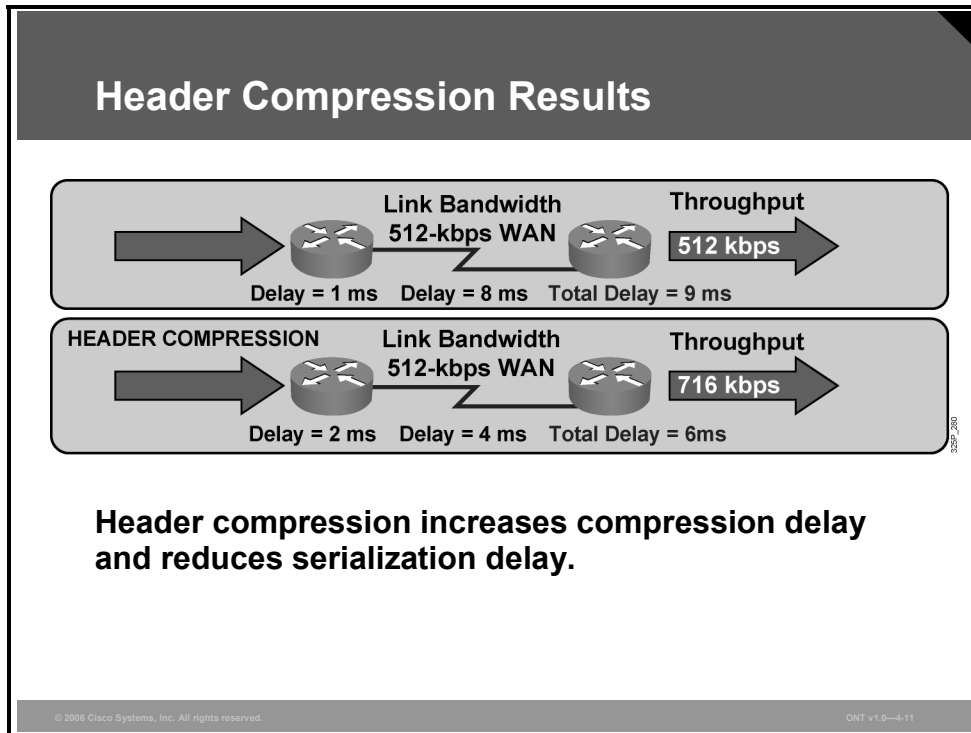
The header-compression algorithm tracks active transport layer connections over an interface. After the packet has been forwarded, the header-compression algorithm compresses the Layer 3 and Layer 4 headers within the frame and replaces the headers with a session index from the session dictionary. Only the nonconstant parameters in the headers are sent along with the session index. The packet is then sent to the output queue and transmitted to the remote peer.

When the remote peer receives the packet, the header is decompressed using the local session table and passed to the forwarding process.

For example, without RTP header compression, the IP, User Datagram Protocol (UDP), and RTP header overhead of the voice packet shown in the figure is about 67 percent ( $40 / 60 * 100$  percent). With RTP header compression, the IP, UDP, and RTP header can be reduced to 2 or 4 bytes (without and with checksum, respectively) for most packets. Thus the IP, UDP, and RTP header overhead can be reduced to about 9 percent ( $2 / 22 * 100$  percent) or 17 percent ( $4 / 24 * 100$  percent).

# Header Compression Results

This figure compares two throughput and latency scenarios on a WAN link.

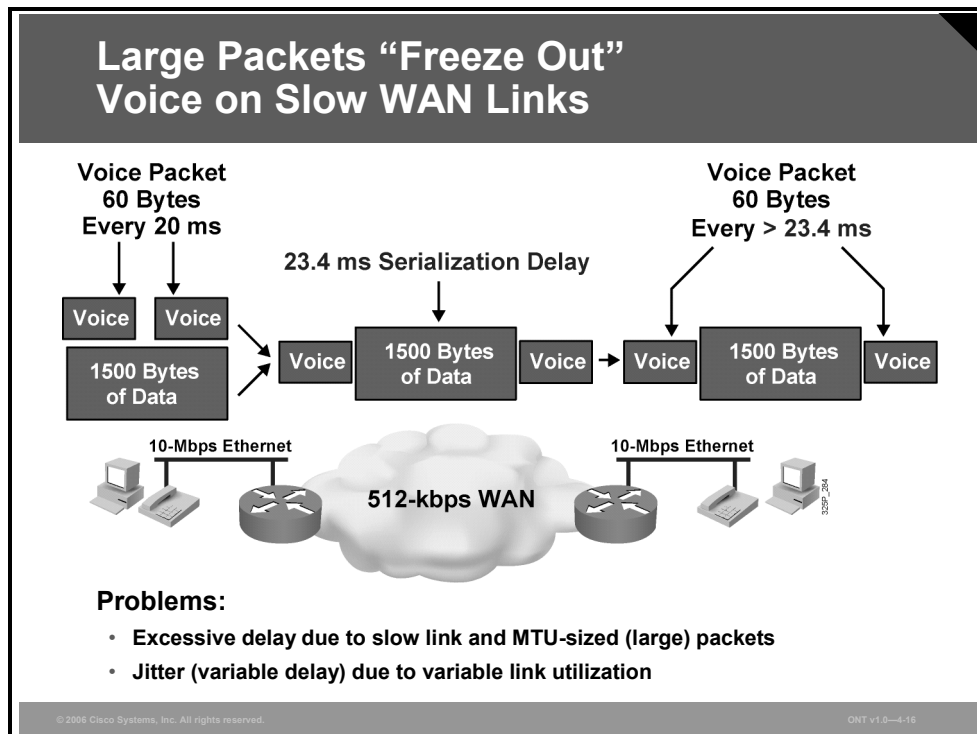


If header compression is not used, throughput is limited by the link bandwidth, and the average delay is influenced only by forwarding or buffering delay, serialization, and propagation delay.

If header compression is enabled, compressing the protocol headers causes the packet to become smaller, allowing more packets to be sent through a transmission resource in a given time period to increase throughput. Because the packet size is smaller, the serialization delay also becomes smaller, reducing the overall delay. Header compression has a low compression delay and a relatively low CPU overhead and is recommended on links slower than 2 Mbps.

# Large Packets “Freeze Out” Voice on Slow WAN Links

This topic describes the susceptibility of VoIP packets to increased latency when large packets, such as FTP transfers, traverse slow WAN links.



In considering delay between two hops in a network, queuing delay in a router must be taken into account because it may be comparable to—or even exceed—serialization and propagation delay on a link. In an empty network, an interactive or voice session experiences low or no queuing delay, because the session does not compete with other applications on an interface output queue. Also, the small delay does not vary enough to produce significant jitter on the receiving side.

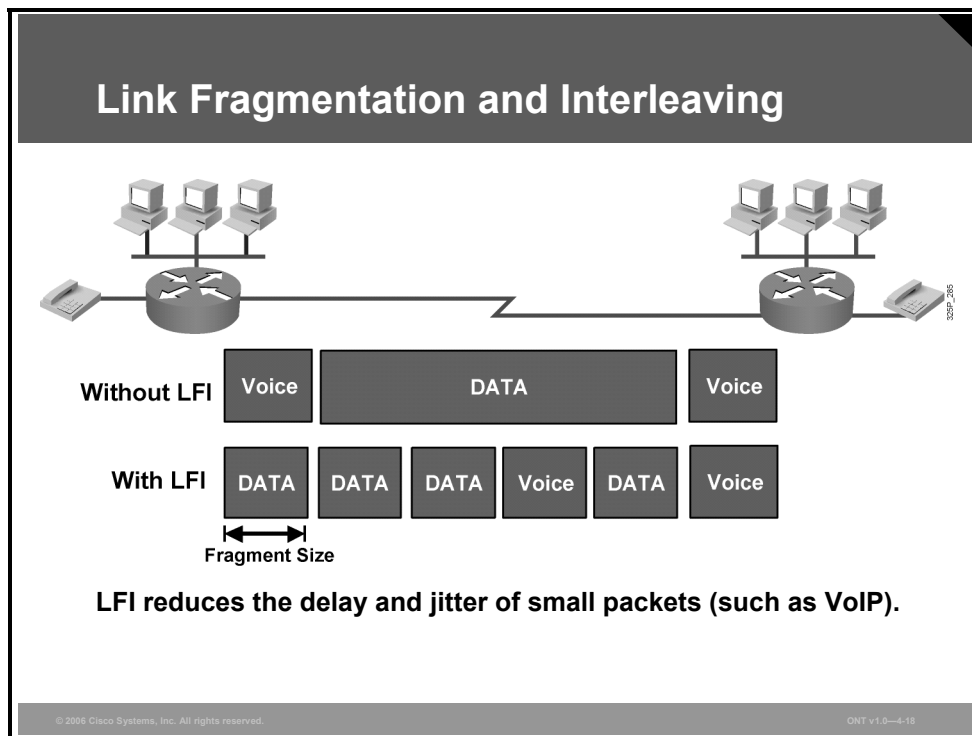
In a congested network, interactive data and voice applications compete in the router queue with other applications. Queuing mechanisms may prioritize voice traffic in the software queue, but the hardware queue (TxQ) always uses a FIFO scheduling mechanism. After packets of different applications leave the software queue, the packets will mix with other packets in the hardware queue, even if their software queue processing was expedited. Thus, a voice packet may be immediately sent to the hardware queue where two large FTP packets are waiting for transmission. The voice packet must wait until the FTP packets are transmitted, thus producing an unacceptable delay in the voice path. Because links are used variably, the delay varies with time and may produce unacceptable jitter in jitter-sensitive applications such as voice.

For example, the serialization delay of a 1500-byte packet over a 512-kbps link will be 24.3 ms. For VoIP traffic, the maximum recommended one-way, end-to-end delay is 150 ms. Therefore, having a 1500-byte packet ahead of a VoIP packet in the hardware queue on a 512-kbps link can cause the end-to-end delay of the voice packet to be over the budget of 24.3 ms.

**Note** It is highly recommended that you run the same version of code on both sides of the WAN link to ensure compatibility and correct compression results.

# Link Fragmentation and Interleaving

This topic describes LFI operation and how LFI reduces delay and jitter for VoIP packets.



The use of a hybrid queuing method such as low latency queuing (LLQ) can provide low latency and low jitter for VoIP packets while servicing other data packets in a fair manner. But even if VoIP packets are always sent to the front of the software queue, there is still the issue of serialization delay. A large packet may be on its way out of the hardware queue, which uses FIFO. When a VoIP packet is sent to the front of the software queue, the serialization of the large packet in the hardware transmit queue can cause the VoIP packet to wait for a long time before it can be transmitted out.

The solution is to fragment the large packets so that they never cause a VoIP packet to wait for more than a predefined amount of time. The VoIP packets must also be allowed to transmit in between the fragments of the larger packets (interleaving), or there will be no point in doing the fragmenting.

When you are configuring the proper fragment size to use on a link, a typical goal is to have a maximum serialization delay of around 10 to 15 ms. Depending on the LFI mechanisms being configured, the fragment size is either configured in bytes or in milliseconds, as shown in the figure.

# Applying Link Efficiency Mechanisms

This topic describes the points in a network where link efficiency mechanisms can most effectively be employed.

## Applying Link Efficiency Mechanisms

- **Identify bottlenecks in the network.**
- **Calculate Layer 2 and Layer 3 overhead.**
- **Decide which type of compression to use, such as TCP header compression.**
- **Enable compression on WAN interfaces.**

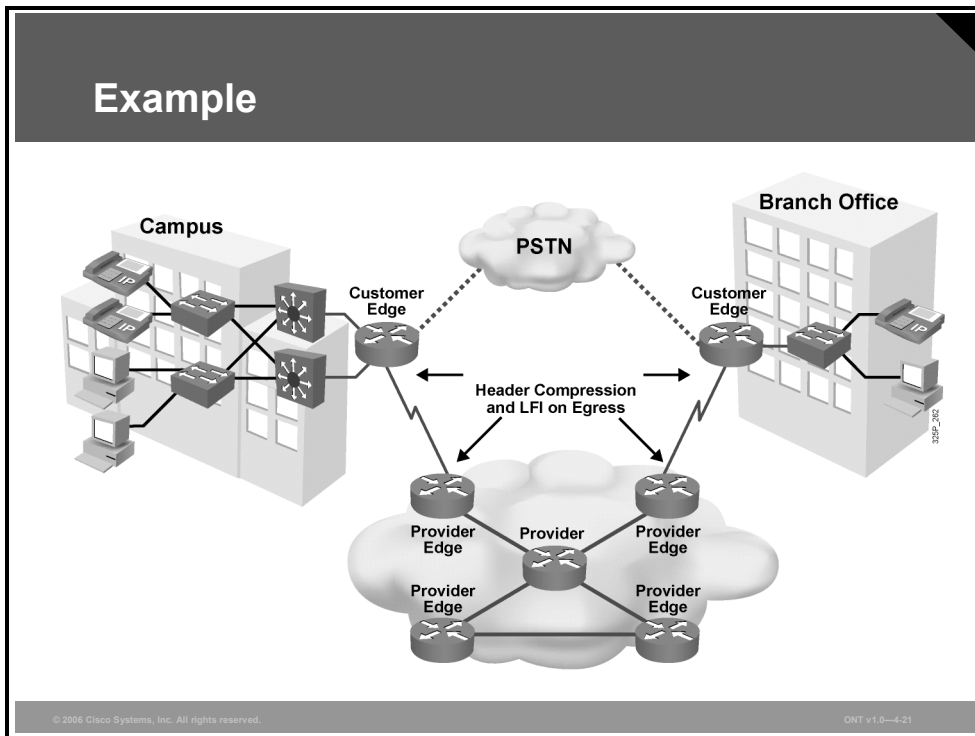
© 2006 Cisco Systems, Inc. All rights reserved. OMT v1.0—4-20

The following guidelines should be used for applying link efficiency mechanisms:

- Identify slow links to help determine where the bottlenecks in the network are located and decide how to apply link efficiency mechanisms at the appropriate interfaces.
- Calculate the Layer 2 and Layer 3 overhead for each media type that will transport the business-critical traffic. This process will help you choose the correct compression type.
- Decide which type of compression should be used.
- Enable compression on the WAN interfaces.

