Cisco IOS Quality of Service Solutions Configuration Guide

Release 12.2SR

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

Cisco IOS Quality of Service Solutions Configuration Guide

About Cisco IOS Software Documentation

Last Updated: November 20, 2009

This document describes the objectives, audience, conventions, and organization used in Cisco IOS software documentation. Also included are resources for obtaining technical assistance, additional documentation, and other information from Cisco. This document is organized into the following sections:

Documentation Objectives

Cisco IOS documentation describes the tasks and commands available to configure and maintain Cisco networking devices.

Audience

The Cisco IOS documentation set is intended for users who configure and maintain Cisco networking devices (such as routers and switches) but who may not be familiar with the configuration and maintenance tasks, the relationship among tasks, or the Cisco IOS commands necessary to perform particular tasks. The Cisco IOS documentation set is also intended for those users experienced with Cisco IOS software who need to know about new features, new configuration options, and new software characteristics in the current Cisco IOS release.

Documentation Conventions

In Cisco IOS documentation, the term *router* may be used to refer to various Cisco products; for example, routers, access servers, and switches. These and other networking devices that support Cisco IOS software are shown interchangeably in examples and are used only for illustrative purposes. An example that shows one product does not necessarily mean that other products are not supported.

This section contains the following topics:

Typographic Conventions

Cisco IOS documentation uses the following typographic conventions:

Convention	Description
^ or Ctrl	Both the ^ symbol and Ctrl represent the Control (Ctrl) key on a keyboard. For example, the key combination **^D** or **Ctrl-D** means that you hold down the Control key while you press the D key. (Keys are indicated in capital letters but are not case sensitive.)
string	A string is a nonquoted set of characters shown in italics. For example, when setting a Simple Network Management Protocol (SNMP) community string to *public*, do not use quotation marks around the string; otherwise, the string will include the quotation marks.

Command Syntax Conventions

Cisco IOS documentation uses the following command syntax conventions:

Convention	Description
bold	Bold text indicates commands and keywords that you enter as shown.
italic	Italic text indicates arguments for which you supply values.
[x]	Square brackets enclose an optional keyword or argument.
...	An ellipsis (three consecutive nonbolded periods without spaces) after a syntax element indicates that the element can be repeated.
|	A vertical line, called a pipe, that is enclosed within braces or square brackets indicates a choice within a set of keywords or arguments.
[x | y]	Square brackets enclosing keywords or arguments separated by a pipe indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a pipe indicate a required choice.
[x {y | z}]	Braces and a pipe within square brackets indicate a required choice within an optional element.

Software Conventions

Cisco IOS software uses the following program code conventions:

Convention	Description
`Courier font`	Courier font is used for information that is displayed on a PC or terminal screen.
`Bold Courier font`	Bold Courier font indicates text that the user must enter.
`<  >`	Angle brackets enclose text that is not displayed, such as a password. Angle brackets also are used in contexts in which the italic font style is not supported; for example, ASCII text.
!	An exclamation point at the beginning of a line indicates that the text that follows is a comment, not a line of code. An exclamation point is also displayed by Cisco IOS software for certain processes.
[ ]	Square brackets enclose default responses to system prompts.

Reader Alert Conventions

Cisco IOS documentation uses the following conventions for reader alerts:

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

Timesaver Means *the described action saves time*. You can save time by performing the action described in the paragraph.

Documentation Organization

This section describes the Cisco IOS documentation set, how it is organized, and how to access it on Cisco.com. It also lists the configuration guides, command references, and supplementary references and resources that comprise the documentation set. It contains the following topics:

Cisco IOS Documentation Set

The Cisco IOS documentation set consists of the following:

- Release notes and caveats provide information about platform, technology, and feature support for a release and describe severity 1 (catastrophic), severity 2 (severe), and select severity 3 (moderate) defects in released Cisco IOS software. Review release notes before other documents to learn whether updates have been made to a feature.
- Sets of configuration guides and command references organized by technology and published for each standard Cisco IOS release.
  - Configuration guides—Compilations of documents that provide conceptual and task-oriented descriptions of Cisco IOS features.
  - Command references—Compilations of command pages in alphabetical order that provide detailed information about the commands used in the Cisco IOS features and the processes that comprise the related configuration guides. For each technology, there is a single command reference that supports all Cisco IOS releases and that is updated at each standard release.
- Lists of all the commands in a specific release and all commands that are new, modified, removed, or replaced in the release.
- Command reference book for **debug** commands. Command pages are listed in alphabetical order.
- Reference book for system messages for all Cisco IOS releases.

Cisco IOS Documentation on Cisco.com

The following sections describe the organization of the Cisco IOS documentation set and how to access various document types.

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

New Features List

The New Features List for each release provides a list of all features in the release with hyperlinks to the feature guides in which they are documented.

Feature Guides

Cisco IOS features are documented in feature guides. Feature guides describe one feature or a group of related features that are supported on many different software releases and platforms. Your Cisco IOS software release or platform may not support all the features documented in a feature guide. See the Feature Information table at the end of the feature guide for information about which features in that guide are supported in your software release.

Configuration Guides

Configuration guides are provided by technology and release and comprise a set of individual feature guides relevant to the release and technology.

Command References

Command reference books contain descriptions of Cisco IOS commands that are supported in many different software releases and on many different platforms. The books are organized by technology. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html.

Cisco IOS Supplementary Documents and Resources

Supplementary documents and resources are listed in Table 2 on page xi.

Configuration Guides, Command References, and Supplementary Resources

Table 1 lists, in alphabetical order, Cisco IOS software configuration guides and command references, including brief descriptions of the contents of the documents. The Cisco IOS command references contain commands for Cisco IOS software for all releases. The configuration guides and command references support many different software releases and platforms. Your Cisco IOS software release or platform may not support all these technologies.

Table 2 lists documents and resources that supplement the Cisco IOS software configuration guides and command references. These supplementary resources include release notes and caveats; master command lists; new, modified, removed, and replaced command lists; system messages; and the debug command reference.

For additional information about configuring and operating specific networking devices, and to access Cisco IOS documentation, go to the Product/Technologies Support area of Cisco.com at the following location:

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

Table 1 Cisco IOS Configuration Guides and Command References

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
• *Cisco IOS AppleTalk Configuration Guide* • *Cisco IOS AppleTalk Command Reference*	AppleTalk protocol.
• *Cisco IOS Asynchronous Transfer Mode Configuration Guide* • *Cisco IOS Asynchronous Transfer Mode Command Reference*	LAN ATM, multiprotocol over ATM (MPoA), and WAN ATM.

Table 1 Cisco IOS Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
• *Cisco IOS Bridging and IBM Networking Configuration Guide* • *Cisco IOS Bridging Command Reference* • *Cisco IOS IBM Networking Command Reference*	Transparent and source-route transparent (SRT) bridging, source-route bridging (SRB), Token Ring Inter-Switch Link (TRISL), and token ring route switch module (TRRSM). Data-link switching plus (DLSw+), serial tunnel (STUN), block serial tunnel (BSTUN); logical link control, type 2 (LLC2), synchronous data link control (SDLC); IBM Network Media Translation, including Synchronous Data Logical Link Control (SDLLC) and qualified LLC (QLLC); downstream physical unit (DSPU), Systems Network Architecture (SNA) service point, SNA frame relay access, advanced peer-to-peer networking (APPN), native client interface architecture (NCIA) client/server topologies, and IBM Channel Attach.
• *Cisco IOS Broadband Access Aggregation and DSL Configuration Guide* • *Cisco IOS Broadband Access Aggregation and DSL Command Reference*	PPP over ATM (PPPoA) and PPP over Ethernet (PPPoE).
• *Cisco IOS Carrier Ethernet Configuration Guide* • *Cisco IOS Carrier Ethernet Command Reference*	Operations, Administration, and Maintenance (OAM); Ethernet connectivity fault management (CFM); ITU-T Y.1731 fault management functions; Ethernet Local Management Interface (ELMI); MAC address support on service instances, bridge domains, and pseudowire; IEEE 802.3ad Link Bundling; Link Aggregation Control Protocol (LACP) support for Ethernet and Gigabit Ethernet links and EtherChannel bundles; LACP support for stateful switchover (SSO), in service software upgrade (ISSU), Cisco nonstop forwarding (NSF), and nonstop routing (NSR) on Gigabit EtherChannel bundles; and Link Layer Discovery Protocol (LLDP) and media endpoint discovery (MED).
• *Cisco IOS Configuration Fundamentals Configuration Guide* • *Cisco IOS Configuration Fundamentals Command Reference*	Autoinstall, Setup, Cisco IOS command-line interface (CLI), Cisco IOS file system (IFS), Cisco IOS web browser user interface (UI), basic file transfer services, and file management.
• *Cisco IOS DECnet Configuration Guide* • *Cisco IOS DECnet Command Reference*	DECnet protocol.
• *Cisco IOS Dial Technologies Configuration Guide* • *Cisco IOS Dial Technologies Command Reference*	Asynchronous communications, dial backup, dialer technology, dial-in terminal services and AppleTalk remote access (ARA), dial-on-demand routing, dial-out, ISDN, large scale dial-out, modem and resource pooling, Multilink PPP (MLP), PPP, and virtual private dialup network (VPDN).
• *Cisco IOS Flexible NetFlow Configuration Guide* • *Cisco IOS Flexible NetFlow Command Reference*	Flexible NetFlow.

Table 1 Cisco IOS Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
• *Cisco IOS High Availability Configuration Guide* • *Cisco IOS High Availability Command Reference*	A variety of high availability (HA) features and technologies that are available for different network segments (from enterprise access to service provider core) to facilitate creation of end-to-end highly available networks. Cisco IOS HA features and technologies can be categorized in three key areas: system-level resiliency, network-level resiliency, and embedded management for resiliency.
• *Cisco IOS Integrated Session Border Controller Command Reference*	A VoIP-enabled device that is deployed at the edge of networks. An SBC is a toolkit of functions, such as signaling interworking, network hiding, security, and quality of service (QoS).
• *Cisco IOS Intelligent Services Gateway Configuration Guide* • *Cisco IOS Intelligent Services Gateway Command Reference*	Subscriber identification, service and policy determination, session creation, session policy enforcement, session life-cycle management, accounting for access and service usage, and session state monitoring.
• *Cisco IOS Interface and Hardware Component Configuration Guide* • *Cisco IOS Interface and Hardware Component Command Reference*	LAN interfaces, logical interfaces, serial interfaces, virtual interfaces, and interface configuration.
• *Cisco IOS IP Addressing Services Configuration Guide* • *Cisco IOS IP Addressing Services Command Reference*	Address Resolution Protocol (ARP), Network Address Translation (NAT), Domain Name System (DNS), Dynamic Host Configuration Protocol (DHCP), and Next Hop Address Resolution Protocol (NHRP).
• *Cisco IOS IP Application Services Configuration Guide* • *Cisco IOS IP Application Services Command Reference*	Enhanced Object Tracking (EOT), Gateway Load Balancing Protocol (GLBP), Hot Standby Router Protocol (HSRP), IP Services, Server Load Balancing (SLB), Stream Control Transmission Protocol (SCTP), TCP, Web Cache Communication Protocol (WCCP), User Datagram Protocol (UDP), and Virtual Router Redundancy Protocol (VRRP).
• *Cisco IOS IP Mobility Configuration Guide* • *Cisco IOS IP Mobility Command Reference*	Mobile ad hoc networks (MANet) and Cisco mobile networks.
• *Cisco IOS IP Multicast Configuration Guide* • *Cisco IOS IP Multicast Command Reference*	Protocol Independent Multicast (PIM) sparse mode (PIM-SM), bidirectional PIM (bidir-PIM), Source Specific Multicast (SSM), Multicast Source Discovery Protocol (MSDP), Internet Group Management Protocol (IGMP), and Multicast VPN (MVPN).
• *Cisco IOS IP Routing: BFD Configuration Guide*	Bidirectional forwarding detection (BFD).
• *Cisco IOS IP Routing: BGP Configuration Guide* • *Cisco IOS IP Routing: BGP Command Reference*	Border Gateway Protocol (BGP), multiprotocol BGP, multiprotocol BGP extensions for IP multicast.
• *Cisco IOS IP Routing: EIGRP Configuration Guide* • *Cisco IOS IP Routing: EIGRP Command Reference*	Enhanced Interior Gateway Routing Protocol (EIGRP).
• *Cisco IOS IP Routing: ISIS Configuration Guide* • *Cisco IOS IP Routing: ISIS Command Reference*	Intermediate System-to-Intermediate System (IS-IS).