## Example

Header compression and LFI are typically configured at the WAN edge for WAN links below T1 or E1 speeds to optimize the use of the WAN link and to prevent long serialization delay.



Layer 2 payload compression is less commonly deployed on WAN links, especially without the use of hardware-assisted payload compression. In this case, TCP and RTP compression, as well as LFI mechanisms, should be used because this network carries converged network traffic.



# Summary

This topic summarizes the key points that were discussed in this lesson.

## Summary

- **Data compression effectively increases bandwidth.**
- **Link efficiency mechanisms (including Layer 2 payload compression, header compression, and LFI) deployed on WAN links can increase throughput and decrease delay and jitter.**
- **Payload compression uses a compression algorithm to compress the payload of Layer 2 frames.**
- **Header compression reduces overhead by compressing the IP and upper-layer headers.**
- **A VoIP packet may be sent to the hardware TxQ, where large FTP packets may still be waiting for transmission. The VoIP packet must wait until the large packets are transmitted, producing an unacceptable delay in the voice path.**

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—4.22

## Summary (Cont.)

- **LFI reduces delay and jitter for small packets (for example, VoIP) by fragmenting large packets to allow a VoIP packet to wait no more than a predefined amount of time.**
- **Header compression and LFI are typically configured at the WAN edge for WAN links below T1 or E1 speeds, to optimize the use of the WAN link and to prevent long serialization delay. Layer 2 payload compression is less commonly being deployed on WAN links.**

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—4.23



# Implementing QoS Preclassify

---

## Overview

The quality of service (QoS) preclassify feature ensures that Cisco IOS QoS services operate in conjunction with tunneling and encryption. Using the QoS preclassify mechanism, Cisco IOS software can classify packets and apply the appropriate QoS service *before* data is encrypted and tunneled. This allows service providers and enterprises to treat voice, video, and mission-critical traffic with a higher priority across service provider networks while using virtual private networks (VPNs) for secure transport. When QoS preclassify is enabled, packets are classified before encryption on the output interface, allowing traffic flows to be priority-managed in congested environments. This lesson describes QoS preclassify, using QoS policies on VPN interfaces, and configuring and monitoring VPN QoS.

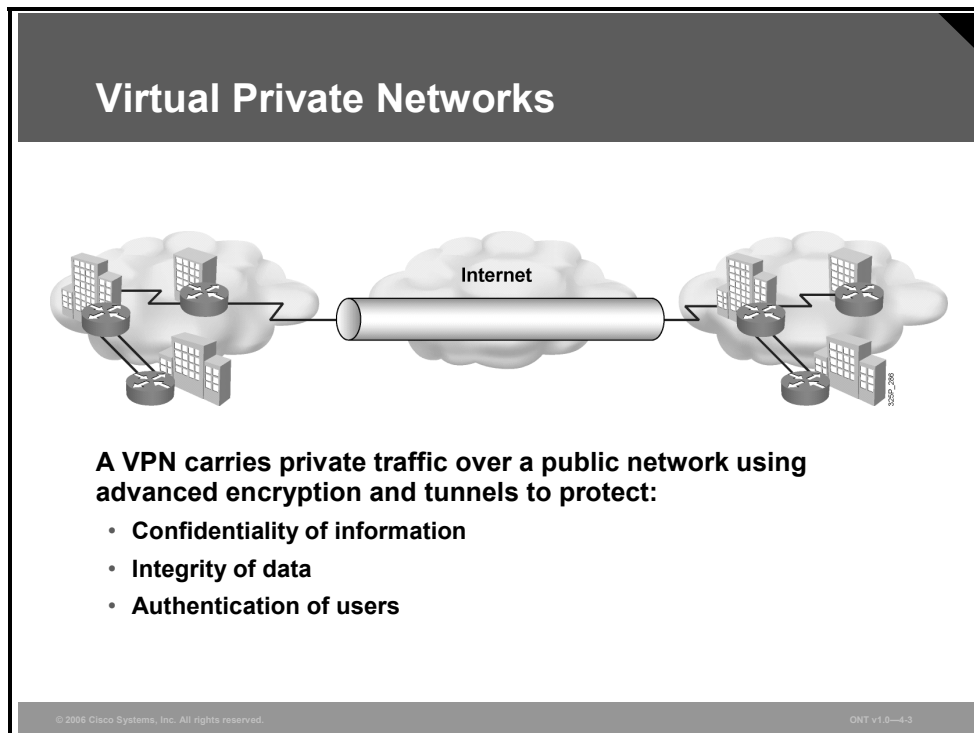
## Objectives

Upon completing this lesson, you will be able to explain the purpose and basic configuration of QoS preclassify for traffic going over IPsec and GRE tunnels. This ability includes being able to meet these objectives:

- Describe and explain the purpose of VPNs
- Describe the purpose of preclassification to support QoS in various VPN configurations
- Describe situations where preclassification is appropriate
- Describe some of the VPN applications that support QoS preclassification and situations where preclassification is not appropriate

# Virtual Private Networks

This topic describes the purpose of VPNs.



A virtual private network (VPN) is defined as network connectivity deployed on a shared (public) infrastructure with the same policies and security as a private network.

A VPN is established between two end systems or between two or more networks. A VPN can be built using tunnels, encryption, or both, at essentially any layer of the Open System Interconnection (OSI) protocol stack. A VPN is an alternative WAN infrastructure that replaces or augments private networks that use leased-line or enterprise-owned Frame Relay networks.

VPNs provide three critical functions:

- **Confidentiality (encryption):** The sender can encrypt the packets before transmitting them across a network, prohibiting anyone from eavesdropping on the communication. If intercepted, the communication cannot be read.
- **Data integrity:** The receiver can verify that the data was transmitted through the Internet without being changed or altered in any way.
- **Origin authentication:** The receiver can authenticate the source of the packet, guaranteeing and certifying the source of the information.

# VPN Types

This section defines the various VPN types.

## VPN Types

- **Remote access:**
  - **Client-initiated**
  - **Network access server**
- **Site-to-site:**
  - **Intranet**
  - **Extranet**

© 2006 Cisco Systems, Inc. All rights reserved. OMT v1.0—4-4

There are two types of remote-access VPNs:

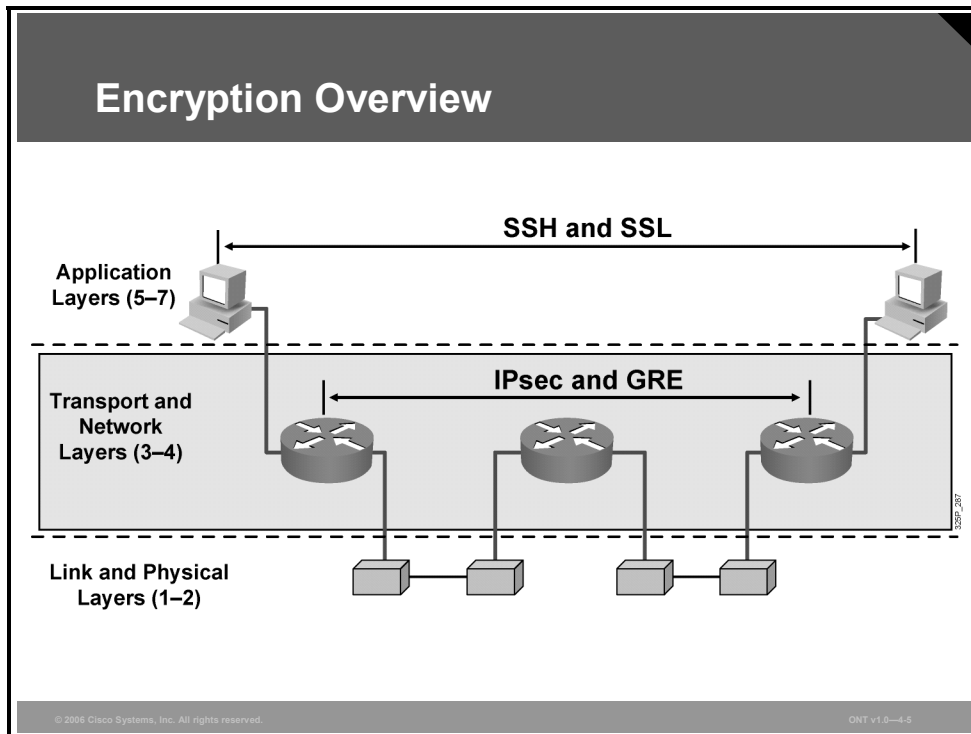
- **Client-initiated:** Remote users use clients to establish a secure tunnel across an Internet service provider (ISP)-shared network to the enterprise.
- **Network access server (NAS)-initiated:** Remote users dial in to an ISP. The NAS establishes a secure tunnel to the enterprise private network that might support multiple remote user-initiated sessions.

Site-to-site VPNs include two main types:

- **Intranet VPNs:** Connect corporate headquarters, remote offices, and branch offices over a public infrastructure
- **Extranet VPNs:** Link customers, suppliers, partners, or communities of interest to a corporate intranet over a public infrastructure

# Encryption Overview

Various methods for VPN protection are implemented on different layers. Providing privacy and other cryptographic services at the application layer was very popular in the past, and in some situations is still done today. For example, Secure Shell Protocol (SSH) offers Internet-based data-security technologies and solutions, especially for cryptography and authentication products.



The Internet Engineering Task Force (IETF) has defined a standards-based protocol called Secure Multipurpose Internet Mail Extensions (S/MIME) for VPN applications generated by a number of communication system components (for example, message transfer agents, guards, and gateways). However, application-layer security is application-specific, and protection methods must be implemented anew in every application.

Some standardization has been successful at Layer 4 (transport) of the OSI model, with protocols such as Secure Socket Layer (SSL) providing privacy, authenticity, and integrity to TCP-based applications. SSL is popular in modern e-commerce sites, but it fails to address the issues of flexibility, ease of implementation, and application independence.

Protection at lower levels of the OSI stack, especially the data link layer, was also used in early communication systems, because it provides protocol-independent protection on specific untrusted links. However, data link layer protection is expensive to deploy on a large scale (protecting every link separately), potentially allowing man-in-the-middle attacks (the hijacking of a network session) on intermediate stations (routers).

Because of these limitations, Layer 3 has become the most popular level on which to apply cryptographic protection to network traffic.

# VPN Protocols

The table describes three VPN tunneling protocols: Layer 2 Tunneling Protocol (L2TP), Generic Routing Encapsulation (GRE), and IPsec.

VPN Protocols		
Protocol	Description	Standard
GRE	Generic Routing Encapsulation	RFC 1701, RFC 1702, RFC 2748
IPsec	Internet Protocol Security	RFC 4301

© 2006 Cisco Systems, Inc. All rights reserved. OMT v1.0—4-6

## GRE

This multiprotocol transport encapsulates IP and any other protocol packets inside IP tunnels.

With GRE tunneling, a Cisco router at each site encapsulates protocol-specific packets in an IP header, creating a virtual point-to-point link to Cisco routers at other ends of an IP cloud where the additional IP header is stripped off. GRE does not provide encryption and can be monitored with a protocol analyzer.

## IPsec

IPsec is the choice for secure corporate VPNs. IPsec is a framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers.

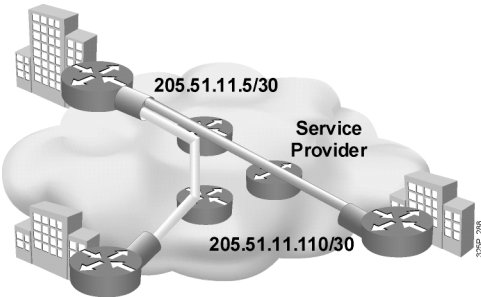
IPsec provides security services to handle negotiation of protocols and algorithms based on local policy and to generate the encryption and authentication keys to be used by IPsec.

# Implementing QoS with Preclassification

This topic describes the purpose of preclassification to support QoS in various VPN configurations.

## QoS Preclassify

- **VPNs are growing in popularity.**
- **The need to classify traffic within a traffic tunnel is also gaining importance.**
- **QoS preclassify is a Cisco IOS feature that allows packets to be classified before tunneling and encryption occur.**
- **Preclassification allows traffic flows to be adjusted in congested environments.**



© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—4.9

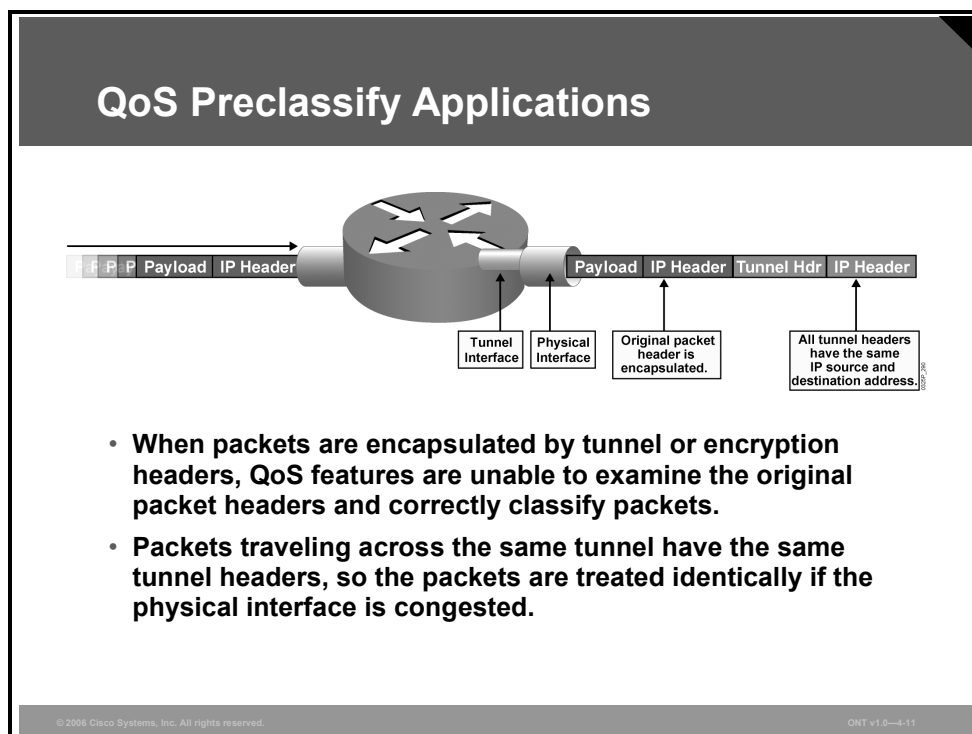
Quality of service (QoS) preclassify is designed for tunnel interfaces. When the feature is enabled, the QoS features on the output interface classify packets *before* encryption, allowing traffic flows to be managed in congested environments. The result is more effective packet tunneling.