Table 1 Cisco IOS Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
• *Cisco IOS IP Routing: ODR Configuration Guide* • *Cisco IOS IP Routing: ODR Command Reference*	On-Demand Routing (ODR).
• *Cisco IOS IP Routing: OSPF Configuration Guide* • *Cisco IOS IP Routing: OSPF Command Reference*	Open Shortest Path First (OSPF).
• *Cisco IOS IP Routing: Protocol-Independent Configuration Guide* • *Cisco IOS IP Routing: Protocol-Independent Command Reference*	IP routing protocol-independent features and commands. Generic policy-based routing (PBR) features and commands are included.
• *Cisco IOS IP Routing: RIP Configuration Guide* • *Cisco IOS IP Routing: RIP Command Reference*	Routing Information Protocol (RIP).
• *Cisco IOS IP SLAs Configuration Guide* • *Cisco IOS IP SLAs Command Reference*	Cisco IOS IP Service Level Agreements (IP SLAs).
• *Cisco IOS IP Switching Configuration Guide* • *Cisco IOS IP Switching Command Reference*	Cisco Express Forwarding, fast switching, and Multicast Distributed Switching (MDS).
• *Cisco IOS IPv6 Configuration Guide* • *Cisco IOS IPv6 Command Reference*	For IPv6 features, protocols, and technologies, go to the IPv6 "Start Here" document.
• *Cisco IOS ISO CLNS Configuration Guide* • *Cisco IOS ISO CLNS Command Reference*	ISO Connectionless Network Service (CLNS).
• *Cisco IOS LAN Switching Configuration Guide* • *Cisco IOS LAN Switching Command Reference*	VLANs, Inter-Switch Link (ISL) encapsulation, IEEE 802.10 encapsulation, IEEE 802.1Q encapsulation, and multilayer switching (MLS).
• *Cisco IOS Mobile Wireless Gateway GPRS Support Node Configuration Guide* • *Cisco IOS Mobile Wireless Gateway GPRS Support Node Command Reference*	Cisco IOS Gateway GPRS Support Node (GGSN) in a 2.5-generation general packet radio service (GPRS) and 3-generation universal mobile telecommunication system (UMTS) network.
• *Cisco IOS Mobile Wireless Home Agent Configuration Guide* • *Cisco IOS Mobile Wireless Home Agent Command Reference*	Cisco Mobile Wireless Home Agent, an anchor point for mobile terminals for which mobile IP or proxy mobile IP services are provided.
• *Cisco IOS Mobile Wireless Packet Data Serving Node Configuration Guide* • *Cisco IOS Mobile Wireless Packet Data Serving Node Command Reference*	Cisco Packet Data Serving Node (PDSN), a wireless gateway that is between the mobile infrastructure and standard IP networks and that enables packet data services in a code division multiple access (CDMA) environment.
• *Cisco IOS Mobile Wireless Radio Access Networking Configuration Guide* • *Cisco IOS Mobile Wireless Radio Access Networking Command Reference*	Cisco IOS radio access network products.

Table 1 Cisco IOS Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
• *Cisco IOS Multiprotocol Label Switching Configuration Guide* • *Cisco IOS Multiprotocol Label Switching Command Reference*	MPLS Label Distribution Protocol (LDP), MPLSLayer 2 VPNs, MPLS Layer 3 VPNs, MPLS traffic engineering (TE), and MPLS Embedded Management (EM) and MIBs.
• *Cisco IOS Multi-Topology Routing Configuration Guide* • *Cisco IOS Multi-Topology Routing Command Reference*	Unicast and multicast topology configurations, traffic classification, routing protocol support, and network management support.
• *Cisco IOS NetFlow Configuration Guide* • *Cisco IOS NetFlow Command Reference*	Network traffic data analysis, aggregation caches, and export features.
• *Cisco IOS Network Management Configuration Guide* • *Cisco IOS Network Management Command Reference*	Basic system management; system monitoring and logging; troubleshooting, logging, and fault management; Cisco Discovery Protocol; Cisco IOS Scripting with Tool Control Language (Tcl); Cisco networking services (CNS); DistributedDirector; Embedded Event Manager (EEM); Embedded Resource Manager (ERM); Embedded Syslog Manager (ESM); HTTP; Remote Monitoring (RMON); SNMP; and VPN Device Manager Client for Cisco IOS software (XSM Configuration).
• *Cisco IOS Novell IPX Configuration Guide* • *Cisco IOS Novell IPX Command Reference*	Novell Internetwork Packet Exchange (IPX) protocol.
• *Cisco IOS Optimized Edge Routing Configuration Guide* • *Cisco IOS Optimized Edge Routing Command Reference*	Optimized edge routing (OER) monitoring; Performance Routing (PfR); and automatic route optimization and load distribution for multiple connections between networks.
• *Cisco IOS Quality of Service Solutions Configuration Guide* • *Cisco IOS Quality of Service Solutions Command Reference*	Traffic queueing, traffic policing, traffic shaping, Modular QoS CLI (MQC), Network-Based Application Recognition (NBAR), Multilink PPP (MLP) for QoS, header compression, AutoQoS, Resource Reservation Protocol (RSVP), and weighted random early detection (WRED).
• *Cisco IOS Security Command Reference*	Access control lists (ACLs); authentication, authorization, and accounting (AAA); firewalls; IP security and encryption; neighbor router authentication; network access security; network data encryption with router authentication; public key infrastructure (PKI); RADIUS; TACACS+; terminal access security; and traffic filters.
• *Cisco IOS Security Configuration Guide: Securing the Data Plane*	Access Control Lists (ACLs); Firewalls: Context-Based Access Control (CBAC) and Zone-Based Firewall; Cisco IOS Intrusion Prevention System (IPS); Flexible Packet Matching; Unicast Reverse Path Forwarding (uRPF); Threat Information Distribution Protocol (TIDP) and TMS.
• *Cisco IOS Security Configuration Guide: Securing the Control Plane*	Control Plane Policing, Neighborhood Router Authentication.

Table 1 Cisco IOS Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
• *Cisco IOS Security Configuration Guide: Securing User Services*	AAA (includes 802.1x authentication and Network Admission Control [NAC]); Security Server Protocols (RADIUS and TACACS+); Secure Shell (SSH); Secure Access for Networking Devices (includes Autosecure and Role-Based CLI access); Lawful Intercept.
• *Cisco IOS Security Configuration Guide: Secure Connectivity*	Internet Key Exchange (IKE) for IPsec VPNs; IPsec Data Plane features; IPsec Management features; Public Key Infrastructure (PKI); Dynamic Multipoint VPN (DMVPN); Easy VPN; Cisco Group Encrypted Transport VPN (GETVPN); SSL VPN.
• *Cisco IOS Service Advertisement Framework Configuration Guide* • *Cisco IOS Service Advertisement Framework Command Reference*	Cisco Service Advertisement Framework.
• *Cisco IOS Service Selection Gateway Configuration Guide* • *Cisco IOS Service Selection Gateway Command Reference*	Subscriber authentication, service access, and accounting.
• *Cisco IOS Software Activation Configuration Guide* • *Cisco IOS Software Activation Command Reference*	An orchestrated collection of processes and components to activate Cisco IOS software feature sets by obtaining and validating Cisco software licenses.
• *Cisco IOS Software Modularity Installation and Configuration Guide* • *Cisco IOS Software Modularity Command Reference*	Installation and basic configuration of software modularity images, including installations on single and dual route processors, installation rollbacks, software modularity binding, software modularity processes, and patches.
• *Cisco IOS Terminal Services Configuration Guide* • *Cisco IOS Terminal Services Command Reference*	DEC, local-area transport (LAT), and X.25 packet assembler/disassembler (PAD).
• *Cisco IOS Virtual Switch Command Reference*	Virtual switch redundancy, high availability, and packet handling; converting between standalone and virtual switch modes; virtual switch link (VSL); Virtual Switch Link Protocol (VSLP). **Note** For information about virtual switch configuration, see the product-specific software configuration information for the Cisco Catalyst 6500 series switch or for the Metro Ethernet 6500 series switch.
• *Cisco IOS Voice Configuration Library* • *Cisco IOS Voice Command Reference*	Cisco IOS support for voice call control protocols, interoperability, physical and virtual interface management, and troubleshooting. The library includes documentation for IP telephony applications.
• *Cisco IOS VPDN Configuration Guide* • *Cisco IOS VPDN Command Reference*	Layer 2 Tunneling Protocol (L2TP) dial-out load balancing and redundancy; L2TP extended failover; L2TP security VPDN; multihop by Dialed Number Identification Service (DNIS); timer and retry enhancements for L2TP and Layer 2 Forwarding (L2F); RADIUS Attribute 82 (tunnel assignment ID); shell-based authentication of VPDN users; tunnel authentication via RADIUS on tunnel terminator.

Table 1 Cisco IOS Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
• *Cisco IOS Wide-Area Networking Configuration Guide* • *Cisco IOS Wide-Area Networking Command Reference*	Frame Relay; Layer 2 Tunnel Protocol Version 3 (L2TPv3); L2VPN Pseudowire Redundancy; L2VPN Interworking; Layer2 Local Switching; Link Access Procedure, Balanced (LAPB); and X.25.
• *Cisco IOS Wireless LAN Configuration Guide* • *Cisco IOS Wireless LAN Command Reference*	Broadcast key rotation, IEEE 802.11x support, IEEE 802.1x authenticator, IEEE 802.1x local authentication service for Extensible Authentication Protocol-Flexible Authentication via Secure Tunneling (EAP-FAST), Multiple Basic Service Set ID (BSSID), Wi-Fi Multimedia (WMM) required elements, and Wi-Fi Protected Access (WPA).

Table 2 lists documents and resources that supplement the Cisco IOS software configuration guides and command references.

Table 2 Cisco IOS Supplementary Documents and Resources

Document Title or Resource	Description
Cisco IOS Master Command List, All Releases	Alphabetical list of all the commands documented in all Cisco IOS releases.
Cisco IOS New, Modified, Removed, and Replaced Commands	List of all the new, modified, removed, and replaced commands for a Cisco IOS release.
Cisco IOS System Message Guide	List of Cisco IOS system messages and descriptions. System messages may indicate problems with your system, may be informational only, or may help diagnose problems with communications lines, internal hardware, or system software.
Cisco IOS Debug Command Reference	Alphabetical list of **debug** commands including brief descriptions of use, command syntax, and usage guidelines.
Release Notes and Caveats	Information about new and changed features, system requirements, and other useful information about specific software releases; information about defects in specific Cisco IOS software releases.
MIBs	Files used for network monitoring. To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator.
RFCs	Standards documents maintained by the Internet Engineering Task Force (IETF) that Cisco IOS documentation references where applicable. The full text of referenced RFCs may be obtained at the following URL: http://www.rfc-editor.org/

Additional Resources and Documentation Feedback

What's New in Cisco Product Documentation is released monthly and describes all new and revised Cisco technical documentation. The *What's New in Cisco Product Documentation* publication also provides information about obtaining the following resources:

- Technical documentation
- Cisco product security overview
- Product alerts and field notices
- Technical assistance

Cisco IOS technical documentation includes embedded feedback forms where you can rate documents and provide suggestions for improvement. Your feedback helps us improve our documentation.

Using the Command-Line Interface in Cisco IOS Software

Last Updated: October 14, 2009

This document provides basic information about the command-line interface (CLI) in Cisco IOS software and how you can use some of the CLI features. This document contains the following sections:

For more information about using the CLI, see the "Using the Cisco IOS Command-Line Interface" section of the *Cisco IOS Configuration Fundamentals Configuration Guide*.

For information about the software documentation set, see the "About Cisco IOS Software Documentation" document.

Initially Configuring a Device

Initially configuring a device varies by platform. For information about performing an initial configuration, see the hardware installation documentation that is provided with the original packaging of the product or go to the Product/Technologies Support area of Cisco.com at http://www.cisco.com/go/techdocs.

After you have performed the initial configuration and connected the device to your network, you can configure the device by using the console port or a remote access method, such as Telnet or Secure Shell (SSH), to access the CLI or by using the configuration method provided on the device, such as Security Device Manager.

Changing the Default Settings for a Console or AUX Port

There are only two changes that you can make to a console port and an AUX port:

- Change the port speed with the **config-register 0x** command. Changing the port speed is not recommended. The well-known default speed is 9600.
- Change the behavior of the port; for example, by adding a password or changing the timeout value.

Note The AUX port on the Route Processor (RP) installed in a Cisco ASR 1000 series router does not serve any useful customer purpose and should be accessed only under the advisement of a customer support representative.

Using the CLI

This section describes the following topics:

Understanding Command Modes

The CLI command mode structure is hierarchical, and each mode supports a set of specific commands. This section describes the most common of the many modes that exist.

Table 1 lists common command modes with associated CLI prompts, access and exit methods, and a brief description of how each mode is used.

Table 1 CLI Command Modes

Command Mode	Access Method	Prompt	Exit Method	Mode Usage
User EXEC	Log in.	`Router>`	Issue the **logout** or **exit** command.	• Change terminal settings. • Perform basic tests. • Display device status.
Privileged EXEC	From user EXEC mode, issue the **enable** command.	`Router#`	Issue the **disable** command or the **exit** command to return to user EXEC mode.	• Issue **show** and **debug** commands. • Copy images to the device. • Reload the device. • Manage device configuration files. • Manage device file systems.
Global configuration	From privileged EXEC mode, issue the **configure terminal** command.	`Router(config)#`	Issue the **exit** command or the **end** command to return to privileged EXEC mode.	Configure the device.
Interface configuration	From global configuration mode, issue the **interface** command.	`Router(config-if)#`	Issue the **exit** command to return to global configuration mode or the **end** command to return to privileged EXEC mode.	Configure individual interfaces.
Line configuration	From global configuration mode, issue the **line vty** or **line console** command.	`Router(config-line)#`	Issue the **exit** command to return to global configuration mode or the **end** command to return to privileged EXEC mode.	Configure individual terminal lines.

Table 1 CLI Command Modes (continued)

Command Mode	Access Method	Prompt	Exit Method	Mode Usage
ROM monitor	From privileged EXEC mode, issue the **reload** command. Press the **Break** key during the first 60 seconds while the system is booting.	`rommon # >` The # symbol represents the line number and increments at each prompt.	Issue the **continue** command.	• Run as the default operating mode when a valid image cannot be loaded. • Access the fall-back procedure for loading an image when the device lacks a valid image and cannot be booted. • Perform password recovery when a Ctrl-Break sequence is issued within 60 seconds of a power-on or reload event.
Diagnostic (available only on Cisco ASR 1000 series routers)	The router boots or enters diagnostic mode in the following scenarios. When a Cisco IOS process or processes fail, in most scenarios the router will reload. • A user-configured access policy was configured using the **transport-map** command, which directed the user into diagnostic mode. • The router was accessed using an RP auxiliary port. • A break signal (**Ctrl-C**, **Ctrl-Shift-6**, or the **send break** command) was entered, and the router was configured to enter diagnostic mode when the break signal was received.	`Router(diag)#`	If a Cisco IOS process failure is the reason for entering diagnostic mode, the failure must be resolved and the router must be rebooted to exit diagnostic mode. If the router is in diagnostic mode because of a transport-map configuration, access the router through another port or use a method that is configured to connect to the Cisco IOS CLI. If the RP auxiliary port was used to access the router, use another port for access. Accessing the router through the auxiliary port is not useful for customer purposes.	• Inspect various states on the router, including the Cisco IOS state. • Replace or roll back the configuration. • Provide methods of restarting the Cisco IOS software or other processes. • Reboot hardware (such as the entire router, an RP, an ESP, a SIP, a SPA) or other hardware components. • Transfer files into or off of the router using remote access methods such as FTP, TFTP, and SCP.

EXEC commands are not saved when the software reboots. Commands that you issue in a configuration mode can be saved to the startup configuration. If you save the running configuration to the startup configuration, these commands will execute when the software is rebooted. Global configuration mode is the highest level of configuration mode. From global configuration mode, you can enter a variety of other configuration modes, including protocol-specific modes.

ROM monitor mode is a separate mode that is used when the software cannot load properly. If a valid software image is not found when the software boots or if the configuration file is corrupted at startup, the software might enter ROM monitor mode. Use the question symbol (?) to view the commands that you can use while the device is in ROM monitor mode.

```
rommon 1 > ?
alias               set and display aliases command
boot                boot up an external process
confreg             configuration register utility
cont                continue executing a downloaded image
context             display the context of a loaded image
cookie              display contents of cookie PROM in hex
.
.
.
rommon 2 >
```

The following example shows how the command prompt changes to indicate a different command mode:

```
Router> enable
Router# configure terminal
Router(config)# interface ethernet 1/1
Router(config-if)# ethernet
Router(config-line)# exit
Router(config)# end
Router#
```

Note A keyboard alternative to the **end** command is Ctrl-Z.

Using the Interactive Help Feature

The CLI includes an interactive Help feature. Table 2 describes the purpose of the CLI interactive Help commands.

Table 2 CLI Interactive Help Commands

Command	Purpose
help	Provides a brief description of the Help feature in any command mode.
?	Lists all commands available for a particular command mode.
*partial command***?**	Provides a list of commands that begin with the character string (no space between the command and the question mark).
partial command<**Tab**>	Completes a partial command name (no space between the command and <Tab>).
command **?**	Lists the keywords, arguments, or both associated with the command (space between the command and the question mark).
command keyword **?**	Lists the arguments that are associated with the keyword (space between the keyword and the question mark).

The following examples show how to use the help commands:

help

```
Router> help

Help may be requested at any point in a command by entering a question mark '?'. If nothing
matches, the help list will be empty and you must backup until entering a '?' shows the
available options.

Two styles of help are provided:

1. Full help is available when you are ready to enter a command argument (e.g. 'show ?')
and describes each possible argument.

2. Partial help is provided when an abbreviated argument is entered and you want to know
what arguments match the input (e.g. 'show pr?'.)
```

?

```
Router# ?
Exec commands:
  access-enable        Create a temporary access-List entry
  access-profile       Apply user-profile to interface
  access-template      Create a temporary access-List entry
  alps                 ALPS exec commands
  archive              manage archive files
<snip>
```

***partial command*?**

```
Router(config)# zo?
zone  zone-pair
```

***partial command*<Tab>**

```
Router(config)# we<Tab> webvpn
```

***command* ?**

```
Router(config-if)# pppoe ?
  enable        Enable pppoe
  max-sessions  Maximum PPPOE sessions
```

***command keyword* ?**

```
Router(config-if)# pppoe enable ?
  group  attach a BBA group
  <cr>
```

Understanding Command Syntax

Command syntax is the format in which a command should be entered in the CLI. Commands include the name of the command, keywords, and arguments. Keywords are alphanumeric strings that are used literally. Arguments are placeholders for values that a user must supply. Keywords and arguments may be required or optional.

Specific conventions convey information about syntax and command elements. Table 3 describes these conventions.

Table 3 CLI Syntax Conventions

Symbol/Text	Function	Notes
< > (angle brackets)	Indicate that the option is an argument.	Sometimes arguments are displayed without angle brackets.
A.B.C.D.	Indicates that you must enter a dotted decimal IP address.	Angle brackets (< >) are not always used to indicate that an IP address is an argument.
WORD (all capital letters)	Indicates that you must enter one word.	Angle brackets (< >) are not always used to indicate that a WORD is an argument.
LINE (all capital letters)	Indicates that you must enter more than one word.	Angle brackets (< >) are not always used to indicate that a LINE is an argument.
<cr> (carriage return)	Indicates the end of the list of available keywords and arguments, and also indicates when keywords and arguments are optional. When <cr> is the only option, you have reached the end of the branch or the end of the command if the command has only one branch.	—

The following examples show syntax conventions:

```
Router(config)# ethernet cfm domain ?
  WORD  domain name
Router(config)# ethernet cfm domain dname ?
  level
Router(config)# ethernet cfm domain dname level ?
  <0-7>  maintenance level number
Router(config)# ethernet cfm domain dname level 7 ?
  <cr>

Router(config)# snmp-server file-transfer access-group 10 ?
  protocol  protocol options
  <cr>

Router(config)# logging host ?
  Hostname or A.B.C.D  IP address of the syslog server
  ipv6                 Configure IPv6 syslog server
```

Understanding Enable and Enable Secret Passwords

Some privileged EXEC commands are used for actions that impact the system, and it is recommended that you set a password for these commands to prevent unauthorized use. Two types of passwords, enable (not encrypted) and enable secret (encrypted), can be set. The following commands set these passwords and are issued in global configuration mode:

- **enable** *password*
- **enable secret** *password*

Using an enable secret password is recommended because it is encrypted and more secure than the enable password. When you use an enable secret password, text is encrypted (unreadable) before it is written to the config.text file. When you use an enable password, the text is written as entered (readable) to the config.text file.

Each type of password is case sensitive, can contain from 1 to 25 uppercase and lowercase alphanumeric characters, and can start with a numeral. Spaces are also valid password characters; for example, "two words" is a valid password. Leading spaces are ignored, but trailing spaces are recognized.

Note Both password commands have numeric keywords that are single integer values. If you choose a numeral for the first character of your password followed by a space, the system will read the number as if it were the numeric keyword and not as part of your password.

When both passwords are set, the enable secret password takes precedence over the enable password.

To remove a password, use the **no** form of the commands: **no enable** *password* or **no enable secret** *password*.

For more information about password recovery procedures for Cisco products, see http://www.cisco.com/en/US/products/sw/iosswrel/ps1831/products_tech_note09186a00801746e6.shtml.

Using the Command History Feature

The command history feature saves, in a command history buffer, the commands that you enter during a session. The default number of saved commands is 10, but the number is configurable within the range of 0 to 256. This commandhistory feature is particularly useful for recalling long or complex commands.

To change the number of commands saved in the history buffer for a terminal session, issue the **terminal history size** command:

```
Router# terminal history size num
```

A command history buffer is also available in line configuration mode with the same default and configuration options. To set the command history buffer size for a terminal session in line configuration mode, issue the **history** command:

```
Router(config-line)# history [size num]
```

To recall commands from the history buffer, use the following methods:

- Press Ctrl-P or the Up Arrow key—Recalls commands beginning with the most recent command. Repeat the key sequence to recall successively older commands.
- Press Ctrl-N or the Down Arrow key—Recalls the most recent commands in the history buffer after they have been recalled using Ctrl-P or the Up Arrow key. Repeat the key sequence to recall successively more recent commands.

  Note The arrow keys function only on ANSI-compatible terminals such as the VT100.

- Issue the **show history** command in user EXEC or privileged EXEC mode—Lists the most recent commands that you entered. The number of commands that are displayed is determined by the setting of the **terminal history size** and **history** commands.

The command history feature is enabled by default. To disable this feature for a terminal session, issue the **terminal no history** command in user EXEC or privileged EXEC mode or the **no history** command in line configuration mode.

Abbreviating Commands

Typing a complete command name is not always required for the command to execute. The CLI recognizes an abbreviated command when the abbreviation contains enough characters to uniquely identify the command. For example, the **show version** command can be abbreviated as **sh ver**. It cannot be abbreviated as **s ver** because **s** could mean **show**, **set**, or **systat**. The **sh v** abbreviation also is not valid because the **show** command has **vrrp** as a keyword in addition to **version**. (Command and keyword examples are from Cisco IOS Release 12.4(13)T.)

Using Aliases for CLI Commands

To save time and the repetition of entering the same command multiple times, you can use a command alias. An alias can be configured to do anything that can be done at the command line, but an alias cannot move between modes, type in passwords, or perform any interactive functions.

Table 4 shows the default command aliases.

Table 4 Default Command Aliases

Command Alias	Original Command
h	help
lo	logout
p	ping
s	show
u or **un**	undebug
w	where

To create a command alias, issue the **alias** command in global configuration mode. The syntax of the command is **alias** *mode command-alias original-command*. Following are some examples:

- Router(config)# **alias exec prt partition**—privileged EXEC mode
- Router(config)# **alias configure sb source-bridge**—global configuration mode
- Router(config)# **alias interface rl rate-limit**—interface configuration mode

To view both default and user-created aliases, issue the **show alias** command.

For more information about the **alias** command, see http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_a1.html.

Using the no and default Forms of Commands

Most configuration commands have a **no** form that is used to reset a command to its default value or disable a feature or function. For example, the **ip routing** command is enabled by default. To disable this command, you would issue the **no ip routing** command. To re-enable IP routing, you would issue the **ip routing** command.

Configuration commands may also have a **default** form, which returns the command settings to their default values. For commands that are disabled by default, using the **default** form has the same effect as using the **no** form of the command. For commands that are enabled by default and have default settings, the **default** form enables the command and returns the settings to their default values.

The **no** form is documented in the command pages of command references. The **default** form is generally documented in the command pages only when the **default** form performs a different function than the plain and **no** forms of the command. To see what **default** commands are available on your system, enter **default ?** in the appropriate command mode.

Using the debug Command

A **debug** command produces extensive output that helps you troubleshoot problems in your network. These commands are available for many features and functions within Cisco IOS software. Some **debug** commands are **debug all**, **debug aaa accounting**, and **debug mpls packets**. To use **debug** commands during a Telnet session with a device, you must first enter the **terminal monitor** command. To turn off debugging completely, you must enter the **undebug all** command.

For more information about **debug** commands, see the *Cisco IOS Debug Command Reference* at http://www.cisco.com/en/US/docs/ios/debug/command/reference/db_book.html.

Caution Debugging is a high priority and high CPU utilization process that can render your device unusable. Use **debug** commands only to troubleshoot specific problems. The best times to run debugging are during periods of low network traffic and when few users are interacting with the network. Debugging during these periods decreases the likelihood that the **debug** command processing overhead will affect network performance or user access or response times.

Filtering Output Using Output Modifiers

Many commands produce lengthy output that may use several screens to display. Using output modifiers, you can filter this output to show only the information that you want to see.

The following three output modifiers are available:

- **begin** *regular-expression*—Displays the first line in which a match of the regular expression is found and all lines that follow.
- **include** *regular-expression*—Displays all lines in which a match of the regular expression is found.
- **exclude** *regular-expression*—Displays all lines except those in which a match of the regular expression is found.

To use one of these output modifiers, type the command followed by the pipe symbol (|), the modifier, and the regular expression that you want to search for or filter. A regular expression is a case-sensitive alphanumeric pattern. It can be a single character or number, a phrase, or a more complex string.

The following example illustrates how to filter output of the **show interface** command to display only lines that include the expression "protocol."

```
Router# show interface | include protocol

FastEthernet0/0 is up, line protocol is up
Serial4/0 is up, line protocol is up
Serial4/1 is up, line protocol is up
Serial4/2 is administratively down, line protocol is down
Serial4/3 is administratively down, line protocol is down
```

Understanding CLI Error Messages

You may encounter some error messages while using the CLI. Table 5 shows the common CLI error messages.

Table 5 Common CLI Error Messages

Error Message	Meaning	How to Get Help
% Ambiguous command: "show con"	You did not enter enough characters for the command to be recognized.	Reenter the command followed by a space and a question mark (?). The keywords that you are allowed to enter for the command appear.
% Incomplete command.	You did not enter all the keywords or values required by the command.	Reenter the command followed by a space and a question mark (?). The keywords that you are allowed to enter for the command appear.
% Invalid input detected at "^" marker.	You entered the command incorrectly. The caret (^) marks the point of the error.	Enter a question mark (?) to display all the commands that are available in this command mode. The keywords that you are allowed to enter for the command appear.

For more system error messages, see the following document:

- *Cisco IOS Release 12.4T System Message Guide*

Saving Changes to a Configuration

To save changes that you made to the configuration of a device, you must issue the **copy running-config startup-config** command or the **copy system:running-config nvram:startup-config** command. When you issue these commands, the configuration changes that you made are saved to the startup configuration and saved when the software reloads or power to the device is turned off or interrupted. The following example shows the syntax of the **copy running-config startup-config** command:

```
Router# copy running-config startup-config
Destination filename [startup-config]?
```

You press Enter to accept the startup-config filename (the default), or type a new filename and then press Enter to accept that name. The following output is displayed indicating that the configuration was saved.

```
Building configuration...
[OK]
Router#
```

On most platforms, the configuration is saved to NVRAM. On platforms with a Class A flash file system, the configuration is saved to the location specified by the CONFIG_FILE environment variable. The CONFIG_FILE variable defaults to NVRAM.