The QoS preclassify feature provides a solution for making Cisco IOS QoS services operate in conjunction with tunneling and encryption on an interface. Cisco IOS software can classify packets and apply the appropriate QoS service before data is encrypted and tunneled. This allows service providers and enterprises to treat voice, video, and mission-critical traffic with a higher priority across service provider networks while using VPNs for secure transport



# QoS Preclassify Applications

This topic describes situations where preclassification is appropriate and identifies the VPN applications (IPsec, GRE, and L2TP) that support QoS preclassification.



When packets are encapsulated by a tunneling or encryption protocol, the original packet header is no longer available for examination.

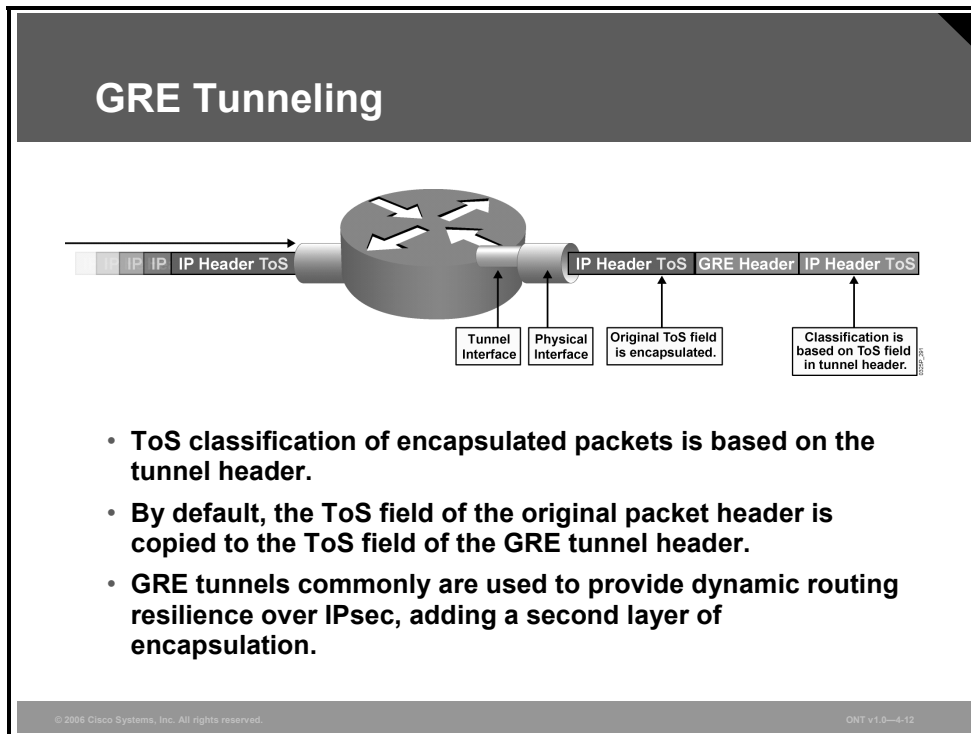
From the QoS perspective, providing differentiated levels of service is extremely difficult without the ability to examine the original packet header. The QoS markers normally found in the header of the IP packet must also be visible in the tunnel packet header, regardless of the type of tunnel in use.

The two primary tunneling protocols relevant to VPNs are these:

- IPsec
- GRE

# GRE Tunneling

GRE tunnels allow any protocol to be tunneled in an IP packet. Today, Cisco offers support for encapsulation of data using either IPsec or GRE. In either of these scenarios, Cisco IOS software offers the ability to copy the IP type of service (ToS) values from the packet header into the tunnel header. This feature allows the ToS bits to be copied to the tunnel header when the router encapsulates the packets.

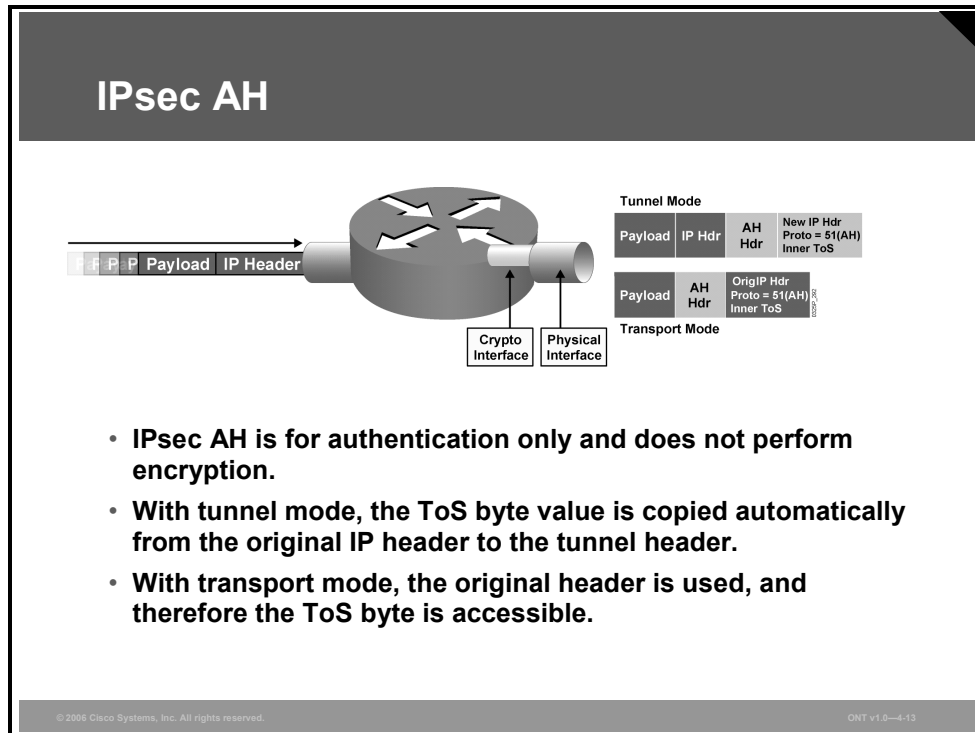


GRE tunneling allows routers between GRE-based tunnel endpoints to see the packet marking, improving the routing of premium service packets. Cisco IOS QoS technologies such as policy routing, weighted fair queuing (WFQ), and weighted random early detection (WRED) can operate on intermediate routers between GRE tunnel endpoints.

GRE tunnels are commonly used to provide dynamic routing resilience over IPsec. Normal IPsec configurations cannot transfer routing protocols, such as Enhanced Interior Gateway Routing Protocol (EIGRP) and Open Shortest Path First (OSPF), or non-IP traffic, such as Internetwork Packet Exchange (IPX) and AppleTalk.

# IPsec AH

IPsec does not define the specific security algorithms to use; rather, IPsec provides an open framework for implementing industry-standard algorithms.

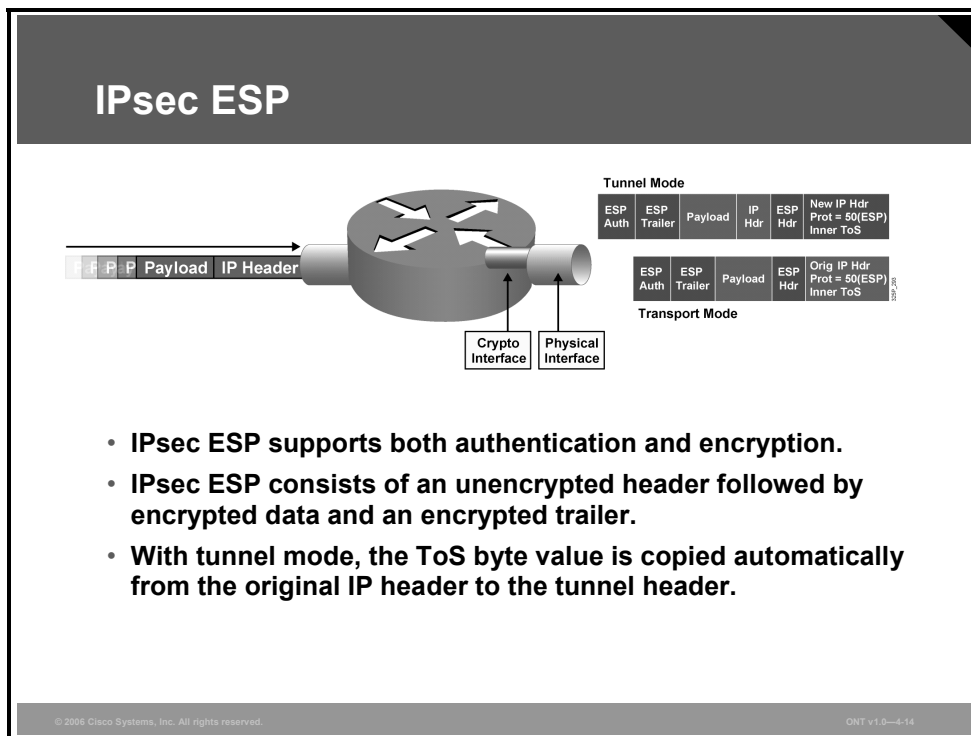


Authentication Header (AH) provides strong integrity and authentication for IP datagrams using the Secure Hash Algorithm (SHA) or Message Digest 5 (MD5) hash algorithm. AH can also provide nonrepudiation. The Internet Assigned Numbers Authority (IANA) has assigned protocol number 51 to AH. Thus, in the presence of an AH header with both tunnel mode and transport mode, the IP header uses a value of 51 in the protocol field.

With tunnel mode, the ToS byte value is copied automatically from the original IP header to the tunnel header.

# IPsec ESP

IPsec does not define the specific security algorithms to use; rather, IPsec provides an open framework for implementing industry-standard algorithms.



Encapsulating Security Payload (ESP) consists of an unencrypted header followed by encrypted data and an encrypted trailer. ESP can provide both encryption and authentication.

As with AH, ESP supports SHA and MD5 hash algorithms for authentication. ESP supports Data Encryption Standard (DES) and Triple-DES (3DES) as encryption protocols. The ESP header is at least 8 bytes. The IANA has assigned protocol number 50 to ESP. Thus, in the presence of (only) an ESP header with both tunnel mode and transport mode, the IP header uses a value of 50 in the protocol field.

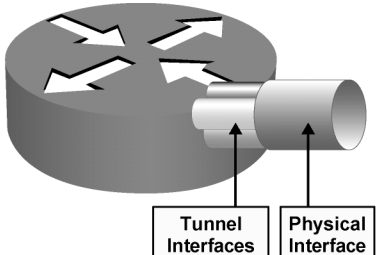
With tunnel mode, the ToS byte value is copied automatically from the original IP header to the tunnel header.

# QoS Preclassification Deployment Options

This topic describes some of the VPN applications that support QoS preclassification, and situations where preclassification is not appropriate.

## QoS Preclassification Deployment Options

- **Tunnel interfaces support many of the same QoS features as physical interfaces.**
- **In VPN environments, a QoS service policy can be applied to the tunnel interface or to the underlying physical interface.**
- **The decision about whether to configure the qos preclassify command depends on which header is used for classification.**



The diagram shows a 3D representation of a router. On the left, a circular face represents the router's front panel with several white arrows pointing outwards, symbolizing traffic exiting. On the right, a cylindrical tunnel interface is shown protruding from the router. Below the tunnel interface is a label 'Tunnel Interfaces' with an arrow pointing to it. To the right of the tunnel interface is a label 'Physical Interface' with an arrow pointing to the router's body. Below these labels are two text boxes: 'Where should the QoS policy be applied?' and 'When should the qos pre-classify command be used?'. A small vertical text '3207\_304' is on the right side of the diagram.

Where should the QoS policy be applied?

When should the qos pre-classify command be used?

© 2006 Cisco Systems, Inc. All rights reserved. OMT v1.0—4-16

Classification defines the process of matching one or more fields in a packet header in Layer 2, 3, or 4 and then placing that packet in a group or class of traffic. Using packet classification, network traffic can be partitioned into multiple priority levels or classes of service.

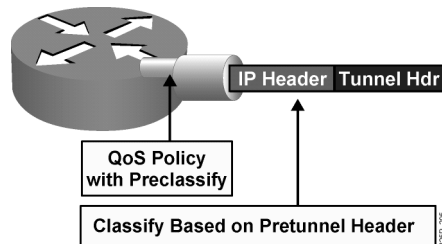
When configuring IPsec with GRE, the simplest classification approach is to match on IP precedence or differentiated services code point (DSCP) values. In addition, with the ToS byte preservation feature, the router automatically copies the ToS header value from the original IP packet to the encapsulating IP header when using IPsec in tunnel mode.

ToS byte preservation also applies to AH. Also note that ESP in transport mode retains the original IP header, and the original ToS value is transmitted even without ToS byte preservation. If packets arrive at the router without set IP precedence or DSCP values, class-based marking is used to re-mark the packet headers before encryption or encapsulation. When the packets reach the egress interface, the QoS output policy can match and act on the re-marked values.

Alternatively, traffic may need to be classified based on values other than IP precedence or DSCP. For example, packets may need to be classified based on IP flow or Layer 3 information, such as source and destination IP address. To do so, use the QoS for VPNs feature enabled with the **qos pre-classify** command.

## QoS Preclassification Deployment Options (Cont.)

- QoS preclassify allows access to the original IP header values.
- QoS preclassify is *not* required if classification based on original ToS values as this is copied by default to new header.



```
IPsec and GRE configuration:
!
crypto map static-crypt 1 ipsec-
  isakmp
  qos pre-classify
  set peer ....etc
!
interface Tunnel 0
  etc..
  qos pre-classify
  crypto map static-crypt
!
interface Ethernet 0/1
  service-policy output minbwto
  crypto map static-crypt
!
```

Note: ToS byte copying is done by the tunneling mechanism and not by the qos pre-classify command.

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0-4.17

The **qos pre-classify** command mechanism allows Cisco routers to make a copy of the inner IP header and to run a QoS classification before encryption, based on fields in the inner IP header. If the classification policy matches on the ToS byte, it is not necessary to use the **qos pre-classify** command, because the ToS value is copied to the outer header by default. In addition, a simple QoS policy that sorts traffic into classes based on IP precedence can be created. However, differentiating traffic within a class and separating it into multiple flow-based queues requires the **qos pre-classify** command.

You can apply a service policy to either the tunnel interface or to the underlying physical interface. The decision about where to apply the policy depends on the QoS objectives and on which header you need to use for classification, as follows:

- Apply the policy to the tunnel interface without **qos pre-classify** when you want to classify packets based on the pretunnel header.
- Apply the policy to the physical interface without **qos pre-classify** when you want to classify packets based on the post-tunnel header. In addition, apply the policy to the physical interface when you want to shape or police all traffic belonging to a tunnel and the physical interface supports several tunnels.
- Apply the policy to a physical interface and enable **qos pre-classify** when you want to classify packets based on the pretunnel header.

---

**Note** ToS byte copying is done by the tunneling mechanism and not by the **qos pre-classify** command.

---

# Configuring QoS Preclassify

The **qos pre-classify** Cisco IOS command enables the QoS preclassification feature.

## Configuring QoS Preclassify

```
router(config-if)#  
qos pre-classify
```

- Enables the QoS preclassification feature.
- This command is restricted to tunnel interfaces, virtual templates, and crypto maps.

GRE Tunnels

```
router(config)# interface tunnel0  
router(config-if)# qos pre-classify
```

IPSec Tunnels

```
router(config)# crypto map secured-partner  
router(config-crypto-map)# qos pre-classify
```

© 2006 Cisco Systems, Inc. All rights reserved. QNT v1.0—4-16

The **qos pre-classify** command can be applied to a tunnel interface, a virtual template interface, or a crypto map.

## Example

This example shows the configuration of the **qos pre-classify** command.

### QoS Preclassify: Example

```
class-map match-any branch110
  match access-group 110
!
policy-map branch-qos
  class branch110
    bandwidth 512
    police 256000
  !
!
interface Tunnel0
  ip address 192.168.16.110 255.255.255.0
  tunnel source serial0/0
  tunnel destination 205.51.11.5
  crypto map vpn
  qos pre-classify
!
crypto map vpn 10 ipsec-isakmp
  set peer 205.51.11.5
  set transform-set branch-vpn
  match address 110
  qos pre-classify
!
interface serial0/0
  ip address 205.51.11.110 255.255.255.252
  service-policy output branch-qos
  crypto map vpn
!
access-list 110 permit gre host
  205.51.11.110 host 205.51.11.5
```

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—4-19

On the serial0/0 interface on the branch router, there is an outgoing service policy that sets the bandwidth of the interface at 256 kbps and policing at a rate of 512 kbps. This policy is applied to any match in the class map branch 110.

A traffic tunnel has been built on interface serial0/0 (whose destination is the headquarters for this branch IP address 205.51.11.5). It is on this traffic tunnel that QoS preclassification has been configured.

The example configuration also shows that QoS preclassify has been successfully enabled on the crypto map named “vpn.” This crypto map has also been applied to serial0/0. If QoS preclassify is enabled only on the crypto map and not on the tunnel interface, the router will see one flow only, the GRE tunnel (protocol 47).



# Summary

This topic summarizes the key points that were discussed in this lesson.

## Summary

- **A VPN is defined as network connectivity deployed on a shared infrastructure with the same policies and security as a private network, and it offers encryption, data integrity and origin authentication.**
- **The QoS preclassify feature is designed for tunnel interfaces.**
- **IPsec and GRE support QoS preclassification. When packets are encapsulated by tunnel or encryption headers, QoS features are unable to examine the original packet headers and correctly classify the packets.**
- **QoS preclassify is enabled by the qos pre-classify Cisco IOS software command.**

© 2006 Cisco Systems, Inc. All rights reserved.

QNT v1.0—4-20



# Deploying End-to-End QoS

---

## Overview

Service level agreements (SLAs) define the basis of understanding between the two parties for delivery of the service itself. An SLA should contain clauses that define a specified level of service, support options, security options, incentive awards for service levels exceeded, or penalty provisions for services not provided. Before instituting such agreements with customers, IT service departments need to provide an adequate level of quality for these services. IT departments will try to improve quality of service (QoS), and the SLAs serve to keep quality at the agreed level and guarantee the QoS to the customer.

## Objectives

Upon completing this lesson, you will be able to describe the set of QoS mechanisms used to implement Cisco end-to-end QoS best practices in a typical enterprise network connected through a service provider that is providing Layer 3 IP services. This ability includes being able to meet these objectives:

- Describe IP QoS SLA and SLA examples
- Explain the typical network requirements within each functional block that makes up an end-to-end network
- Explain the best-practice QoS implementations and configurations within a campus LAN
- Explain the best-practice QoS implementations and configurations on WAN customer edge and provider edge routers
- Define the control plane and CoPP

# QoS SLAs

This topic describes IP QoS SLA and provides some SLA examples.

## QoS SLAs

- **QoS SLAs provide contractual assurance for meeting the traffic QoS requirements.**
- **QoS SLAs typically provide contractual assurance for parameters such as:**
  - **Delay (fixed and variable)**
  - **Jitter**
  - **Packet loss**
  - **Throughput**
  - **Availability**

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—4.3

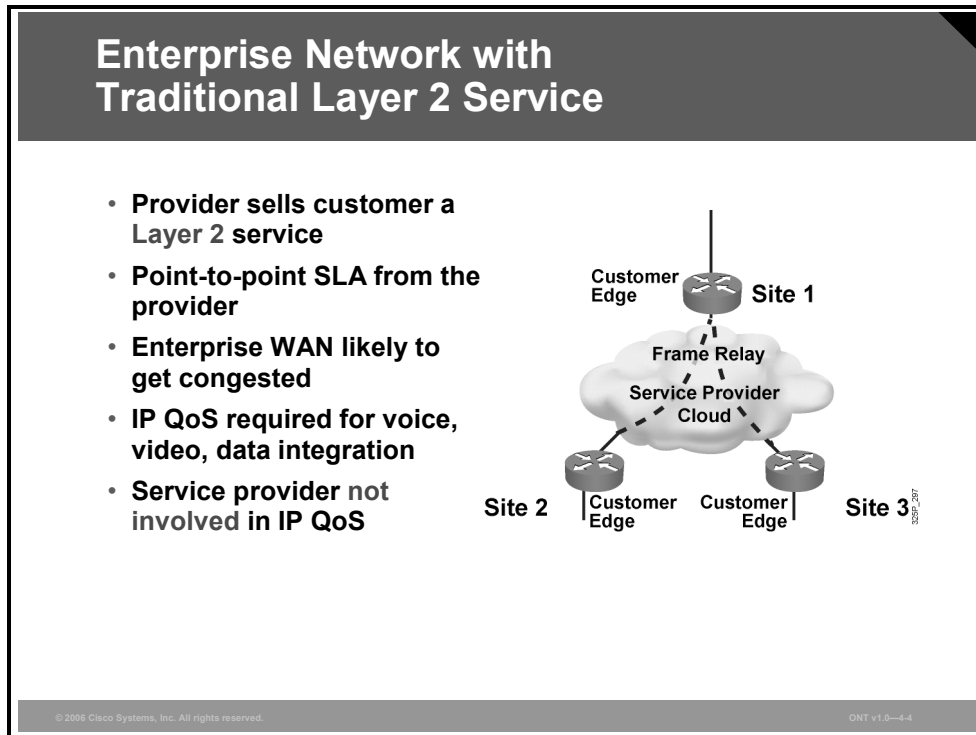
A service level agreement (SLA) stipulates the delivery and pricing of service levels and spells out penalties for shortfalls. SLAs can cover an assortment of data services, such as Frame Relay, leased lines, Internet access, web hosting, and so on. The best way to understand an SLA is to break it into two activities: negotiating the technology agreement and verifying compliance with the agreement.

A quality of service (QoS) SLA typically provides contractual assurance for parameters such as delay, jitter, packet loss, throughput, and availability.

With the rapid growth of new multimedia real-time applications such as IP telephony, web conferencing, and e-learning, IP QoS SLAs are becoming increasingly important for enterprise networks.

## Enterprise Network with Traditional Layer 2 Service

A service provider may provide only Layer 2 services to the enterprise customer. The customer edge routers at the various customer sites are interconnected by Frame Relay virtual circuits (VCs). These VCs can be fully meshed, partially meshed, or set up as hub and spokes, depending on the customer requirements.

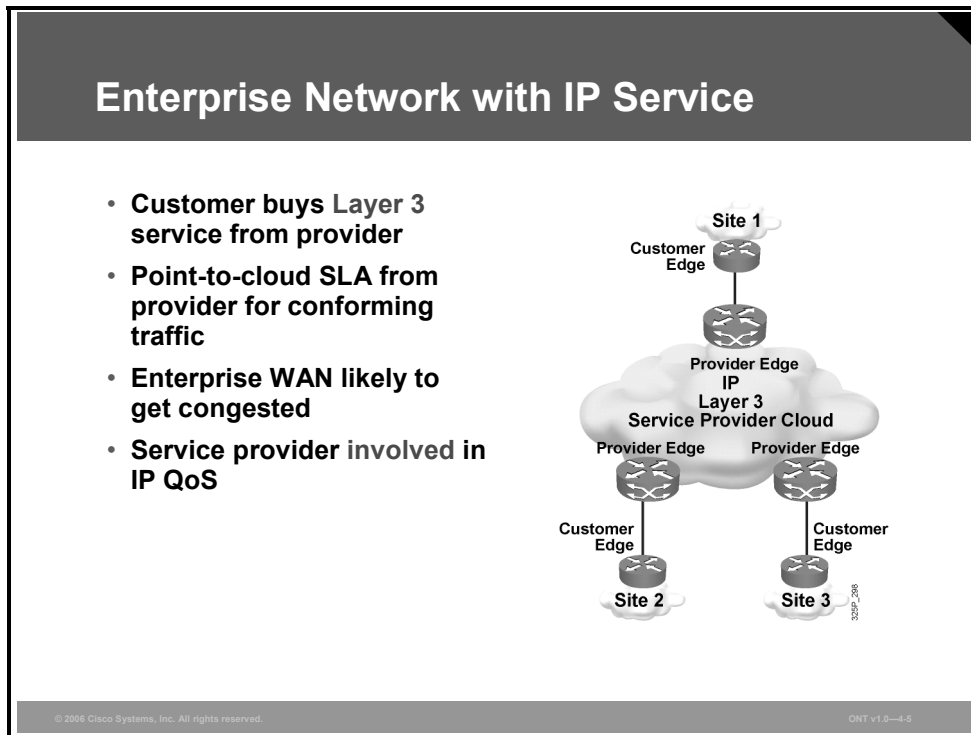


In this environment, the service provider is responsible only for the end-to-end Layer 2 VC connections. The service provider provides only a point-to-point SLA guarantee for each VC connection and is not involved with providing IP QoS to the customer.

To provide IP QoS for voice, video, and data integration over Frame Relay VCs, the customer must configure the proper QoS mechanisms, such as traffic shaping, low latency queuing (LLQ), FRF.12, and compressed Real-Time Transport Protocol (cRTP) at the WAN customer edge routers, because the Frame Relay WAN link is likely to become congested.

## Enterprise Network with IP Service

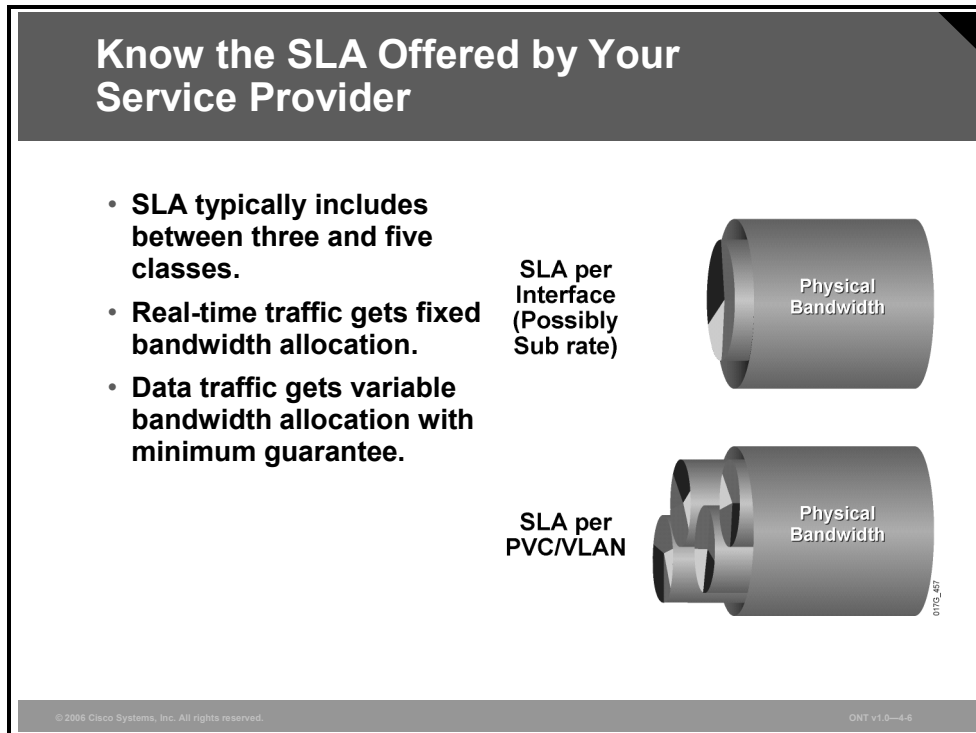
Another service provider may offer Layer 3 services to the enterprise customer. The customer edge routers at the various customer sites connect to the provider edge of the service provider router. From a particular customer site perspective, every IP address that is not located on-site is reachable via the service provider IP backbone network.



In this environment, the service provider can provide value-added IP services to the customer by providing SLAs for the conforming traffic from the customer. An SLA can, for example, divide customer traffic at the network edge into controlled latency, controlled load 1, and controlled load 2 classes and then provide IP QoS assurances to each traffic class conforming to the contractual rate over a Differentiated Services (DiffServ) IP backbone. For all nonconforming (exceeding) traffic, the service provider can re-mark and deliver all nonconforming traffic with best-effort service.