Additional Information

- "Using the Cisco IOS Command-Line Interface" section of the *Cisco IOS Configuration Fundamentals Configuration Guide*

  http://www.cisco.com/en/US/docs/ios/fundamentals/configuration/guide/cf_cli-basics.html
- Cisco Product/Technology Support

  http://www.cisco.com/go/techdocs
- Support area on Cisco.com (also search for documentation by task or product)

  http://www.cisco.com/en/US/support/index.html
- Software Download Center (downloads; tools; licensing, registration, advisory, and general information) (requires Cisco.com user ID and password)

  http://www.cisco.com/kobayashi/sw-center/
- Error Message Decoder, a tool to help you research and resolve error messages for Cisco IOS software

  http://www.cisco.com/pcgi-bin/Support/Errordecoder/index.cgi
- Command Lookup Tool, a tool to help you find detailed descriptions of Cisco IOS commands (requires Cisco.com user ID and password)

  http://tools.cisco.com/Support/CLILookup
- Output Interpreter, a troubleshooting tool that analyzes command output of supported **show** commands

  https://www.cisco.com/pcgi-bin/Support/OutputInterpreter/home.pl

---

Quality of Service Overview

This chapter explains quality of service (QoS) and the service models that embody it. It also suggests benefits that you can gain from implementing Cisco IOS QoS in your network. Then it focuses on the Cisco IOS QoS features and the technologies that implement them.

This chapter contains the following sections:

What Is Quality of Service?

QoS refers to the ability of a network to provide improved service to selected network traffic over various underlying technologies including Frame Relay, ATM, Ethernet and 802.1 networks, SONET, and IP-routed networks. In particular, QoS features provide improved and more predictable network service by implementing the following services:

- Supporting guaranteed bandwidth
- Improving loss characteristics
- Avoiding and managing network congestion
- Shaping network traffic
- Setting traffic priorities across the network

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

About QoS Architecture

You configure QoS features throughout a network to provide for end-to-end QoS delivery. The following three components are necessary to deliver QoS across a heterogeneous network:

- QoS within a single network element, which includes queuing, scheduling, and traffic shaping features.
- QoS signaling techniques for coordinating QoS for end-to-end delivery between network elements.
- QoS policing and management functions to control and administer end-to-end traffic across a network.

Not all QoS techniques are appropriate for all network routers. Because edge routers and backbone routers in a network do not necessarily perform the same operations, the QoS tasks that they perform might differ as well. To configure an IP network for real-time voice traffic, for example, you would need to consider the functions of both edge and backbone routers in the network and then select the appropriate QoS feature or features.

In general, edge routers perform the following QoS functions:

- Packet classification and marking
- Admission control
- Configuration management

In general, backbone routers perform the following QoS functions:

- Congestion management
- Congestion avoidance

Who Could Benefit from Using Cisco IOS QoS?

All networks can take advantage of aspects of QoS for optimum efficiency, whether the network is for a small corporation, an enterprise, or an Internet service provider (ISP). Different categories of networking users—such as major enterprises, network service providers, and small- and medium-sized businesses—have their own QoS requirements; in many areas, however, these requirements overlap. The Cisco IOS QoS features described in the "Cisco IOS QoS Features" section on page 5 address these diverse and common needs.

Enterprise networks, for example, must provide end-to-end QoS solutions across the various platforms that comprise the network. Providing solutions for heterogeneous platforms often requires that you take a different QoS configuration approach for each technology. As enterprise networks carry more complex, mission-critical applications and experience increased traffic from web multimedia applications, QoS serves to prioritize this traffic to ensure that each application gets the service that it requires.

ISPs require assured scalability and performance. For example, ISPs that have long offered best-effort IP connectivity now also transfer voice, video, and other real-time critical application data. QoS answers the scalability and performance needs of these ISPs to distinguish different kinds of traffic, thereby enabling them to offer service differentiation to their customers.

In the small- and medium-sized business segment, managers are experiencing firsthand the rapid growth of business on the Internet. These business networks must also handle increasingly complex business applications. QoS lets the network handle the difficult task of utilizing an expensive WAN connection in the most efficient way for business applications.

Why Deploy Cisco IOS QoS?

The Cisco IOS QoS features enable networks to control and predictably service a variety of networked applications and traffic types. Implementing Cisco IOS QoS in your network has the following advantages:

- Control over resources. You have control over which resources (bandwidth, equipment, wide-area facilities, and so on) are being used. For example, you can limit bandwidth consumed over a backbone link by FTP transfers or give priority to an important database access.
- Tailored services. If you are an ISP, the control and visibility provided by QoS enables you to offer carefully tailored grades of service differentiation to your customers.
- Coexistence of mission-critical applications. Cisco IOS QoS features ensures following conditions:
  - That your WAN is used efficiently by mission-critical applications that are most important to your business.
  - That bandwidth and minimum delays required by time-sensitive multimedia and voice applications are available.
  - That other applications using the link get their fair service without interfering with mission-critical traffic.

Moreover, in implementing QoS features in your network, you put in place the foundation for a future fully integrated network.

End-to-End QoS Models

A service model, also called a level of service, describes a set of end-to-end QoS capabilities. End-to-end QoS is the ability of the network to deliver service required by specific network traffic from one end of the network to another. Cisco IOS QoS software supports three types of service models: best effort, integrated, and differentiated services.

Note QoS service models differ in how they enable applications to send data and in the ways in which the network attempts to deliver that data. For instance, one service model can be used for real-time applications, such as audio and video conferencing and IP telephony, while another service model can be used for file transfer and e-mail applications.

Consider the following factors when deciding which type of service to deploy in the network:

- The application or problem that you are trying to solve. Each of the three types of service—best effort, integrated, and differentiated—is appropriate for certain applications.
- The kind of capability that you want to allocate to your resources.
- Cost-benefit analysis. For example, the cost of implementing and deploying differentiated service is certain to be more expensive than the cost for a best-effort service.

The following sections describe the service models that are supported by features in Cisco IOS software:

- Best-Effort Service
- Integrated Service
- Differentiated Service

Best-Effort Service

Best effort is a single service model in which an application sends data whenever it must, in any quantity, and without requesting permission or first informing the network. For best-effort service, the network delivers data if it can, without any assurance of reliability, delay bounds, or throughput.

The Cisco IOS QoS feature that implements best-effort service is first-in, first-out (FIFO) queuing. Best-effort service is suitable for a wide range of networked applications such as general file transfers or e-mail.

Integrated Service

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

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

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

- The Resource Reservation Protocol (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 the following kinds of services:
  - Guaranteed rate service, which allows applications to reserve bandwidth to meet their requirements. For example, a Voice over IP (VoIP) application can reserve the required amount of bandwidth end-to-end using this kind of service. Cisco IOS QoS uses weighted fair queuing (WFQ) with RSVP to provide this kind of service.
  - Controlled load service, which allows applications to have low delay and high throughput even during times of congestion. For example, adaptive real-time applications, such as playback of a recorded conference, can use this kind of service. Cisco IOS QoS uses RSVP with Weighted Random Early Detection (WRED) to provide this kind of service.

Differentiated Service

Differentiated service is a multiple service model that can satisfy differing QoS requirements. However, unlike in the integrated service model, an application using differentiated service does not explicitly signal the router before sending data.

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

The differentiated service model is used for several mission-critical applications and for providing end-to-end QoS. Typically, this service model is appropriate for aggregate flows because it performs a relatively coarse level of traffic classification.

Cisco IOS QoS includes the following features that support the differentiated service model:

- Committed access rate (CAR), which performs metering and policing of traffic, providing bandwidth management.
- Intelligent queuing schemes such as WRED and WFQ and their equivalent features on the Versatile Interface Processor (VIP), which are distributed WRED (DWRED) and distributed WFQ. These features can be used with CAR to deliver differentiated services.

For more information on how to implement differentiated services using the components of Cisco IOS software, see the "Overview of DiffServ for Quality of Service" chapter.

Cisco IOS QoS Features

The Cisco IOS QoS software provides the major features described in the following sections. Some of these features have been previously mentioned, and all of them are briefly introduced in this chapter.

- Classification
- Congestion Management
- Congestion Avoidance
- Policing and Shaping
- Signaling
- Link Efficiency Mechanisms
- QoS Solutions
- Modular QoS Command-Line Interface
- Security Device Manager
- AutoQoS

The features listed are described more fully in the overview chapters of this book, which is organized into parts, one for each of the major features listed. Each book part contains an overview chapter and one or more configuration chapters.

Classification

Classifying network traffic allows you to organize traffic (that is, packets) into traffic classes or categories on the basis of whether the traffic matches specific criteria. Classifying network traffic (used in conjunction with marking network traffic) is the foundation for enabling many QoS features on your network.

For more conceptual information about classification, see the "Classification Overview" chapter.

For more information about classifying network traffic, see the "Classifying Network Traffic" chapter.

For more information about classifying network traffic using Network-Based Application Recognition (NBAR), see the "Classifying Network Traffic Using NBAR" chapter.

For more information about marking network traffic, see the "Marking Network Traffic" chapter.

Congestion Management

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

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

- FIFO queuing
- Priority queuing (PQ)
- Frame Relay permanent virtual circuit (PVC) interface priority queuing (FR PIPQ)
- Custom queuing (CQ)
- Weighted fair queuing (WFQ) and distributed WFQ (DWFQ)
- Class-based WFQ (CBWFQ) and Distributed CBWFQ (DCBWFQ)
- IP RTP Priority
- Frame Relay IP RTP Priority
- Low Latency Queuing (LLQ)
- Distributed LLQ (DLLQ)
- LLQ for Frame Relay

For more complete conceptual information on congestion management, see the "Congestion Management Overview" chapter.

For information on how to configure the various protocols that implement congestion management, see the following chapters:

- "Configuring Weighted Fair Queueing"
- "Configuring Custom Queueing"
- "Configuring Priority Queueing"

For complete command syntax information, see the *Cisco IOS Quality of Service Solutions Command Reference.*

What Is Congestion in Networks?

To give you a more definite sense of congestion in networks, this section briefly describes some of its characteristics, drawing on the explanation presented by V. Paxson and S. Floyd in a paper titled *Wide Area Traffic: The Failure of Poisson Modeling*.

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

FIFO Queuing

FIFO provides basic store-and-forward capability. FIFO is the default queuing algorithm in some instances, thus requiring no configuration. See the "FIFO Queuing" section on page 7 for a complete explanation of default configuration.

PQ

Designed to give strict priority to important traffic, PQ ensures that important traffic gets the fastest handling at each point where PQ is used. PQ can flexibly prioritize according to network protocol (such as IP, IPX, or AppleTalk), incoming interface, packet size, source/destination address, and so on.

FR PIPQ

FR PIPQ provides an interface-level PQ scheme in which prioritization is based on destination PVC rather than on packet contents. For example, FR PIPQ allows you to configure PVC transporting voices traffic to have absolute priority over a PVC transporting signaling traffic and a PVC transporting signaling traffic to have absolute priority over a PVC transporting data.

FR PIPQ provides four levels of priority: high, medium, normal, and low. The Frame Relay packet is examined at the interface for the data-link connection identifier (DLCI) value. The packet is then sent to the correct priority queue on the basis of the priority level configured for that DLCI.

CQ

CQ reserves a percentage of the available bandwidth of an interface for each selected traffic type. If a particular type of traffic is not using the bandwidth reserved for it, then other traffic types may use the remaining reserved bandwidth.

WFQ and DWFQ

WFQ applies priority (or weights) to identified traffic to classify traffic into conversations and determine how much bandwidth each conversation is allowed relative to other conversations. WFQ classifies traffic into different flows on the basis of such characteristics as source and destination address, protocol, and port and socket of the session.

To provide large-scale support for applications and traffic classes that require bandwidth allocations and delay bounds over the network infrastructure, Cisco IOS QoS includes a version of WFQ that runs only in distributed mode on VIPs. This version is called distributed WFQ (DWFQ). It provides increased flexibility in terms of traffic classification, weight assessment, and discard policy, and delivers Internet-scale performance on the Cisco 7500 series platforms.

For serial interfaces at E1 (2.048 Mbps) and below, WFQ is used by default. When no other queuing strategies are configured, all other interfaces use FIFO by default.

CBWFQ and DCBWFQ

The CBWFQ and DCBWFQ features extend the standard WFQ functionality to provide support for user-defined traffic classes. They allow you to specify the exact amount of bandwidth to be allocated for a specific class of traffic. Taking into account available bandwidth on the interface, you can configure up to 64 classes and control distribution among them.

DCBWFQ is intended for use on the VIP-based Cisco 7000 series routers with the Route Switch Processors (RSPs) and on the Cisco 7500 series routers.

IP RTP Priority

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

Frame Relay IP RTP Priority

The Frame Relay IP RTP Priority feature provides a strict PQ scheme on a Frame Relay PVC for delay-sensitive data such as voice. Voice traffic can be identified by its Real-Time Transport Protocol (RTP) port numbers and can be classified into a priority queue configured by the **frame-relay ip rtp priority** command. With this feature, voice traffic receives preferential treatment over nonvoice traffic.

LLQ

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

Additionally, the functionality of LLQ has been extended to allow you to specify the committed burst (Bc) size in LLQ and to change (or vary) the number of packets contained in the hold queue per-VC (on ATM adapters that support per-VC queuing). For more information, see the "Congestion Management Overview" chapter.

DLLQ

The DLLQ feature provides the ability to specify low-latency behavior for a traffic class on a VIP-based Cisco 7500 series router. DLLQ allows delay-sensitive data such as voice to be dequeued and sent before packets in other queues are dequeued.