## Know the SLA Offered by Your Service Provider

The typical IP QoS SLA offered by most service providers often includes three to five traffic classes; for example, a real-time traffic class, a mission-critical data traffic class, one or two other data traffic classes, and a best-effort traffic class. The SLA for the real-time traffic class should be guaranteed a fixed maximum bandwidth, while the data traffic classes should be guaranteed a minimum bandwidth. Typically, the bandwidth allocation is configured as a percentage of the interface bandwidth. Each traffic class can also have a latency, delay, jitter, and packet-loss guarantee.



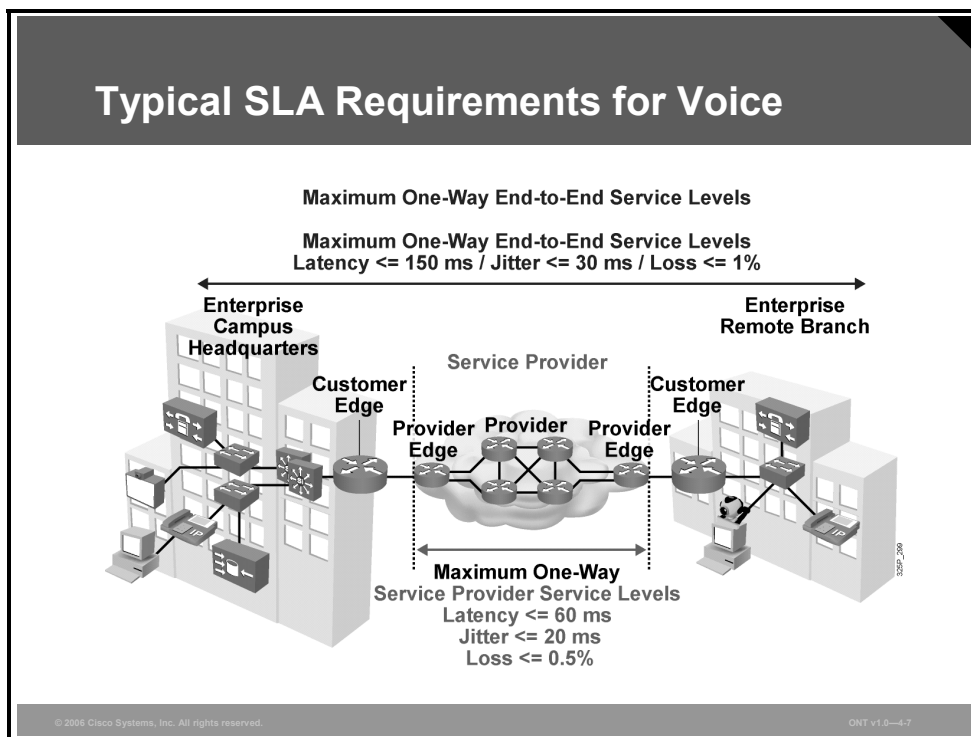
Between the customer edge and provider edge, there may be additional traffic classes that are used by the service providers only. For example, there may be a management traffic class for traffic such as Telnet or Simple Network Management Protocol (SNMP) from the service provider to the service provider-managed customer edge routers.

If a single physical interface is serving only one customer, the SLA is typically set up per interface. To provide easy bandwidth upgrades, service providers often install a high-speed link to the customer and then offer a subrate access.

If a single physical interface is serving many different customers, the SLA is typically set up per permanent virtual circuit (PVC) or per VLAN. To provide easy bandwidth upgrades, the service provider often installs a high-speed link to the customer and then offers a subrate access.

## Typical SLA Requirements for Voice

To meet the QoS requirements for the different traffic types, both the enterprise and the service provider must implement the proper QoS mechanisms to provide end-to-end QoS for the packets traversing a service provider IP network. In the figure, the enterprise headquarters and the enterprise branch office are connected to a service provider that is providing Layer 3 services.

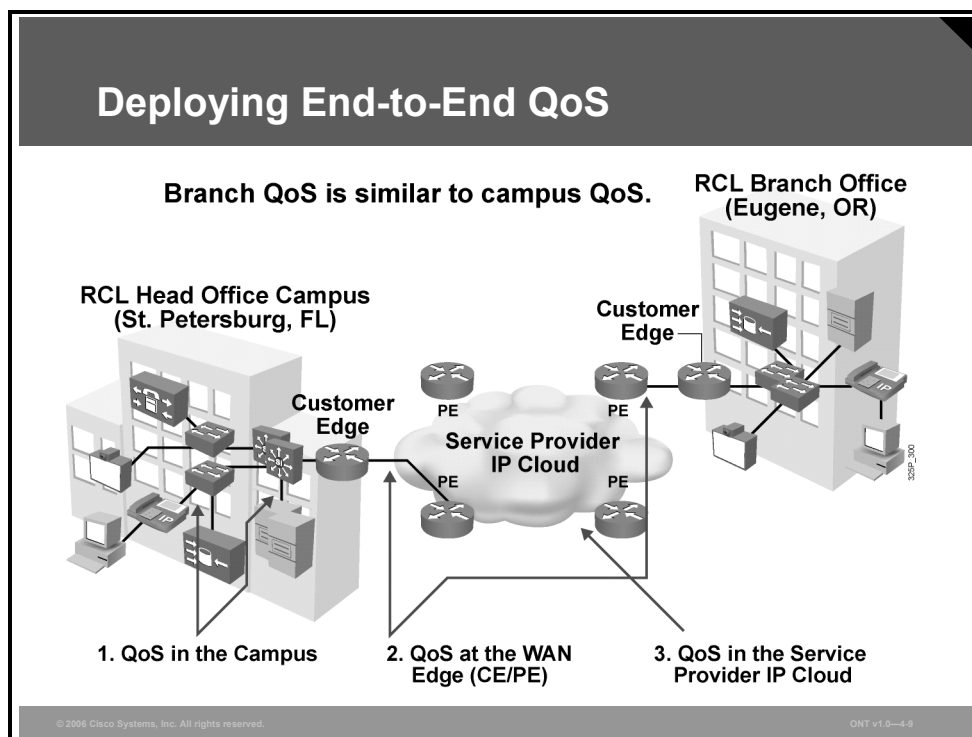


In this example, the service provider is providing an SLA for voice traffic with a latency of 60 ms or less, jitter of 20 ms or less, and packet loss of 0.5 percent or less. To meet the end-to-end QoS requirements for voice packets, the entire enterprise network must contribute less than 90 ms of delay—that is, 90 ms (enterprise network) + 60 ms (service provider network)  $\leq 150$  ms total one-way delay. Similarly, the jitter in the enterprise network must be less than 10 ms—that is, 10 ms + 20 ms  $\leq 30$  ms total one-way jitter. Finally, packet loss within the enterprise network must be less than 0.5 percent—that is, 0.5 percent + 0.5 percent  $\leq 1.0$  percent total packet loss.



# Deploying End-to-End QoS

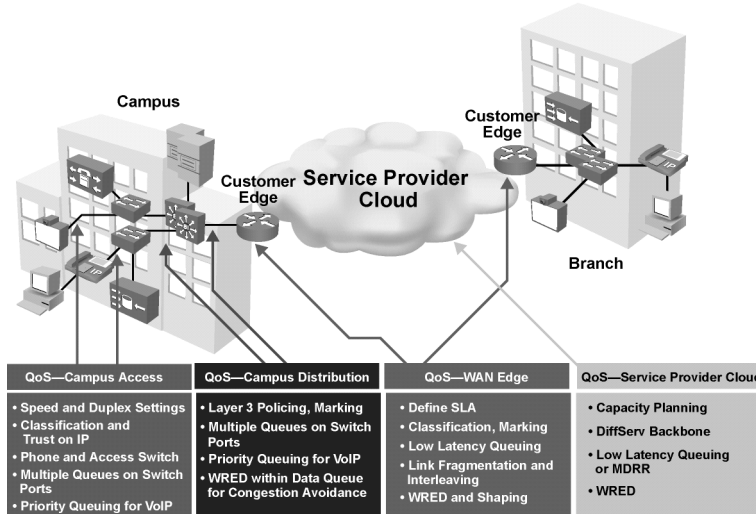
This topic describes the typical network requirements within each functional block (headquarter campus LAN, WAN edge, service provider backbone, and branch office) in an end-to-end network.



To meet the QoS requirements for different traffic types, both the enterprise and the service provider must implement the proper IP QoS mechanisms to provide end-to-end QoS for the packets traversing a service provider network. This means that at both customer locations, traffic classifications and marking need to be performed (for example, VoIP, data). Depending on the customer connection to the service provider, these markings can be mapped into Multiprotocol Label Switching (MPLS) Experimental (EXP) bits, for example, and given priorities. The provider now must guarantee correct transfer over the core to the branch office. The traffic arrives there with the same markings that were set at the head office, allowing again the classification that is needed for end-to-end QoS.

## Deploying End-to-End QoS (Cont.)

End-to-End QoS = Enterprise QoS + Service Provider QoS



© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—4-10

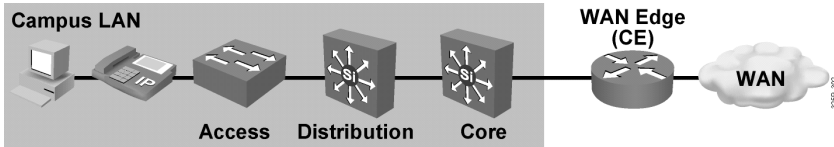
To provide end-to-end QoS, both the enterprise and service provider must implement the proper QoS mechanisms to ensure the correct per-hop behavior (PHB) for each traffic class across the whole network. In the past, IP QoS was not an issue in an enterprise campus network, where bandwidth is plentiful. But as more applications, such as IP telephony, videoconferencing, e-learning, and mission-critical data applications, are implemented in the campus, it has become evident that buffer management, not just bandwidth, is an issue that must be addressed. IP QoS functions such as classification, scheduling, and provisioning are now required within the campus to manage bandwidth and buffers to minimize loss, delay, and jitter.

This figure lists some of the requirements within the building blocks that constitute the end-to-end network. QoS at the campus access layer focuses on speed and duplex settings, classification, hardware requirements, and queuing. QoS at the campus distribution layer deals with policing, marking, and congestion avoidance. Complex QoS configurations typically occur at the WAN edge. In the IP core (service provider cloud), only congestion-management and congestion-avoidance mechanisms are in operation. Key QoS mechanisms used in an IP core include LLQ and weighted random early detection (WRED).

# Enterprise Campus QoS Implementations

This topic describes some of the best-practice QoS implementations and configurations within the campus LAN.

## Campus QoS General Guidelines



The diagram illustrates the network architecture for Campus QoS. It starts with a 'Campus LAN' section containing a computer and an IP phone. This connects to an 'Access' switch, followed by a 'Distribution' switch, and then a 'Core' switch. The traffic then flows through a 'WAN Edge (CE)' router to a 'WAN' cloud. The switches and router are represented by icons with 'S' and 'R' labels.

- **Multiple queues are required on all interfaces to prevent transmit queue congestion and drops.**
- **Voice traffic should always go into the highest-priority queue.**
- **Trust the Cisco IP phone CoS setting but not the PC CoS setting.**
- **Classify and mark traffic as close to the source as possible.**
- **Use class-based policing to rate-limit certain unwanted excess traffic.**

© 2006 Cisco Systems, Inc. All rights reserved. OMT v1.0—4-12

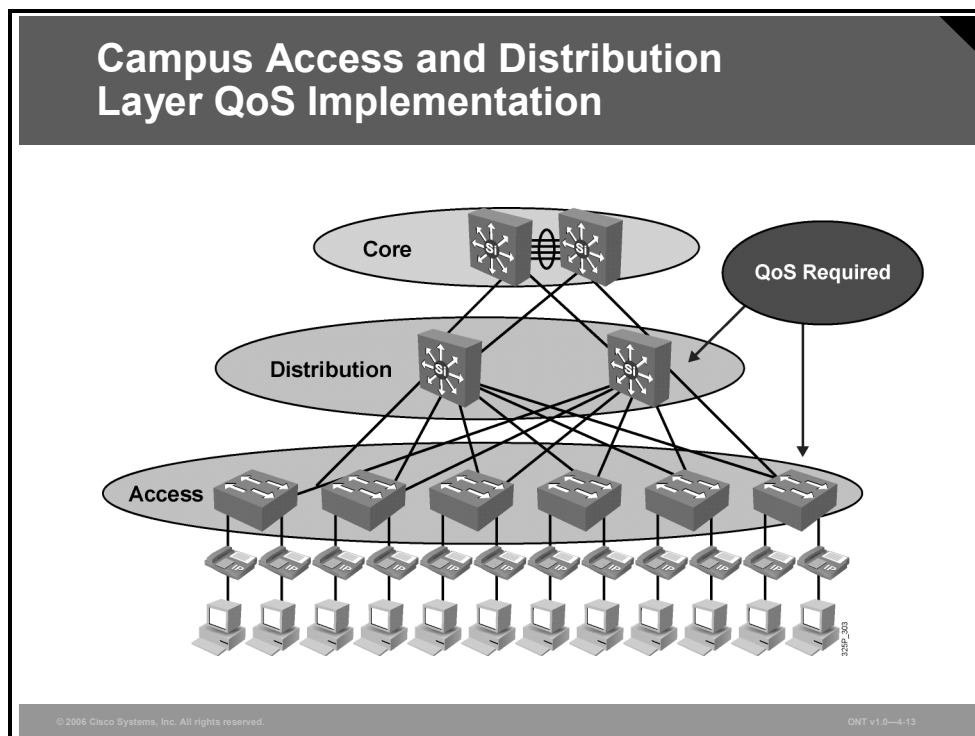
IP telephony, videoconferencing, e-learning, and mission-critical data applications are becoming more common in enterprise networks. IP QoS functions such as classification, scheduling, and provisioning are required within the campus to manage the switch output buffers to minimize packet loss, delay, and jitter. Some of the general guidelines when implementing campus QoS include the following:

- **Classify and mark the traffic as soon as possible:** This principle promotes end-to-end DiffServ PHBs. Sometimes endpoints can be trusted to set class of service (CoS) and differentiated services code point (DSCP) markings correctly, but this practice is not recommended because users can easily abuse provisioned QoS policies if they are permitted to mark their own traffic. For example, if DSCP Expedited Forwarding (EF) receives priority services throughout the enterprise, a user could easily configure the network interface card (NIC) on a PC to mark all traffic to DSCP EF, thus hijacking network priority queues to service non-real-time traffic. Such abuse could easily ruin the service quality of real-time applications (such as VoIP) throughout the enterprise.
- **Police unwanted traffic flows as close to their sources as possible:** There is little sense in forwarding unwanted traffic only to police and drop it at a subsequent node. This is especially the case when the unwanted traffic is the result of denial-of-service (DoS) or worm attacks. Such attacks can cause network outages by overwhelming network device processors with traffic.