The DLLQ feature also introduces the ability to limit the depth of a device transmission ring.

LLQ for Frame Relay

LLQ for Frame Relay provides strict PQ for voice traffic and WFQs for other classes of traffic. Before the release of this feature, LLQ was available at the interface and ATM VC levels. It is now available at the Frame Relay VC level when Frame Relay Traffic Shaping is configured.

Strict PQ improves QoS by allowing delay-sensitive traffic such as voice to be pulled from the queue and sent before other classes of traffic.

LLQ for Frame Relay allows you to define classes of traffic according to protocol, interface, or access lists. You can then assign characteristics to those classes, including priority, bandwidth, queue limit, and WRED.

Congestion Avoidance

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) class traffic under congestion situations while concurrently maximizing network throughput and capacity utilization and minimizing packet loss and delay. WRED and DWRED are the Cisco IOS QoS congestion avoidance features.

Router behavior allows output buffers to fill during periods of congestion, using the tail drop feature to resolve the problem when WRED is not configured. During tail drop, a potentially large number of packets from numerous connections are discarded because of lack of buffer capacity. This behavior can result in waves of congestion followed by periods during which the transmission link is not fully used. WRED obviates this situation proactively by providing congestion avoidance. That is, instead of waiting for buffers to fill before dropping packets, the router monitors the buffer depth and performs early discards on selected packets sent over selected connections.

WRED is the Cisco implementation of the RED class of congestion avoidance algorithms. When RED is used and the source detects the dropped packet, the source slows its transmission. RED is primarily designed to work with TCP in IP internetwork environments.

WRED can also be configured to use the DSCP value when it calculates the drop probability of a packet, enabling WRED to be compliant with the DiffServ standard being developed by the Internet Engineering Task Force (IETF).

For more complete conceptual information, see the "Congestion Avoidance Overview" chapter.

For information on how to configure WRED, DWRED, flow-based WRED, and DiffServ compliant WRED, see the "Configuring Weighted Random Early Detection" chapter.

For complete command syntax information, see the *Cisco IOS Quality of Service Solutions Command Reference.*

WRED

WRED, the Cisco implementation of RED, combines the capabilities of the RED algorithm with IP Precedence to provide preferential traffic handling for higher priority packets. It can selectively discard lower priority traffic when the interface begins to get congested and provide differentiated performance characteristics for different classes of service. WRED is also RSVP-aware. WRED is available on the Cisco 7200 series Route Switch Processor (RSP).

DWRED

DWRED is the Cisco high-speed version of WRED. The DWRED algorithm was designed with ISP providers in mind; it allows an ISP to define minimum and maximum queue depth thresholds and drop capabilities for each class of service. DWRED, which is available on the Cisco 7500 series routers or the Cisco 7000 series router with RSPs, is analogous in function to WRED, which is available on the Cisco 7200 series RSP.

Flow-Based WRED

The Flow-Based WRED feature forces WRED to afford greater fairness to all flows on an interface in regard to how packets are dropped. To provide fairness to all flows, the Flow-Based WRED feature has the following functionality:

- It ensures that flows that respond to WRED packet drops by backing off packet transmission are protected from flows that do not respond to WRED packet drops.
- It prohibits a single flow from monopolizing the buffer resources at an interface.

DiffServ Compliant WRED

The DiffServ Compliant WRED feature extends the functionality of WRED to enable support for Differentiated Services (DiffServ) and Assured Forwarding (AF) Per Hop Behavior (PHB). This feature enables customers to implement AF PHB by coloring packets according to DSCP values and then assigning preferential drop probabilities to those packets.

The DiffServ and the AF PHB standards are supported by this feature.

Policing and Shaping

For traffic policing, Cisco IOS QoS includes traffic policing capabilities implemented through the rate-limiting aspects of CAR and the Traffic Policing feature.

For traffic shaping, Cisco IOS QoS includes Generic Traffic Shaping (GTS), Class-Based Shaping, and Frame Relay Traffic Shaping (FRTS). These features allow you to regulate packet flow (that is, the flow of traffic) on your network.

For more complete conceptual information about traffic policing and traffic shaping, see the "Policing and Shaping Overview" chapter.

Signaling

Cisco IOS QoS signaling provides a way for an end station or network node to signal its neighbors to request special handling of certain traffic. QoS signaling is useful for coordinating the traffic-handling techniques provided by other QoS features. It plays a key role in configuring successful overall end-to-end QoS service across your network.

Cisco IOS QoS signaling takes advantage of IP. Either in-band (IP Precedence, 802.1p) or out-of-band (RSVP) signaling is used to indicate that a particular QoS service is desired for a particular traffic classification. Together, IP Precedence and RSVP provide a robust combination for end-to-end QoS signaling: IP Precedence signals for differentiated QoS, and RSVP signals for guaranteed QoS.

Cisco IOS software offers the following features and functionality associated with signaling:

- ATM User Network Interface (UNI) signaling and Frame Relay Local Management Interface (LMI)

  Achieves the end-to-end benefits of IP Precedence and RSVP signaling, and provides signaling into their respective backbone technologies.
- Common Open Policy Service (COPS) with RSVP

  Achieves centralized monitoring and control of RSVP signaling.
- Subnetwork Bandwidth Manager (SBM)

  Enables admission control over IEEE 802-styled networks.
- RSVP-ATM QoS Interworking feature

  Provides support for Controlled Load Service using RSVP over an ATM core network.
- RSVP support for Low Latency Queuing (LLQ) and Frame Relay.

For more complete conceptual information, see the "Signalling Overview" chapter.

For information on how to configure the various protocols that implement signaling, see the following chapters:

- "Configuring RSVP"
- "Configuring RSVP Support for LLQ"
- "Configuring RSVP Support for Frame Relay"
- "Configuring COPS for RSVP"
- "Configuring Subnetwork Bandwidth Manager"
- "Configuring RSVP-ATM QoS Interworking"

For complete command syntax information, see the *Cisco IOS Quality of Service Solutions Command Reference.*

Link Efficiency Mechanisms

Cisco IOS software offers a number of link-layer efficiency mechanisms or features designed to reduce latency and jitter for network traffic. These link efficiency mechanisms include the following:

- Multilink PPP (MLP)
- Frame Relay Fragmentation
- Header Compression

These mechanisms work with queuing and fragmentation to improve the efficiency and predictability of the application service levels.

For more complete conceptual information, see the "Link Efficiency Mechanisms Overview" chapter.

Multilink PPP

At the highest level, MLP provides packet interleaving, packet fragmentation, and packet resequencing across multiple logical data links. The packet interleaving, packet fragmentation, and packet resequencing are used to accommodate the fast transmission times required for sending real-time packets (for example, voice packets) across the network links. MLP is especially useful over slow network links (that is, a network link with a link speed less than or equal to 768 kbps).

For more conceptual information about MLP, see "Reducing Latency and Jitter for Real-Time Traffic Using Multilink PPP" module.

Frame Relay Fragmentation

Cisco has developed the following a number of methods of performing Frame Relay fragmentation, including the following:

- End-to-end FRF.12 (and higher) fragmentation
- Frame Relay fragmentation using FRF.11 Annex C (and higher)
- Cisco proprietary encapsulation

For more information about Frame Relay fragmentation, see the "Frame Relay Queueing and Fragmentation at the Interface" module.

Header Compression

Header compression is a mechanism that compresses the IP header in a packet before the packet is transmitted. Header compression reduces network overhead and speeds up the transmission of Real-Time Transport Protocol (RTP) and Transmission Control Protocol (TCP) packets. Header compression also reduces the amount of bandwidth consumed when the RTP or TCP packets are transmitted.

For more information about header compression, see the "Link Efficiency Mechanisms Overview" chapter.

QoS Solutions

The Cisco IOS QoS software includes a number of features collectively referred to as "QoS solutions." These software features include the following:

- IP to ATM CoS
- QoS features for voice
- Differentiated services implementations
- QoS: Classification, Policing, and Marking on a Layer 2 Tunneling Protocol (L2TP) Access Concentrator (LAC)
- QoS Bandwidth Estimation

IP to ATM CoS

IP to ATM CoS is a feature suite that maps QoS characteristics between IP and ATM, making it possible to support differentiated services in network service provider environments.

Network managers can use existing features such as CAR or PBR to classify and mark different IP traffic by modifying the IP Precedence field in the IPv4 packet header. Subsequently, WRED or DWRED can be configured on a per-VC basis so that the IP traffic is subject to different drop probabilities (and therefore priorities) as IP traffic coming into a router competes for bandwidth on a particular VC.

IP to ATM CoS provides support for ATM VC bundle management, allowing you to configure multiple VCs that have different QoS characteristics between any pair of ATM-connected routers.

IP to ATM CoS also provides for per-VC WFQ and CBWFQ, which allows you to apply CBWFQ functionality—normally applicable at the interface or subinterface levels only—to an individual VC configured for IP to ATM CoS. You can use this feature to apply either CBWFQ or flow-based WFQ on a per-VC basis.

For more complete conceptual information, see the "IP to ATM Class of Service Overview" chapter.

For information on how to configure IP to ATM CoS, see the "Configuring IP to ATM Class of Service" chapter.

QoS Features for Voice

Many of the QoS features already mentioned in this chapter are useful for voice applications. For a high-level overview of Cisco IOS QoS features for voice, see the "Introduction to QoS Features for Voice" chapter.

Differentiated Services Implementations

Many of the QoS features can be used to implement Differentiated Services on your network. For a high-level overview of how to use the Cisco IOS components to implement Differentiated Services, see the "Overview of DiffServ for Quality of Service" chapter.

QoS: Classification, Policing, and Marking on a LAC

The QoS: Classification, Policing, and Marking on a Layer 2 Tunneling Protocol (L2TP) Access Concentrator (LAC) feature allows service providers to classify packets based upon the IP type of service (ToS) bits in an embedded IP packet. The classification will be used to police the incoming traffic according to the differentiated services code point (DSCP) value. The purpose of classifying the packet by examining its encapsulation is to simplify the implementation and configuration needed for a large number of Point-to-Point Protocol (PPP) sessions.

For more information about the this feature, see the "QoS: Classification, Policing, and Marking on a LAC" module.

QoS Bandwidth Estimation

The QoS Bandwidth Estimation feature uses Corvil Bandwidth technology to allow you as a network manager to determine the bandwidth requirements to achieve user-specified QoS targets for networked applications.

For more information about the QoS Bandwidth Estimation feature, see the QoS Bandwidth Estimation module.

Modular QoS Command-Line Interface

The Modular Quality of Service Command-Line Interface (MQC) is a CLI structure that allows you to define a traffic class, create a traffic policy (policy map), and attach the traffic policy to an interface. The traffic policy contains the QoS feature that will be applied to the traffic class.

The MQC structure consists of the following three high-level steps:

Step 1 Define a traffic class by using the **class-map** command. A traffic class is used to classify traffic.

Step 2 Create a traffic policy by using the **policy-map** command. (The terms *traffic policy* and *policy map* are often synonymous). A traffic policy (policy map) contains a traffic class and one or more QoS features that will be applied to the traffic class. The QoS features in the traffic policy determine how to treat the classified traffic.

Step 3 Attach the traffic policy (policy map) to the interface by using the **service-policy** command.

For concepts and tasks associated with the MQC, see the "Applying QoS Features Using the MQC" module.

Security Device Manager

The Cisco Router and Security Device Manager (SDM) provides an intuitive, graphical user interface for configuring and monitoring advanced IP-based QoS functionality within Cisco routers.

For a high-level overview of SDM, see the "Security Device Manager Overview" chapter.

AutoQoS

The AutoQoS feature allows you to automate the delivery of QoS on your network and provides a means for simplifying the implementation and provisioning of QoS.

For more information about AutoQoS, see the "AutoQoS — VoIP" module or the "AutoQoS for the Enterprise" module.

Classification

Classification Overview

Classifying network traffic allows you to organize traffic (that is, packets) into traffic classes or categories on the basis of whether the traffic matches specific criteria. Classifying network traffic (used in conjunction with marking network traffic) is the foundation for enabling many quality of service (QoS) features on your network.

Packet classification is pivotal to policy techniques that select packets traversing a network element or a particular interface for different types of QoS service. For example, you can use classification to mark certain packets for IP Precedence, and you can identify other packets as belonging to a Resource Reservation Protocol (RSVP) flow.

Methods of classification were once limited to use of the contents of the packet header. Current methods of marking a packet with its classification allow you to set information in the Layer 2, 3, or 4 headers, or even by setting information within the payload of a packet. Criteria for classification of a group might be as broad as "traffic destined for subnetwork X" or as narrow as a single flow. For more information about classifying network traffic, see the "Classifying Network Traffic" chapter.

This chapter explains IP Precedence, and then it gives a brief description of the kinds of traffic classification provided by the Cisco IOS QoS features. It discusses features described in the following sections:

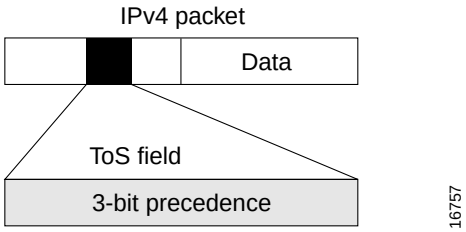
- Committed Access Rate
- Classifying Network Traffic Using NBAR
- Marking Network Traffic

About IP Precedence

Use of IP Precedence allows you to specify the class of service (CoS) for a packet. You use the three precedence bits in the type of service (ToS) field of the IP version 4 (IPv4) header for this purpose. Figure 1 shows the ToS field.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Figure 1 IPv4 Packet Type of Service Field



Using the ToS bits, you can define up to six classes of service. Other features configured throughout the network can then use these bits to determine how to treat the packet. These other QoS features can assign appropriate traffic-handling policies including congestion management strategy and bandwidth allocation. For example, although IP Precedence is not a queueing method, queueing methods such as weighted fair queueing (WFQ) and Weighted Random Early Detection (WRED) can use the IP Precedence setting of the packet to prioritize traffic.