- **Always perform QoS in hardware rather than software when a choice exists:** Cisco IOS routers perform QoS in software. This design places additional demands on the CPU, depending on the complexity and functionality of the policy. Cisco Catalyst switches, on the other hand, perform QoS in dedicated hardware ASICs, which does not tax their main CPUs to administer QoS policies. You can therefore apply complex QoS policies at Gigabit and 10 Gigabit Ethernet line speeds in these switches.
- **Establish proper trust boundaries:** For example, at the access layer switches, trust only the IP Phone CoS marking, not the PC CoS marking.
- **Classify real-time voice and video as higher-priority traffic:** Classify real-time voice and video traffic at a higher priority than data traffic.
- **Use multiple queues on the transmit interfaces:** Use multiple queues on the transmit interfaces to minimize the potential for dropped or delayed traffic caused by transmit buffer congestion.

# Campus Access and Distribution Layer QoS Implementation

The typical QoS configurations are required at the access and distribution layer switches.



It is quite rare under normal operating conditions for campus networks to suffer congestion. And if congestion does occur, it is usually momentary and not sustained, as at a WAN edge. However, critical applications like VoIP require service guarantees regardless of network conditions. *The only way to provide service guarantees is to enable queuing at any node that has the potential for congestion*—regardless of how rarely, in fact, congestion may occur. The potential for congestion exists in campus uplinks because of oversubscription ratios and speed mismatches in campus downlinks (for example, Gigabit Ethernet to Fast Ethernet links). The only way to provision service guarantees in these cases is to enable queuing at these points.

Queuing helps to meet network requirements under normal operating conditions, but enabling QoS within the campus is even more critical under abnormal network conditions, such as DoS and worm attacks. During such conditions, network traffic may increase exponentially until links are fully utilized. Without QoS, the worm-generated traffic drowns out applications and causes denial of service through unavailability. Enabling QoS policies within the campus, as detailed later in this lesson, maintains network availability by protecting and servicing critical applications, such as VoIP, and even best-effort traffic.

The intrinsic interdependencies of network QoS, high availability, and security are clearly manifest in such worst-case scenarios.

So where is QoS required in campus? Access switches require the following QoS policies:

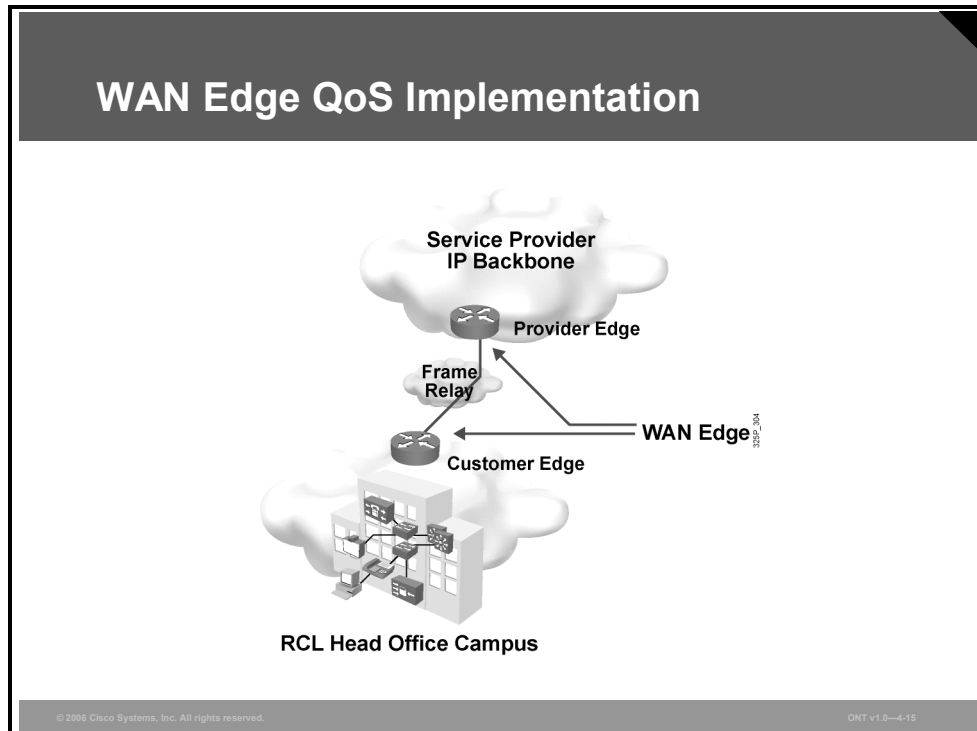
- Appropriate (endpoint-dependent) trust policies and classification and marking policies
- Policing and markdown policies
- Queuing policies

Distribution and core switches require the following QoS policies:

- DSCP trust policies
- Queuing policies
- Optional per-user microflow policing policies (only on supported platforms)

# WAN Edge QoS Implementations

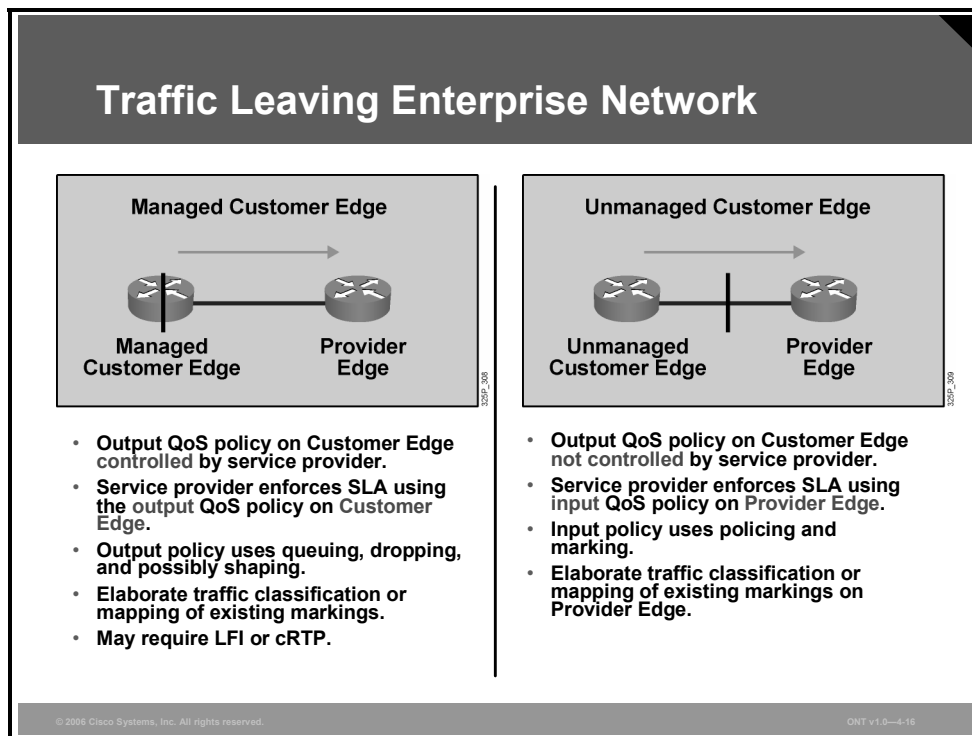
This topic describes some of the best-practice QoS implementations and configurations on WAN customer edge and provider edge routers.



Basically, the routers that terminate the WAN link between the customer edge and the provider edge are the devices that require a high configuration effort from the network administrators. Several WAN technologies, such as Frame Relay and ATM, and QoS features, such as LLQ, traffic shaping, and compression, need to be enabled and tuned to provide a high level of QoS.

# Traffic Leaving Enterprise Network

The QoS requirements on the customer edge and provider edge routers will differ, depending on whether the customer edge is managed by the service provider.



For traffic leaving the enterprise customer edge router and moving toward the service provider edge router, the figure illustrates the general QoS requirements on the customer edge and provider edge routers.

For managed customer edge service, the WAN edge output QoS policy on the customer edge will be managed and configured by the service provider.

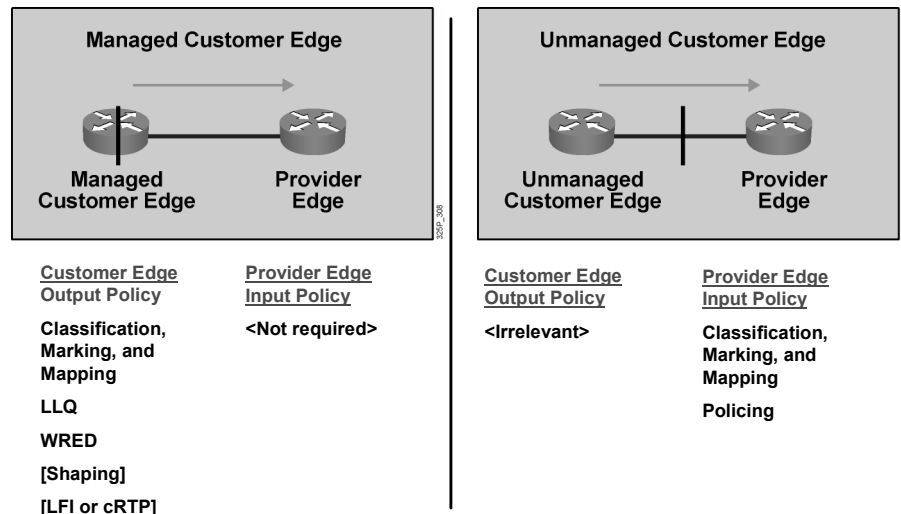
For unmanaged customer edge service, the WAN edge output QoS policy on the customer edge will be managed and configured by the enterprise customer.

For managed customer edge service, the service provider can enforce the SLA for each traffic class using the output QoS policy on the customer edge. For example, you can use LLQ or class-based weighted fair queuing (CBWFQ) to give a maximum bandwidth guarantee to the real-time voice and video traffic class, give a minimum bandwidth guarantee to the data traffic class, and use class-based shaping to provide a maximum rate limit to each data traffic class.

For unmanaged customer edge service, because the service provider has no control over the customer edge, the service provider can enforce the SLA for each traffic class only at the input of the provider edge router. For example, you can use class-based policing to limit the input traffic rate of the different traffic classes and to re-mark the exceeding traffic.



## Traffic Leaving Enterprise Network (Cont.)

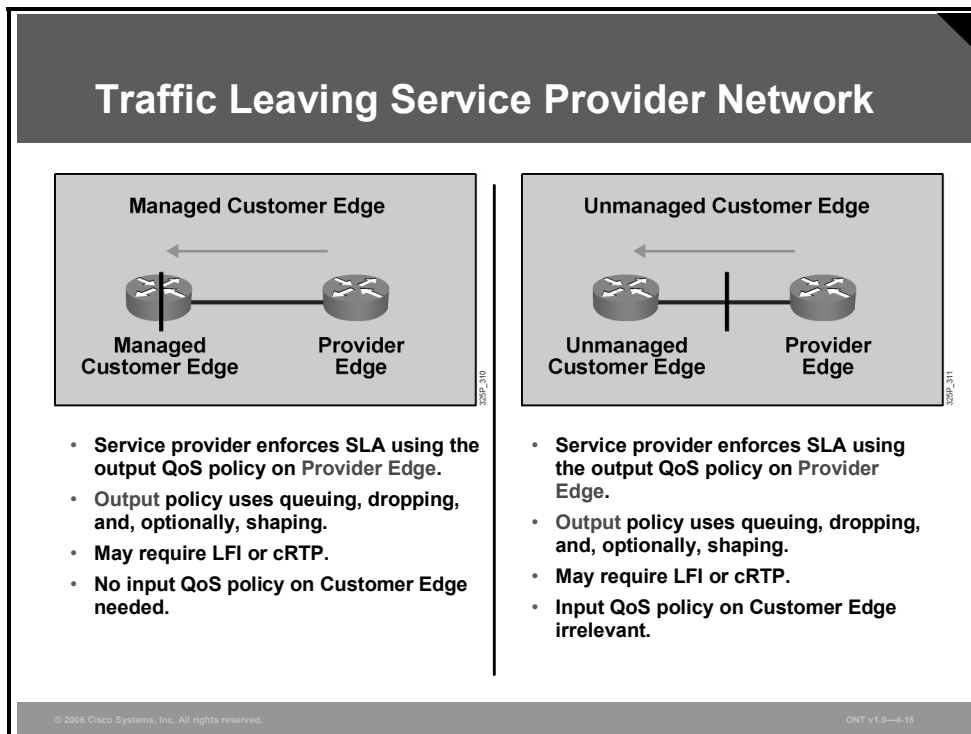


For an unmanaged customer edge, the customer edge output policy is managed and configured by the enterprise customer; therefore, it is irrelevant to the service provider. At the provider edge input interface, the service provider has a policy to classify, mark, or map the traffic. The service provider also typically implements traffic policing to limit the input traffic rate from the enterprise customer so that the traffic rate does not exceed the contractual rate specified in the SLA.

For a managed customer edge, the customer edge output policy is managed and configured by the service provider. The service provider typically has an output policy on the customer edge router to classify and mark the traffic exiting the customer edge router. LLQ or CBWFQ, along with WRED, are used for congestion management and congestion avoidance. To compensate for speed mismatch or oversubscription, traffic shaping may be required. To improve link efficiency, link fragmentation and interleaving (LFI) and cRTP are used for lower-speed links.