By setting precedence levels on incoming traffic and using them in combination with the Cisco IOS QoS queueing features, you can create differentiated service. You can use features such as policy-based routing (PBR) and committed access rate (CAR) to set precedence based on extended access list classification. These features afford considerable flexibility for precedence assignment. For example, you can assign precedence based on application or user, or by destination and source subnetwork.

So that each subsequent network element can provide service based on the determined policy, IP Precedence is usually deployed as close to the edge of the network or the administrative domain as possible. You can think of IP Precedence as an edge function that allows core, or backbone, QoS features such as WRED to forward traffic based on CoS. IP Precedence can also be set in the host or network client, but this setting can be overridden by policy within the network.

The following QoS features can use the IP Precedence field to determine how traffic is treated:

- Distributed WRED (DWRED)
- WFQ
- CAR

How the IP Precedence Bits Are Used to Classify Packets

You use the three IP Precedence bits in the ToS field of the IP header to specify CoS assignment for each packet. You can partition traffic into a maximum of six classes and then use policy maps and extended access lists to define network policies for congestion handling and bandwidth allocation for each class.

For historical reasons, each precedence corresponds to a name. These names, which continue to evolve, are defined in RFC 791. Table 1 lists the numbers and their corresponding names, from least to most important.

Table 1 IP Precedence Values

Number	Name
0	routine
1	priority
2	immediate
3	flash

Table 1 IP Precedence Values

Number	Name
4	flash-override
5	critical
6	internet
7	network

However, the IP Precedence feature allows you considerable flexibility for precedence assignment. That is, you can define your own classification mechanism. For example, you might want to assign precedence based on application or access router.

Note IP Precedence bit settings 6 and 7 are reserved for network control information such as routing updates.

Setting or Changing the IP Precedence Value

By default, the Cisco IOS software leaves the IP Precedence value untouched, preserving the precedence value set in the header, allowing all internal network devices to provide service based on the IP Precedence setting. This policy follows the standard approach that stipulates that network traffic should be sorted into various types of service at the basic perimeter of the network and that those types of service should be implemented in the core of the network. Routers in the core of the network can then use the precedence bits, for example, to determine the order of transmission, the likelihood of packet drop, and so on.

Because traffic that enters your network can have precedence set by outside devices, we recommend that you reset the precedence for all traffic that enters your network. By controlling IP Precedence settings, you prohibit users that have already set the IP Precedence from acquiring better service for their traffic simply by setting a high precedence for all of their packets.

You can use CAR to set the IP Precedence in packets. As mentioned previously, after a packet has been classified, you can use other QoS features such as CAR and WRED to specify and enforce business policies to fit your business model.

Committed Access Rate

CAR is a multifaceted feature that implements both classification services and policing through rate limiting. This section describes the classification services of CAR. For information on CAR's rate limiting features, see the "Policing and Shaping Overview" chapter.

Note In Cisco IOS Release 12.2 SR, the classification services of CAR are not supported on the Cisco 7600 series router.

You can use the classification services of CAR to set the IP Precedence for packets that enter the network. This capability of CAR allows you to partition your network into multiple priority levels or classes of service. Networking devices within your network can then use the adjusted IP Precedence to determine how to treat the traffic. For example, VIP-distributed WRED uses the IP Precedence to determine the probability a packet being dropped.

As discussed in the"About IP Precedence" section, you can use the three precedence bits in the ToS field of the IP header to define up to six classes of service.

You can classify packets using policies based on physical port, source or destination IP or MAC address, application port, IP protocol type, or other criteria specifiable by access lists or extended access lists. You can classify packets by categories external to the network, for example, by a customer. After a packet has been classified, a network can either accept or override and reclassify the packet according to a specified policy. CAR includes commands that you can use to classify and reclassify packets.

CAR is supported on the majority of Cisco routers. Additionally, distributed CAR is supported on Cisco 7000 series routers with an RSP7000 interface processor or Cisco 7500 series routers with a VIP-based VIP2-40 or greater interface processor. A VIP2-50 interface processor is strongly recommended when the aggregate line rate of the port adapters on the VIP is greater than DS3. A VIP2-50 interface processor is required for OC-3 rates.

For information on how to configure CAR, see the "Configuring Committed Access Rate" chapter.

Classifying Network Traffic Using NBAR

Network-Based Application Recognition (NBAR) is a classification engine that recognizes and classifies a wide variety of protocols and applications. When NBAR recognizes and classifies a protocol or application, the network can be configured to apply the appropriate QoS for that application or traffic with that protocol.

For more information about NBAR, see the "Classifying Network Traffic Using NBAR" chapter.

Marking Network Traffic

Marking network traffic allows you to set or modify the attributes for traffic (that is, packets) belonging to a specific class or category. When used in conjunction with network traffic classification, network traffic marking is the foundation for enabling many QoS features on your network.

For more information about marking network traffic, see the "Marking Network Traffic" chapter.

Configuring Committed Access Rate

This chapter describes the tasks for configuring committed access rate (CAR) and distributed CAR (DCAR).

Note In Cisco IOS Release 12.2 SR, CAR is not supported on the Cisco 7600 series router.

For complete conceptual information about these features, see the "Classification Overview" module and the "Policing and Shaping Overview" module.

For a complete description of the CAR commands in this chapter, see the *Cisco IOS Quality of Service Solutions Command Reference*. To locate documentation of other commands that appear in this module, use the command reference master index or search online.

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

Note CAR and DCAR can only be used with IP traffic. Non-IP traffic is not rate limited.

CAR and DCAR can be configured on an interface or subinterface. However, CAR and DCAR are not supported on the Fast EtherChannel, tunnel, or PRI interfaces, nor on any interface that does not support Cisco Express Forwarding (CEF).

CEF must be enabled on the interface before you configure CAR or DCAR.

CAR is not supported for Internetwork Packet Exchange (IPX) packets.

Committed Access Rate Configuration Task List

The CAR and DCAR services limit the input or output transmission rate on an interface or subinterface based on a flexible set of criteria. CAR is often configured on interfaces at the edge of a network to limit traffic into or out of the network.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

CAR can rate limit traffic based on certain matching criteria, such as incoming interface, IP precedence, or IP access list. You configure the actions that CAR will take when traffic conforms to or exceeds the rate limit.

You can set CAR rate policies that are associated with one of the following:

- All IP traffic
- IP precedence
- MAC address
- IP access list, both standard and extended. Matching to IP access lists is more processor-intensive than matching based on other criteria.

Each interface can have multiple CAR policies, corresponding to different types of traffic. For example, low priority traffic may be limited to a lower rate than high-priority traffic. With multiple rate policies, the router examines each policy in the order entered until the packet matches. If a match is not found, the default action is to send.

The rate policies can be independent; each rate policy deals with a different type of traffic. Alternatively, rate policies can be cascading; a packet can be compared to multiple different rate policies in succession. You can configure up to 100 rate policies on a subinterface.

Note Because of the linear search for the matching rate-limit statement, the CPU load increases with the number of rate policies.

To configure CAR, perform the tasks described in the following sections. The tasks in the first two sections are required; the tasks in the remaining sections are optional.

- Configuring CAR and DCAR for All IP Traffic (Required)
- Configuring CAR and DCAR Policies (Required)
- Configuring a Class-Based DCAR Policy (Optional)
- Monitoring CAR and DCAR (Optional)

See the end of this chapter for the section "CAR and DCAR Configuration Examples."

Configuring CAR and DCAR for All IP Traffic

To configure CAR (or DCAR on Cisco 7000 series routers with RSP7000 or Cisco 7500 series routers with a VIP2-40 or greater interface processor) for all IP traffic, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface interface-type interface-number`	Specifies the interface or subinterface. This command puts the router in interface configuration mode.
Step 2	`Router(config-if)# rate-limit {input | output} bps burst-normal burst-max conform-action action exceed-action action`	Specifies a basic CAR policy for all IP traffic. See Table 1 for a description of conform and exceed *action* keywords.

Basic CAR and DCAR functionality requires that the following criteria be defined:

- Packet direction, incoming or outgoing.

- An average rate, determined by a long-term average of the transmission rate. Traffic that falls under this rate will always conform.
- A normal burst size, which determines how large traffic bursts can be before some traffic is considered to exceed the rate limit.
- An excess burst size (Be). Traffic that falls between the normal burst size and the Excess Burst size exceeds the rate limit with a probability that increases as the burst size increases. CAR propagates bursts. It does no smoothing or shaping of traffic.

Conform and exceed actions are described in Table 1.

Table 1 Rate-Limit Command Action Keywords

Keyword	Description
continue	Evaluates the next **rate-limit** command.
drop	Drops the packet.
set-prec-continue *new-prec*	Sets the IP Precedence and evaluates the next **rate-limit** command.
set-prec-transmit *new-prec*	Sets the IP Precedence and sends the packet.
transmit	Sends the packet.

See the sections "Configuring CAR and DCAR Policies" and "Configuring a Class-Based DCAR Policy" to understand how to configure other CAR and DCAR policy options. See the sections "Subrate IP Services Example" and "Input and Output Rate Limiting on an Interface Example" for examples of how to configure CAR for all IP traffic.

Configuring CAR and DCAR Policies

To configure CAR (orDCAR on Cisco 7000 series routers with the RSP7000 or Cisco 7500 series routers with a VIP2-40 or greater interface processor), use the following commands beginning in interface configuration mode. The tasks listed in this section are required unless noted as optional.

	Command	Purpose
Step 1	`Router(config-if)# interface interface-type interface-number`	Specifies the interface or subinterface. This command puts the router in interface configuration mode.
Step 2	`Router(config-if)# rate-limit {input | output} [access-group [rate-limit] acl-index] bps burst-normal burst-max conform-action action exceed-action action`	Specifies the rate policy for each particular class of traffic. See Table 1 for a description of the **rate-limit** command action keywords. Repeat this command for each different class of traffic.
Step 3	`Router(config-if) exit`	(Optional) Returns to global configuration mode. **Note** This change in configuration mode is needed only if you complete optional Step 4 or Step 5.

	Command	Purpose
Step 4	`Router(config)# access-list rate-limit acl-index {precedence | mac-address | mask prec-mask}`	(Optional) Specifies a rate-limited access list. Repeat this command if you wish to specify a new access list.
Step 5	`Router(config)# access-list acl-index {deny | permit} source [source-wildcard]` or `Router(config)# access-list acl-index {deny | permit} protocol source source-wildcard destination destination-wildcard [precedence precedence][tos tos] [log]`	(Optional) Specifies a standard or extended access list. Repeat this command to further configure the access list or specify a new access list.

The following sections describe requirements for specific policies.

IP Precedence or MAC Address

Use the **access-list rate-limit** command to classify packets using either IP Precedence or MAC addresses. You can then apply CAR policies using the **rate-limit** command to individual rate-limited access lists. Packets with different IP precedences or MAC addresses are treated differently by the CAR service. See the section "Rate Limiting in an IXP Example" for an example of how to configure a CAR policy using MAC addresses.

IP Access List

Use the **access-list** command to define CAR policy based on an access list. The *acl-index* argument is an access list number. Use a number from 1 to 99 to classify packets by precedence or precedence mask. Use a number from 100 to 199 to classify by MAC address.

Note If an access list is not present, the **rate-limit** command will act as if no access list is defined and all traffic will be rate limited accordingly.

See the section "Rate Limiting by Access List Example" for an example of how to configure a CAR policy using IP access lists.

Configuring a Class-Based DCAR Policy

When you configure DCAR on Cisco 7000 series routers with RSP7000 or Cisco 7500 series routers with a VIP2-40 or greater interface processor, you can classify packets by group, to allow you to partition your network into multiple priority levels or classes of service. This classification is achieved by setting IP precedences based on different criteria for use by other QoS features such as Weighted Random Early Detection (WRED) or weighted fair queueing (WFQ).

To configure a class-based DCAR policy, use the following commands beginning in interface configuration mode. The tasks listed in this section are required unless noted as optional.

	Command	Purpose
Step 1	`Router(config-if)# interface interface-type interface-number`	Specifies the interface or subinterface. This command puts the router in interface configuration mode.
Step 2	`Router(config-if)# rate-limit {input | output} [access-group [rate-limit] acl-index] bps burst-normal burst-max conform-action action exceed-action action`	Specifies the rate policy for each particular class of traffic. See Table 1 for a description of the **rate-limit** command action keywords. Repeat this command for each different class of traffic.
Step 3	`Router(config-if)# random-detect precedence precedence min-threshold max-threshold mark-prob-denominator`	Configures WRED and specifies parameters for packets with specific IP Precedence.
Step 4	`Router(config-if)# access-list acl-index {deny | permit} source [source-wildcard]` or `Router(config-if)# access-list acl-index {deny | permit} protocol source source-wildcard destination destination-wildcard [precedence precedence] [tos tos] [log]`	(Optional) Specifies a standard or extended access list. Repeat this command to further configure the access list or specify a new access list.

Monitoring CAR and DCAR

To monitor CAR and DCAR services in your network, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router# show access-lists`	Displays the contents of current IP and rate-limited access lists.
`Router# show access-lists rate-limit [access-list-number]`	Displays information about rate-limited access lists.
`Router# show interfaces [interface-type interface-number] rate-limit`	Displays information about an interface configured for CAR.

CAR and DCAR Configuration Examples

The following sections provide CAR and DCAR configuration examples:

- Subrate IP Services Example
- Input and Output Rate Limiting on an Interface Example
- Rate Limiting in an IXP Example
- Rate Limiting by Access List Example

For information on how to configure CAR and DCAR, see the section "Committed Access Rate Configuration Task List" in this chapter.

Subrate IP Services Example

The following example illustrates how to configure a basic CAR policy that allows all IP traffic. In the example, the network operator delivers a physical T3 link to the customer, but offers a less expensive 15 Mbps subrate service. The customer pays only for the subrate bandwidth, which can be upgraded with additional access bandwidth based on demand. The CAR policy limits the traffic rate available to the customer and delivered to the network to the agreed upon rate limit, plus the ability to temporarily burst over the limit.

```
interface hssi 0/0/0
rate-limit output 15000000 2812500 5625000 conform-action transmit exceed-action drop
ip address 10.1.0.9 255.255.255.0
```

Input and Output Rate Limiting on an Interface Example

In this example, a customer is connected to an Internet service provider (ISP) by a T3 link. The ISP wants to rate limit transmissions from the customer to 15 Mbps of the 45 Mbps. In addition, the customer is allowed to send bursts of 2,812,500 bytes. All packets exceeding this limit are dropped. The following commands are configured on the High-Speed Serial Interface (HSSI) of the ISP connected to the customer:

```
interface Hssi0/0/0
 description 45Mbps to R1
 rate-limit input 15000000 2812500 2812500 conform-action transmit exceed-action drop
 ip address 200.200.14.250 255.255.255.252
 rate-limit output 15000000 2812500 2812500 conform-action transmit exceed-action drop
```

The following sample output shows how to verify the configuration and monitor CAR statistics using the **show interfaces rate-limit** command:

```
Router# show interfaces hssi 0/0/0 rate-limit

Hssi0/0/0 45Mbps to R1
 Input
  matches: all traffic
   params: 15000000 bps, 2812500 limit, 2812500 extended limit
   conformed 8 packets, 428 bytes; action: transmit
   exceeded 0 packets, 0 bytes; action: drop
   last packet: 8680ms ago, current burst: 0 bytes
   last cleared 00:03:59 ago, conformed 0 bps, exceeded 0 bps
 Output
  matches: all traffic
   params: 15000000 bps, 2812500 limit, 2812500 extended limit
   conformed 0 packets, 0 bytes; action: transmit
   exceeded 0 packets, 0 bytes; action: drop
   last packet: 8680ms ago, current burst: 0 bytes
   last cleared 00:03:59 ago, conformed 0 bps, exceeded 0 bps
```

Rate Limiting in an IXP Example

The following example uses rate limiting to control traffic in an Internet Exchange Point (IXP). Because an IXP comprises many neighbors around an FDDI ring, MAC address rate-limited access lists are used to control traffic from a particular ISP. Traffic from one ISP (at MAC address 00e0.34b0.7777) is compared to a rate limit of 80 Mbps of the 100 Mbps available on the FDDI connection. Traffic that conforms to this rate is sent. Nonconforming traffic is dropped.

```
interface Fddi2/1/0
 rate-limit input access-group rate-limit 100 80000000 15000000 30000000 conform-action
  transmit exceed-action drop
 ip address 200.200.6.1 255.255.255.0
!
access-list rate-limit 100 00e0.34b0.7777
```

The following sample output shows how to verify the configuration and monitor the CAR statistics using the **show interfaces rate-limit** command:

```
Router# show interfaces fddi2/1/0 rate-limit

Fddi2/1/0
 Input
  matches: access-group rate-limit 100
   params: 800000000 bps, 15000000 limit, 30000000 extended limit
   conformed 0 packets, 0 bytes; action: transmit
   exceeded 0 packets, 0 bytes; action: drop
   last packet: 4737508ms ago, current burst: 0 bytes
   last cleared 01:05:47 ago, conformed 0 bps, exceeded 0 bps
```

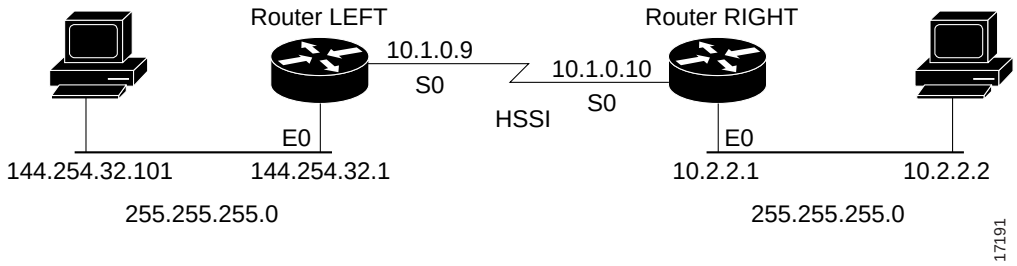
Rate Limiting by Access List Example

The following example shows how CAR can be used to limit the rate by application to ensure capacity for other traffic including mission-critical applications:

- All World Wide Web traffic is sent. However, the IP precedence for Web traffic that conforms to the first rate policy is set to 5. For nonconforming Web traffic, the IP precedence is set to 0 (best effort).
- File Transfer Protocol (FTP) traffic is sent with an IP precedence of 5 if it conforms to the second rate policy. If the FTP traffic exceeds the rate policy, it is dropped.
- Any remaining traffic is limited to 8 Mbps, with a normal burst size of 15,000 bytes and an Excess Burst size of 30,000 bytes. Traffic that conforms is sent with an IP precedence of 5. Traffic that does not conform is dropped.

Figure 1 illustrates the configuration. Notice that two access lists are created to classify the Web and FTP traffic so that they can be handled separately by CAR.

Figure 1 Rate Limiting by Access List



Router LEFT Configuration

```
interface Hssi0/0/0
description 45Mbps to R2
rate-limit output access-group 101 20000000 3750000 7500000 conform-action set-prec-
transmit 5 exceed-action set-prec-transmit 0
rate-limit output access-group 102 10000000 1875000 3750000 conform-action
set-prec-transmit 5 exceed-action drop
```

```
rate-limit output 8000000 1500000 3000000 conform-action set-prec-transmit 5
exceed-action drop
ip address 10.1.0.9 255.255.255.0
!
access-list 101 permit tcp any any eq www
access-list 102 permit tcp any any eq ftp
```

The following sample output shows how to verify the configuration and monitor CAR statistics using the **show interfaces rate-limit** command:

```
Router# show interfaces hssi 0/0/0 rate-limit

Hssi0/0/0 45Mbps to R2
 Input
  matches: access-group 101
   params: 20000000 bps, 3750000 limit, 7500000 extended limit
   conformed 3 packets, 189 bytes; action: set-prec-transmit 5
   exceeded 0 packets, 0 bytes; action: set-prec-transmit 0
   last packet: 309100ms ago, current burst: 0 bytes
   last cleared 00:08:00 ago, conformed 0 bps, exceeded 0 bps
  matches: access-group 102
   params: 10000000 bps, 1875000 limit, 3750000 extended limit
   conformed 0 packets, 0 bytes; action: set-prec-transmit 5
   exceeded 0 packets, 0 bytes; action: drop
   last packet: 19522612ms ago, current burst: 0 bytes
   last cleared 00:07:18 ago, conformed 0 bps, exceeded 0 bps
  matches: all traffic
   params: 8000000 bps, 1500000 limit, 3000000 extended limit
   conformed 5 packets, 315 bytes; action: set-prec-transmit 5
   exceeded 0 packets, 0 bytes; action: drop
   last packet: 9632ms ago, current burst: 0 bytes
   last cleared 00:05:43 ago, conformed 0 bps, exceeded 0 bps
```

Marking Network Traffic

First Published: May 02, 2005
Last Updated: November 4, 2009

Marking network traffic allows you to set or modify the attributes for traffic (that is, packets) belonging to a specific class or category. When used in conjunction with network traffic classification, marking network traffic is the foundation for enabling many quality of service (QoS) features on your network. This module contains conceptual information and the configuration tasks for marking network traffic.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "Feature Information for Marking Network Traffic" section on page 23.

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

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Marking Network Traffic

In order to mark network traffic, Cisco Express Forwarding (CEF) must be configured on both the interface receiving the traffic and the interface sending the traffic.

Restrictions for Marking Network Traffic

Traffic marking can be configured on an interface, a subinterface, or an ATM permanent virtual circuit (PVC). Marking network traffic is not supported on the following interfaces:

- Any interface that does not support CEF
- ATM switched virtual circuit (SVC)
- Fast EtherChannel
- PRI
- Tunnel

Information About Marking Network Traffic

To mark network traffic, you should understand the following concepts:

- Purpose of Marking Network Traffic, page 2
- Benefits of Marking Network Traffic, page 3
- Two Methods for Marking Traffic Attributes, page 4
- MQC and Network Traffic Marking, page 8
- Traffic Classification Compared with Traffic Marking, page 8

Purpose of Marking Network Traffic

Traffic marking is a method used to identify certain traffic types for unique handling, effectively partitioning network traffic into different categories.

After the network traffic is organized into classes by traffic classification, traffic marking allows you to mark (that is, set or change) a value (attribute) for the traffic belonging to a specific class. For instance, you may want to change the class of service (CoS) value from 2 to 1 in one class, or you may want to change the differentiated services code point (DSCP) value from 3 to 2 in another class. In this module, these values are referred to as attributes.

Attributes that can be set and modified include the following:

- Cell loss priority (CLP) bit
- CoS value of an outgoing packet
- Discard eligible (DE) bit setting in the address field of a Frame Relay frame
- Discard-class value
- DSCP value in the type of service (ToS) byte
- MPLS EXP field value in the topmost label on either an input or an output interface

- Multiprotocol Label Switching (MPLS) experimental (EXP) field on all imposed label entries
- Precedence value in the packet header
- QoS group identifier (ID)
- ToS bits in the header of an IP packet

Benefits of Marking Network Traffic

Improved Network Performance

Traffic marking allows you to fine-tune the attributes for traffic on your network. This increased granularity helps single out traffic that requires special handling, and thus, helps to achieve optimal application performance.

Traffic marking allows you to determine how traffic will be treated, based on how the attributes for the network traffic are set. It allows you to segment network traffic into multiple priority levels or classes of service based on those attributes, as follows:

- Traffic marking is often used to set the IP precedence or IP DSCP values for traffic entering a network. Networking devices within your network can then use the newly marked IP precedence values to determine how traffic should be treated. For example, voice traffic can be marked with a particular IP precedence or DSCP and low latency queuing (LLQ) can then be configured to put all packets of that mark into a priority queue. In this case, the marking was used to identify traffic for LLQ.
- Traffic marking can be used to identify traffic for any class-based QoS feature (any feature available in policy-map class configuration mode, although some restrictions exist).
- Traffic marking can be used to assign traffic to a QoS group within a router. The router can use the QoS groups to determine how to prioritize traffic for transmission. The QoS group value is usually used for one of the two following reasons:
  - To leverage a large range of traffic classes. The QoS group value has 100 different individual markings, as opposed to DSCP and Precedence, which have 64 and 8, respectively.
  - If changing the Precedence or DSCP value is undesirable.
- If a packet (for instance, in a traffic flow) needs to be marked to differentiate user-defined QoS services is leaving a router and entering a switch, the router can set the CoS value of the traffic, because the switch can process the Layer 2 CoS header marking. Alternatively, the Layer 2 CoS value of the traffic leaving a switch can be mapped to the Layer 3 IP or MPLS value.
- Weighted random early detection (WRED) uses precedence values or DSCP values to determine the probability that the traffic will be dropped. Therefore, the Precedence and DSCP can be used in conjunction with WRED.

Two Methods for Marking Traffic Attributes

There are two methods for specifying and marking traffic attributes:

- You can specify and mark the traffic attribute by using a **set** command.

  With this method, you configure individual **set** commands for the traffic attribute that you want to mark.

- You can specify and mark the traffic attribute by creating a mapping table (called a "table map").

  With this method, you configure the traffic attributes that you want to mark once in a table map and then the markings can be propagated throughout the network.

These methods are further described in the sections that follow.

Method One: Using a set Command

You specify the traffic attribute you want to change with a **set** command configured in a policy map. Table 1 lists the available **set** commands and the corresponding attribute. Table 1 also includes the network layer and the network protocol typically associated with the traffic attribute.

Table 1 set Commands and Corresponding Traffic Attribute, Network Layer, and Protocol

set Commands[1]	Traffic Attribute	Network Layer	Protocol
set atm-clp	CLP bit	Layer 2	ATM
set cos	Layer 2 CoS value of the outgoing traffic	Layer 2	ATM, Frame Relay
set discard-class	discard-class value	Layer 2	ATM, Frame Relay
set dscp	DSCP value in the ToS byte	Layer 3	IP
set fr-de	DE bit setting in the address field of a Frame Relay frame	Layer 2	Frame Relay
set ip tos (route-map)	ToS bits in the header of an IP packet	Layer 3	IP
set mpls experimental imposition	MPLS EXP field on all imposed label entries	Layer 3	MPLS
set mpls experimental topmost	MPLS EXP field value in the topmost label on either an input or an output interface	Layer 3	MPLS
set precedence	precedence value in the packet header	Layer 3	IP
set qos-group	QoS group ID	Layer 3	IP, MPLS

1. Cisco IOS **set** commands can vary by release. For more information, see the command documentation for the Cisco IOS release that you are using

If you are using individual **set** commands, those **set** commands are specified in a policy map. The following is a sample of a policy map configured with one of the **set** commands listed in Table 3.

In this sample configuration, the **set atm-clp** command has been configured in the policy map (policy1) to mark the CLP attribute.

```
policy-map policy1
 class class1
 set atm-clp
 end
```

For information on configuring a policy map, see the "Creating a Policy Map for Applying a QoS Feature to Network Traffic" section on page 11.