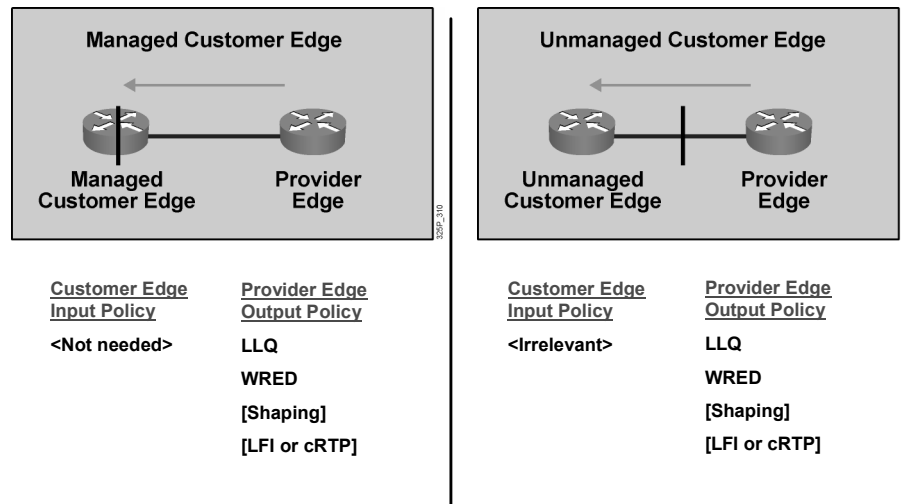
# Traffic Leaving Service Provider Network

For traffic leaving the service provider edge router toward the enterprise customer edge router, the figure illustrates the general QoS requirements on the customer edge and provider edge routers.



For both managed and unmanaged customer edge service, the service provider can enforce the SLA for each traffic class using the output QoS policy on the provider edge. Queuing and compression can be enabled. An input policy is not needed.

## Traffic Leaving Service Provider Network (Cont.)



For traffic leaving the service provider edge router toward the enterprise customer edge router, the figure illustrates the QoS mechanisms that are commonly implemented at the provider edge router.

For both managed and unmanaged customer edge service, the service provider typically has an output policy on the provider edge router using LLQ or CBWFQ, along with WRED, for congestion management and congestion avoidance. To compensate for speed mismatch or oversubscription, traffic shaping may be required. To improve the link efficiency, LFI and cRTP are used for lower-speed links.

A customer edge input policy is not required for managed and unmanaged customer edge services.

## Example: Managed Customer Edge with Three Service Classes

In this example, the service provider is implementing an IP DiffServ backbone and is offering three traffic classes with different SLAs for each:

- **Real-time (VoIP, interactive video, call signaling):** The real-time traffic class is intended for voice traffic, interactive video, and call signaling. This class has a maximum bandwidth limit, a low latency, and no loss guarantee.
- **Critical data (routing, mission-critical data, transactional data and network management):** The critical data traffic class is intended for mission-critical traffic. This class has a minimum bandwidth and a low loss guarantee.
- **Best-effort:** The default (best effort) traffic class is intended for all other traffic. This class has no guarantee.

### Managed Customer Edge with Three Service Classes: Example

- The service provider in this example is offering managed customer edge service with three service classes:
  - Real-time (VoIP, interactive video, call signaling): Maximum bandwidth guarantee, low latency, no loss
  - Critical data (routing, mission-critical data, transactional data, and network management): Minimum bandwidth guarantee, low loss
  - Best-effort: No guarantees (best effort)
- Most DiffServ deployments use a proportional differentiation model:
  - Rather than allocate absolute bandwidths to each class, service provider adjusts relative bandwidth ratios between classes to achieve SLA differentiation.

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—4.20

When you are implementing LLQ or CBWFQ, the bandwidth guarantees can be specified in kilobits per second, in a percentage of the available bandwidth, or in a percentage of the remaining available bandwidth. Most DiffServ deployments today use a proportional differentiation model where the bandwidth guarantees are configured as a percentage instead of a fixed bandwidth in kilobits per second.

For example, with three traffic classes, the bandwidth allocation can be divided as follows:

- The real-time (VoIP, interactive video, call signaling) class LLQ can have a maximum bandwidth guarantee of 35 percent of the link bandwidth.
- The critical data (routing, mission-critical data, transactional data and network management) class can have a minimum bandwidth guarantee of 40 percent of the link bandwidth after the LLQ is serviced.
- The best-effort (default) class can have a minimum bandwidth guarantee of whatever is left—in this case, 25 percent of the link bandwidth after the LLQ is serviced.

---

<b>Note</b>	Extensive testing and production-network customer deployments have shown that limiting the sum of all LLQs to 35 percent is a conservative and safe design ratio for merging real-time applications with data applications. The 35-percent limit for the sum of all LLQs is simply a best-practice design recommendation; it is not a mandate. In some cases, specific business objectives cannot be met while holding to this recommendation. In such cases, enterprises must provision according to their detailed requirements and constraints. However, it is important to recognize the trade-offs involved in overprovisioning LLQ traffic with respect to the negative performance impact on data application response times.
-------------	--

---

# WAN Edge Design

For the real-time traffic class, VoIP packets will be marked with EF and go into the LLQ with these parameters:

- The LLQ will be policed and have a maximum bandwidth of 35 percent of the committed information rate (CIR).
- All excess traffic will be dropped.
- The call-signaling traffic (5 percent) will share the LLQ with the VoIP bearer traffic.

WAN Edge Design	
Class	Parameters
Real-time (VoIP)	<ul style="list-style-type: none"><li>• Packet marked EF class and sent to LLQ</li><li>• Maximum bandwidth = 35% of CIR, policed</li><li>• Excess dropped</li></ul>
Real-time (call-signaling)	<ul style="list-style-type: none"><li>• VoIP signaling (5%) shares the LLQ with VoIP traffic</li></ul>
Critical Data	<ul style="list-style-type: none"><li>• Allocated 40% of remaining bandwidth after LLQ has been serviced</li><li>• Exceeding or violating traffic re-marked</li><li>• WRED configured to optimize TCP throughput</li></ul>
Best-effort	<ul style="list-style-type: none"><li>• Best-effort class sent to CBWFQ</li><li>• Allocated 23% of remaining bandwidth after LLQ has been serviced</li><li>• WRED configured to optimize TCP throughput</li></ul>
Scavenger	<ul style="list-style-type: none"><li>• Best-effort class sent to CBWFQ</li><li>• Whatever is left = 2% of remaining bandwidth</li></ul>

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—4-21

For the critical data traffic class, the packets will be marked with Assured Forwarding (AF) 31 and go into the CBWFQ with these class parameters:

- The class will be policed and have a minimum bandwidth guarantee of 40 percent of the remaining available bandwidth.
- All exceeding and violating traffic will be re-marked and then sent.
- WRED will be used on this traffic class to optimize TCP throughput.

For the best-effort traffic class, the packets will be marked with class selector 0 (CS0) and go into the CBWFQ with these class parameters:

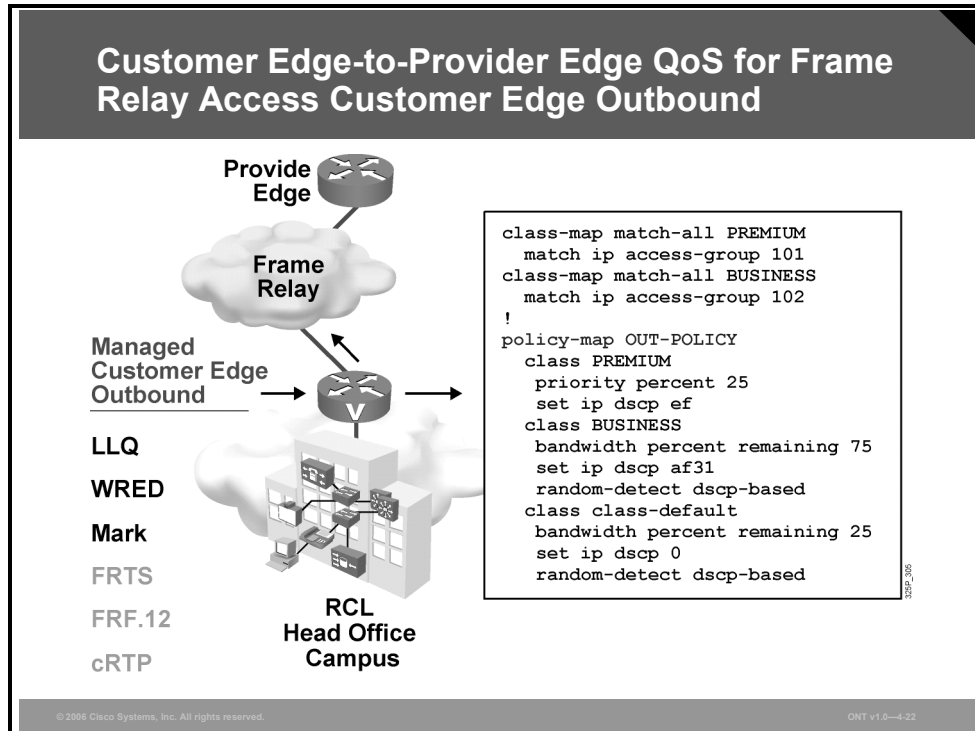
- The class will be policed and have a minimum bandwidth guarantee of 23 percent of the remaining available bandwidth.
- WRED will be used on this traffic class to optimize TCP throughput.

For the scavenger traffic class, the packets will be marked with CS1, with these class parameters:

- The class is not policed and has a minimum bandwidth guarantee of 2 percent of the remaining available bandwidth.

# Customer Edge-to-Provider Edge QoS for Frame Relay Access: Customer Edge Outbound

The traffic policy called “OUT-POLICY” is configured with three service provider traffic classes to provide the LLQ or CBWFQ and WRED. Each traffic class bandwidth guarantee is configured using a percentage rather than a fixed bandwidth in kilobits per second.

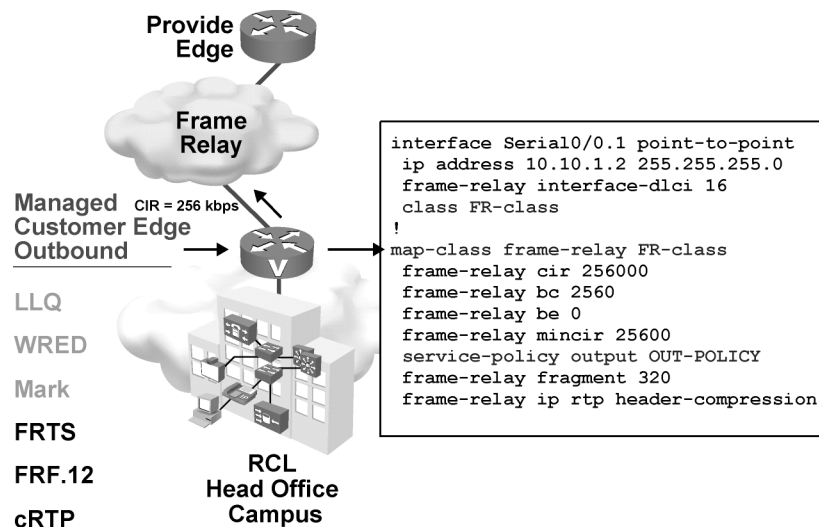


Both the enterprise voice-bearer and voice-control traffic will be matched using access control list (ACL) 101 and will be serviced by the LLQ with a maximum bandwidth guarantee of 25 percent of the link bandwidth.

The enterprise mission-critical traffic will be matched using ACL 102 and have a minimum guaranteed bandwidth of 75 percent of the remaining available bandwidth after the LLQ has been serviced.

All other traffic from the enterprise will be classified into the default class and have a minimum guaranteed bandwidth of 25 percent of the remaining available bandwidth after the LLQ has been serviced.

## Customer Edge-to-Provider Edge QoS for Frame Relay Access Customer Edge Outbound (Cont.)



In this case, the customer edge and provider edge link is a Frame Relay link, and traffic shaping is implemented on the PVC using Frame Relay traffic shaping (FRTS).

FRTS is configured using a Frame Relay map class with a CIR of 256 kbps, a committed burst (Bc) of 2560 bits, an excess burst (Be) of 0 (no bursting), and a minimum CIR (mincir) of 256 kbps. The CIR is the rate at which you normally want to send when there is no congestion. The CIR needs to be the remote end-link speed or the actual CIR of the VC. The Bc is the amount that you will send per time interval. The CIR and Bc will be used to compute a committed time interval (Tc), where  $Tc = Bc / CIR$ . For FRTS, the command-line interface (CLI) will only allow Bc values that would result in a Tc that is between 10 ms and 125 ms. A recommended value for Tc is 10 ms.

To get a Tc of 10 ms, the Bc should be set to 1/100 of the CIR. The mincir parameter is required, because Cisco IOS software will only allow 75 percent of the mincir to be reserved by the QoS policy applied to the FRTS map class. If the mincir is not configured, it will default to 50 percent of the CIR, which is not desired.

FRF.12 fragmentation and interleaving and cRTP are also enabled within the Frame Relay map class. The fragment size in bytes is set to derive a maximum delay of 10 ms to 15 ms.

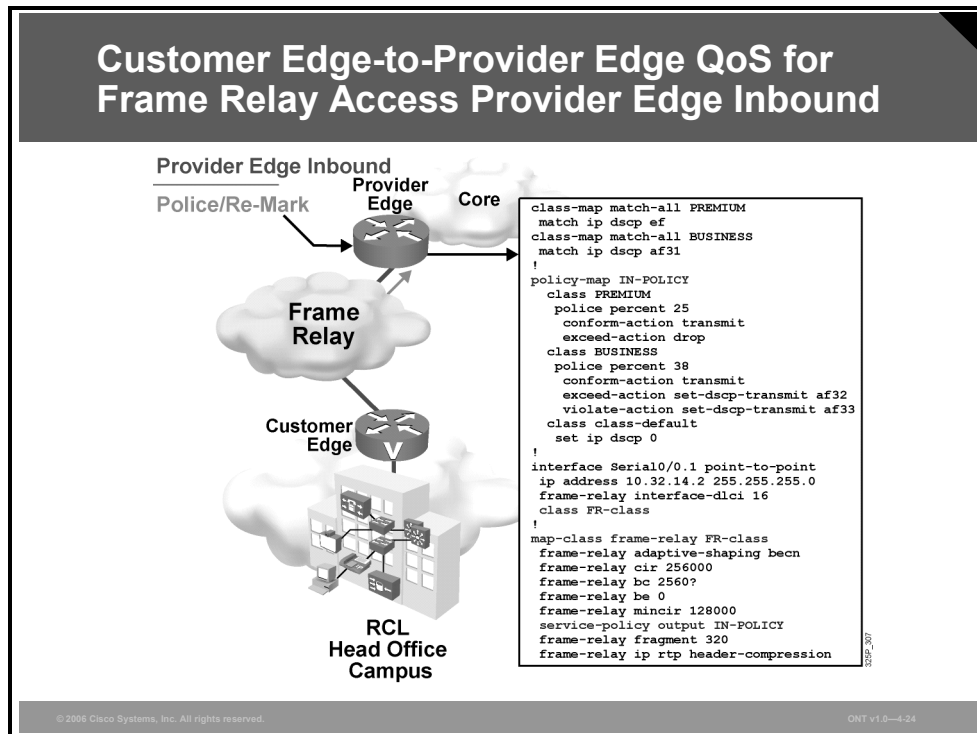
The fragment size should be the same on both ends.

The OUT-POLICY traffic policy is applied within the Frame Relay map class.



# Customer Edge-to-Provider Edge QoS for Frame Relay Access: Provider Edge Inbound

This output shows the QoS configurations on the ingress provider edge router inbound interface to implement the required QoS policy for each of the three service provider traffic classes.



On the provider edge, a traffic policy called “IN-POLICY” is configured to provide the required class-based policing. For the premium class, the rate limit is set to 25 percent of the link bandwidth. All exceeding premium-class traffic is dropped. For the business class, the rate limit is set to 38 percent of the link bandwidth. All exceeding and violating business-class traffic is re-marked with a higher drop probability and then sent. The default class is not policed.

The IN-POLICY traffic policy is applied within the Frame Relay map class.

# What Is CoPP?

This topic explains what the Control Plane Policing (CoPP) feature is and how it relates to QoS.

## What Is CoPP?

- **The CoPP feature allows users to configure a QoS filter that manages the traffic flow of control plane packets to protect the control plane against DoS attacks.**
- **CoPP has been available since Cisco IOS Software Release 12.2(18)S.**

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—4-26

Infrastructure attacks are becoming increasingly common, highlighting the need for infrastructure protection. The CoPP feature allows users to configure a QoS filter that manages the traffic flow of control plane packets to protect the control plane of Cisco IOS routers and switches against reconnaissance and DoS attacks. In this way, the control plane can help maintain packet forwarding and protocol states despite an attack or a heavy traffic load on the router or switch.

By protecting the Route Processor, CoPP helps ensure router and network stability during an attack. For this reason, a best-practice recommendation is to deploy CoPP as a key protection mechanism.

The CoPP feature was introduced in Cisco IOS Software Release 12.2(18)S.

# Cisco Router Planes

The vast majority of traffic travels through the router via the data plane; however, the Route Processor must handle certain packets, such as routing updates, keepalives, and network management. This traffic is often referred to as control and management plane traffic.

## Cisco Router Planes

- **A Cisco router is divided into four functional planes:**
  - **Data plane**
  - **Management plane**
  - **Control plane**
  - **Service plane**
- **Any service disruption to the route processor or the control and management planes can result in business-impacting network outages.**

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—4-27

Because the Route Processor is critical to network operations, any service disruption to the Route Processor or the control and management planes can result in business-impacting network outages. A DoS attack targeting the Route Processor, which can be perpetrated either inadvertently or maliciously, typically involves high rates of punted traffic that result in excessive CPU utilization on the Route Processor itself. This type of attack, which can be devastating to network stability and availability, may display the following symptoms:

- High Route Processor CPU utilization (near 100 percent)
- Loss of line protocol keepalives and routing protocol updates, leading to route flaps and major network transitions
- Slow or completely unresponsive interactive sessions via the command-line interface (CLI) due to high CPU utilization
- Route Processor resource exhaustion, such as memory and buffers that are unavailable for legitimate IP data packets
- Packet queue backup, which leads to indiscriminate drops (or drops due to lack of buffer resources) of other incoming packets

CoPP addresses the need to protect the control and management planes, ensuring routing stability, availability, and packet delivery. It uses the dedicated **control-plane** configuration command via the Cisco Modular QoS CLI (MQC) to provide filtering and rate-limiting capabilities for control plane packets.

# CoPP Deployment

CoPP leverages the MQC to define traffic classification criteria and to specify configurable policy actions for the classified traffic. Traffic of interest must first be identified via class maps, which are used to define packets for a particular traffic class. After classification, enforceable policy actions for the identified traffic are created with policy maps. The **control-plane** global command allows the control plane service policies to be attached to control plane itself.

## CoPP Deployment

- To deploy CoPP, take the following steps:
  1. Define a packet classification criteria.
  2. Define a service policy.
  3. Enter control-plane configuration mode.
  4. Apply QoS policy.
- Use MQC for configuring CoPP.

© 2006 Cisco Systems, Inc. All rights reserved. ONT v1.0—4-23

There are four steps required to configure CoPP:

- Step 1** Define a packet classification criteria.
- Step 2** Define a service policy.
- Step 3** Enter control-plane configuration mode.
- Step 4** Apply QoS policy.

## CoPP Policy and MQC

The MQC provides a flexible interface for creating service policies. Traffic can be identified via the **class-map** command and be dropped or permitted access to the route processor.

The MQC also permits multiple match criteria within a class-map configuration. The router needs to determine how packets are evaluated when there are multiple match criteria within a single class. Packets must either meet all of the specified match criteria (match all) or any one of the match criteria (match any) to be considered a member of the class. Traffic destined to the undesirable class should follow a match-any classification scheme; traffic that has any of the match criteria specified in this class may be dropped.

## CoPP Example

The example shows how to apply a QoS policy for aggregate control plane services to Telnet traffic transmitted from the control plane.

### CoPP Example

```
access-list 140 deny tcp host 10.1.1.1 any eq telnet
access-list 140 deny tcp host 10.1.1.2 any eq telnet
access-list 140 permit tcp any any eq telnet
!
class-map telnet-class
match access-group 140
!
policy-map control-plane-in
class telnet-class
  police 80000 conform transmit exceed drop
!
control-plane slot 1
service-policy input control-plane-in
```

© 2006 Cisco Systems, Inc. All rights reserved. OMT v1.0--4.29

The example shows how to configure rate limiting (on input) for distributed control plane traffic. QoS policy is applied to the data plane to perform distributed control plane services on packets destined for the control plane from the interfaces on the line card in slot 1. Trusted hosts are configured with source addresses 10.1.1.1 and 10.1.1.2 to forward Telnet packets to the control plane without constraint, while allowing all remaining Telnet packets that enter through slot 1 to be policed at the specified rate. The MQC is used to match the traffic, limit the traffic, and apply the policy to the control plane (on input) in slot 1 at the end.

# Summary

This topic summarizes the key points that were discussed in this lesson.

## Summary

- **An SLA stipulates the delivery and pricing of numerous service levels. SLAs cover an assortment of data services, such as Frame Relay, leased lines, Internet access, web hosting, and so on.**
- **There are QoS requirements for different traffic types. Both the enterprise and the service provider must implement the proper IP QoS mechanisms to provide end-to-end QoS.**
- **General guidelines for enterprise QoS implementations are as follows:**
  - **Use robust switching design.**
  - **Use buffer management.**
  - **Use multiple queues.**
  - **Ensure that voice traffic always gets the highest priority.**
  - **Trust the Cisco IP phone CoS setting.**
  - **Classify and mark traffic as close to the source as possible.**
  - **Use class-based policing.**

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—4-30

## Summary (Cont.)

- **On a Customer Edge-to-Provider Edge WAN link, LLQ or CBWFQ, traffic shaping, cRTP, and LFI are typically required.**
- **CoPP is a hardware-independent mechanism for defining and implementing sophisticated router protection schemes.**
- **CoPP is easily deployed by leveraging the existing MQC infrastructure.**

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—4-31

# Module Summary

This topic summarizes the key points that were discussed in this module.

## Module Summary

- DiffServ allows packet classification based on DSCP values.
- Layer 2 and Layer 3 have different methods for packet classification and marking.
- NBAR is a tool for class-based packet classification and marking.
- Queuing algorithms like FIFO, PQ, and round robin are basic packet dispatcher systems.
- Queuing algorithms like WFQ, CBWFQ, and LLQ combine several queuing techniques into advanced packet dispatcher systems.
- Congestion-avoidance mechanisms like RED, WRED, and CBWRED monitor network traffic loads to anticipate and avoid congestion before it becomes a problem.

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—4-1

## Module Summary (Cont.)

- Traffic policing and traffic shaping are used to protect the network resources from malicious connections and to enforce the compliance of every connection to its negotiated SLA.
- Compression is the process of encoding information using fewer bits.
- Packets entering IPsec or GRE tunnels need to be preclassified for QoS to work.
- End-to-end QoS is the strategy to deploy full QoS coverage.

© 2006 Cisco Systems, Inc. All rights reserved.

ONT v1.0—4-2

Packet classification is the process of identifying traffic and categorizing it into different classes. Packet classification allows some packets to be handled more quickly or with a higher priority than other packets. The Differentiated Services (DiffServ) model allows packet classification based on differentiated services code point (DSCP) values. Depending on the Layer 2 and Layer 3 technologies, these classifications can be mapped or copied into the appropriate field. To ease the process of packet classification and marking, Network-Based Application Recognition (NBAR) can be used. After classification and marking, the packets need to be queued with a basic queuing system (FIFO, priority queuing [PQ], or round robin) or an advanced queuing system (weighted fair queuing [WFQ], class-based weighted fair queuing [CBWFQ], or low latency queuing [LLQ]). Each of these queuing systems has its own scope of application. Random early detection (RED), weighted random early detection (WRED), and class-based WRED (CBWRED) mechanisms are used to monitor network traffic loads to anticipate and avoid congestion before it becomes a problem.

To follow service level agreements (SLAs) between Internet service providers (ISPs) and customers, traffic shaping and traffic policing mechanisms are used. Compression algorithms can optimize the usage of slow WAN links.

When packets are transported in IPsec or Generic Routing Encapsulation (GRE) tunnels, they need to be preclassified to make use of QoS and its benefits. The recommended strategy to deploy full QoS coverage is end-to-end QoS deployment.



# Module Self-Check

Use the questions here to review what you learned in this module. The correct answers and solutions are found in the Module Self-Check Answer Key.

- Q1) In the DiffServ model, the DSCP is used for \_\_\_\_\_ IP packets. (Source: Introducing Classification and Marking)
- A) deleting
  - B) marking
  - C) shaping
  - D) dropping
- Q2) Can MPLS EXP bits be mapped in the IEEE 801.2p field of an Ethernet frame? (Source: Introducing Classification and Marking)
- A) Yes, but they need to be converted to the ToS field first.
  - B) Yes, they can be copied directly to the IEEE 802.1p field.
  - C) Yes, but they need to be converted to CoS values first.
  - D) Yes, but they need to be converted to DSCP values first.
- Q3) Can NBAR be used to detect and classify traffic flows on tunnel or encrypted interfaces? (Source: Using NBAR for Classification)
- A) Yes, but special MQC configuration commands are needed.
  - B) Yes, it can be enabled directly on the input tunnel or encrypted interface, and no special MQC commands needed.
  - C) Yes, it can be enabled, but with limited functionality.
  - D) Yes, but it works only on output WAN interfaces.
- Q4) What happens when the highest-priority queue becomes congested in a priority queuing algorithm? (Source: Introducing Queuing Implementations)
- A) All the other queues starve.
  - B) Tail dropping focuses on the highest-priority queue.
  - C) Other queues are served on a round-robin basis.
  - D) Packets in the highest-priority queue are moved to a lower-priority queue.
- Q5) How does WFQ implement tail dropping? (Source: Configuring FIFO and WFQ)
- A) drops the last packet to arrive
  - B) drops all nonvoice packets first
  - C) drops the lowest-priority packets first
  - D) drops packets from the most aggressive flows
- Q6) What additional queue is created when LLQ is used in CBWFQ? (Source: Configuring CBWFQ and LLQ)
- A) WFQ
  - B) FIFO queue
  - C) priority queue
  - D) round-robin queue

- Q7) A specific RED profile has been configured with a mark probability denominator of 1. What is the effect of this configuration on packet loss as the average queue length reaches the maximum threshold? (Source: Introducing Congestion Avoidance)
- A) Given this configuration, no packets are dropped until the average queue length is greater than the maximum threshold.
  - B) For every active traffic flow, one packet is discarded.
  - C) When the average queue length is at the maximum threshold, all packets are dropped.
  - D) This is an invalid configuration.
- Q8) Which is a major difference between traffic policing versus traffic shaping? (Source: Introducing Traffic Policing and Shaping)
- A) Traffic policing drops excess traffic, while traffic shaping delays excess traffic by queuing it.
  - B) Traffic policing is applied only in the outbound direction, while traffic shaping can be applied to both the inbound and outbound directions.
  - C) Traffic policing is not available on Cisco Catalyst switches such as the Cisco Catalyst 2950. Traffic shaping is available on Cisco Catalyst switches such as the Catalyst 2950.
  - D) Traffic policing requires policing queues to buffer excess traffic, while traffic shaping does not require any queues to buffer excess traffic.
- Q9) With Layer 2 payload compression, what can be done to improve the compression and decompression delay of the router? (Source: Understanding WAN Link Efficiency Mechanisms)
- A) enable CEF switching
  - B) enable fast switching
  - C) use the Stacker or Predictor compression algorithm
  - D) use hardware-assisted compression
- Q10) The QoS preclassify feature is designed to operate on \_\_\_\_\_. (Source: Implementing QoS Preclassify)
- A) logical interfaces
  - B) loopback interfaces
  - C) tunnel interfaces
  - D) physical interfaces
- Q11) Which three QoS tools are used to manage the delay, delay variation (jitter), bandwidth, and packet-loss parameters on a network? (Choose three.) (Source: Deploying End-to-End QoS)
- A) coding and decoding (codec) tools
  - B) flow-control (windowing) tools
  - C) scheduling tools
  - D) link efficiency tools
  - E) classification tools
  - F) routing tools
  - G) accounting tools

## Module Self-Check Answer Key

Q1)	B
Q2)	A
Q3)	B
Q4)	A
Q5)	D
Q6)	C
Q7)	C
Q8)	A
Q9)	D
Q10)	C
Q11)	C, D, E