The final task is to attach the policy map to the interface. For information on attaching the policy map to the interface, see the "Attaching the Policy Map to an Interface" section on page 14.

Method Two: Using a Table Map

You can create a table map that can be used to mark traffic attributes. A table map is a kind of two-way conversion chart that lists and maps one traffic attribute to another. A table map supports a many-to-one type of conversion and mapping scheme. The table map establishes a to-from relationship for the traffic attributes and defines the change to be made to the attribute. That is, an attribute is set *to* one value that is taken *from* another value. The values are based on the specific attribute being changed. For instance, the Precedence attribute can be a number from 0 to 7, while the DSCP attribute can be a number from 0 to 63.

The following is a sample table map configuration:

```
table-map table-map1
 map from 0 to 1
 map from 2 to 3
 exit
```

Table 2 lists the traffic attributes for which a to-from relationship can be established using the table map.

Table 2 Traffic Attributes for Which a To–From Relationship Can Be Established

The "To" Attribute	The "From" Attribute
Precedence	CoS
	QoS group
DSCP	CoS
	QoS group
CoS	Precedence
	DSCP
QoS group	Precedence
	DSCP
	MPLS EXP topmost
MPLS EXP topmost	QoS group
MPLS EXP imposition	Precedence
	DSCP

For information on creating a table map, see the "Creating a Table Map for Marking Network Traffic" section on page 10.

Once the table map is created, you configure a policy map to use the table map. In the policy map, you specify the table map name and the attributes to be mapped by using the **table** keyword and the *table-map-name* argument with one of the commands listed in Table 3.

Table 3 Commands Used in Policy Maps to Map Attributes

Command Used in Policy Maps	Maps These Attributes
set cos dscp table *table-map-name*	CoS to DSCP
set cos precedence table *table-map-name*	CoS to Precedence
set dscp cos table *table-map-name*	DSCP to CoS
set dscp qos-group table *table-map-name*	DSCP to qos-group
set mpls experimental imposition dscp table *table-map-name*	MPLS EXP imposition to DSCP
set mpls experimental imposition precedence table *table-map-name*	MPLS EXP imposition to precedence
set mpls experimental topmost qos-group table *table-map-name*	MPLS EXP topmost to QoS-group
set precedence cos table *table-map-name*	Precedence to CoS
set precedence qos-group table *table-map-name*	Precedence to QoS-group
set qos-group dscp table *table-map-name*	QoS-group to DSCP
set qos-group mpls exp topmost table *table-map-name*	QoS-group to MPLS EXP topmost
set qos-group precedence table *table-map-name*	QoS-group to Precedence

The following is an example of a policy map (policy2) configured to use the table map (table-map1) created earlier:

```
policy map policy2
 class class-default
 set cos dscp table table-map1
 exit
```

In this example, a mapping relationship was created between the CoS attribute and the DSCP attribute as defined in the table map.

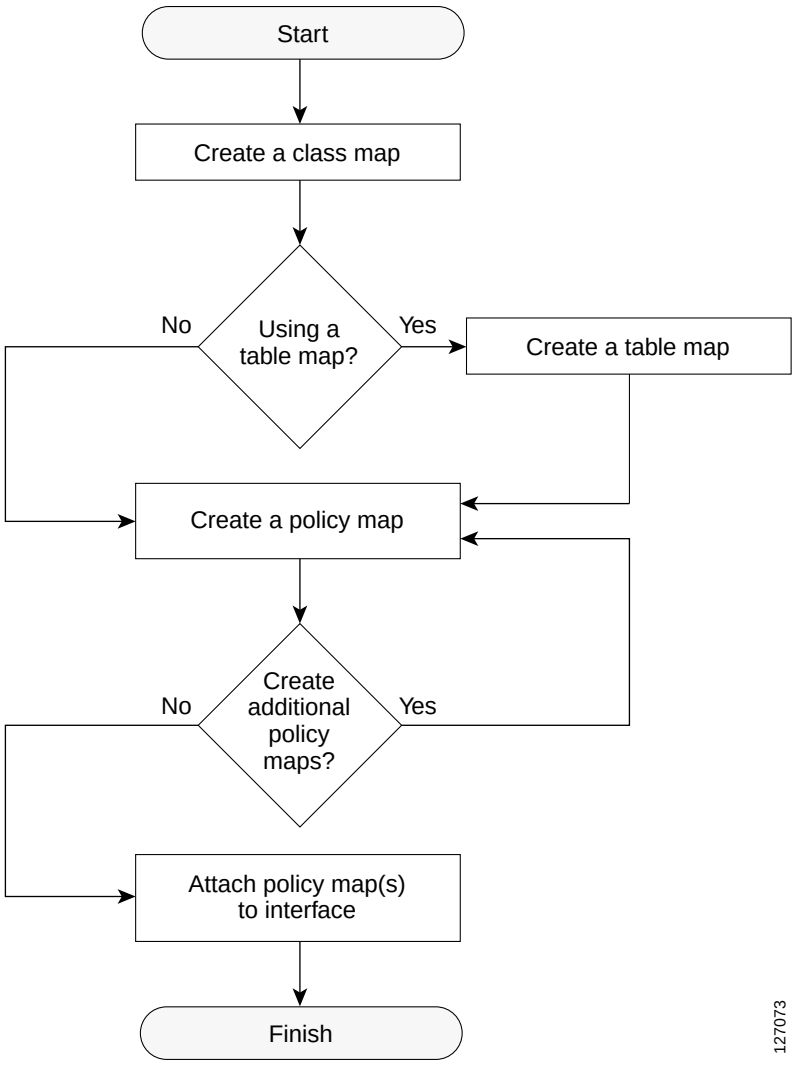
For information on configuring a policy map to use a table map, see the "Creating a Policy Map for Applying a QoS Feature to Network Traffic" section on page 11.

The final task is to attach the policy map to the interface. For information on attaching the policy map to the interface, see the "Attaching the Policy Map to an Interface" section on page 14.

Traffic Marking Procedure Flowchart

Figure 1 illustrates the order of the procedures for configuring traffic marking.

Figure 1 Traffic Marking Procedure Flowchart



For more information on class maps and policy maps, see the "MQC and Network Traffic Marking" section on page 8.

For more information on table maps, see the "Creating a Table Map for Marking Network Traffic" section on page 10.

For more information on completing the processes shown in this flow chart, see the "How to Mark Network Traffic" section on page 9.

MQC and Network Traffic Marking

To configure network traffic marking, you use the Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC).

The MQC is a CLI structure that allows you to complete the following tasks:

- Specify the matching criteria used to define a traffic class.
- Create a traffic policy (policy map). The traffic policy defines the QoS policy actions to be taken for each traffic class.
- Apply the policy actions specified in the policy map to an interface, subinterface, or ATM PVC by using the **service-policy** command.

For more information about the MQC, see the "Applying QoS Features Using the MQC" module.

Traffic Classification Compared with Traffic Marking

Traffic classification and traffic marking are closely related and can be used together. Traffic marking can be viewed as an additional action, specified in a policy map, to be taken on a traffic class.

Traffic classification allows you to organize into traffic classes on the basis of whether the traffic matches specific criteria. For example, all traffic with a CoS value of 2 is grouped into one class, and traffic with DSCP value of 3 is grouped into another class. The match criterion is user-defined.

After the traffic is organized into traffic classes, traffic marking allows you to mark (that is, set or change) an attribute for the traffic belonging to that specific class. For instance, you may want to change the CoS value from 2 to 1, or you may want to change the DSCP value from 3 to 2.

The match criteria used by traffic classification are specified by configuring a **match** command in a class map. The marking action taken by traffic marking is specified by configuring a **set** command in a policy map. These class maps and policy maps are configured using the MQC.

Table 4 compares the features of traffic classification and traffic marking.

Table 4 Traffic Classification Compared with Traffic Marking

	Traffic Classification	Traffic Marking
Goal	Groups network traffic into specific traffic classes on the basis of whether the traffic matches the user-defined criterion.	After the network traffic is grouped into traffic classes, modifies the attributes for the traffic in a particular traffic class.
Configuration Mechanism	Uses class maps and policy maps in the MQC.	Uses class maps and policy maps in the MQC.
CLI	In a class map, uses **match** commands (for example, **match cos**) to define the traffic matching criterion.	Uses the traffic classes and matching criterion specified by traffic classification. In addition, uses **set** commands (for example, **set cos**) in a policy map to modify the attributes for the network traffic. If a table map was created, uses the **table** keyword and *table-map-name* argument with the **set** commands (for example, **set cos precedence table** *table-map-name*) in the policy map to establish the to–from relationship for mapping attributes.

How to Mark Network Traffic

This section contains the following procedures.

- Creating a Class Map for Marking Network Traffic, page 9 (required)
- Creating a Table Map for Marking Network Traffic, page 10 (optional)
- Creating a Policy Map for Applying a QoS Feature to Network Traffic, page 11 (required)
- Attaching the Policy Map to an Interface, page 14 (required)
- Configuring QoS When Using IPsec VPNs, page 16 (optional)

Creating a Class Map for Marking Network Traffic

In this procedure, you create a class map to define traffic classes. Within the class map, the appropriate **match** command is used to specify the matching criteria for the traffic classes.

To create the class map and specify the matching criteria, complete the following steps.

Note The **match fr-dlci** command is included in the steps below. The **match fr-dlci** command is just an example of one of the **match** commands that can be used. See the command documentation for the Cisco IOS release that you are using for a complete list of **match** commands.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** *class-map-name* [**match-all** | **match-any**]
4. **match fr-dlci** *dlci-number*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`class-map` *class-map-name* `[match-all | match-any]` **Example:** `Router(config)# class-map class1`	Creates a class map to be used for matching traffic to a specified class and enters class-map configuration mode. • Enter the class map name.

	Command or Action	Purpose
Step 4	**match fr-dlci** *dlci-number* **Example:** `Router(config-cmap)# match fr-dlci 500`	(Optional) Specifies the Frame Relay DLCI number as a match criterion in a class map. **Note** The **match fr-dlci** command classifies traffic on the basis of the Frame RelayDLCI number. The **match fr-dlci** command is just an example of one of the **match** commands that can be used. The **match** commands vary by Cisco IOS release. See the command documentation for the Cisco IOS release that you are using for a complete list of **match** commands.
Step 5	**end** **Example:** `Router(config-cmap)# end`	(Optional) Returns to privileged EXEC mode.

Creating a Table Map for Marking Network Traffic

Note If you are not using a table map, skip this procedure and advance to "Creating a Policy Map for Applying a QoS Feature to Network Traffic" section on page 11.

The table map contains the mapping scheme used for establishing the to-from relationship and equivalency between one traffic-marking value and another.

The table map can be configured for use with *multiple* policy maps. The policy maps can then be configured to convert and propagate the traffic-marking values defined in the table map. Then the policy maps can be attached to the input or output interface of either the ingress or egress router, as appropriate to serve the QoS requirements of your network.

To create and configure the table map, complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **table-map** *table-map-name* **map from** *from-value* **to** *to-value* [**default** *default-action-or-value*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**table-map** *table-map-name* **map from** *from-value* **to** *to-value* [**default** *default-action-or-value*] **Example:** `Router(config)# table-map table-map1 map from 2 to 1`	Creates a table map using the specified name and enters tablemap configuration mode. • Enter the name of the table map you want to create. • Enter each value mapping on a separate line. Enter as many separate lines as needed for the values you want to map. • The **default** keyword and *default-action-or-value* argument set the default value (or action) to be used if a value is not explicitly designated.
Step 4	**end** **Example:** `Router(config-tablemap)# end`	(Optional) Exits tablemap configuration mode and returns to privileged EXEC mode.

Creating a Policy Map for Applying a QoS Feature to Network Traffic

In this procedure, you create and configure a policy map to use the class map (or the table map). The policy map applies the appropriate QoS feature to the network traffic based on the traffic classification.

To create a policy map, complete the following steps.

Restrictions

- The **set atm-clp** command is supported on the following adapters only:
  - Enhanced ATM Port Adapter (PA-A3)
  - ATM Inverse Multiplexer over ATM Port Adapter with 8 T1 Ports (PA-A3-8T1IMA)
  - ATM Inverse Multiplexer over ATM Port Adapter with 8 E1 Ports (PA-A3-8E1IMA)
- Before modifying the encapsulation type from IEEE 802.1 Q to ISL, or vice versa, on a subinterface, detach the policy map from the subinterface. After changing the encapsulation type, reattach the policy map.
- A policy map containing the **set qos-group** command can only be attached as an input traffic policy. QoS group values are not usable for traffic leaving a router.

- A policy map containing the **set cos** command can only be attached as an output traffic policy.
- A policy map containing the **set atm-clp** command can be attached as an output traffic policy only. The **set atm-clp** command does not support traffic that originates from the router.

Note The **set cos** command and **set cos dscp table** *table-map-name* command are shown in the steps that follow. The **set cos** command and **set cos dscp table** *table-map-name* command are examples the **set** commands that can be used when marking traffic. Other **set** commands can be used. For a list of other **set** commands, see Table 1 on page 4 and Table 3 on page 6.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** {*class-name* | **class-default**}
5. **set cos** *cos-value*

   or

   set cos dscp table *table-map-name*
6. **end**
7. **show policy-map**

   or

   show policy-map *policy-map* **class** *class-name*
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map-name* **Example:** `Router(config)# policy-map policy1`	Specifies the name of the policy map created earlier and enters policy-map configuration mode. • Enter the policy map name.
Step 4	**class** {*class-name* | **class-default**} **Example:** `Router(config-pmap)# class class1`	Specifies the name of the class whose policy you want to create and enters policy-map class configuration mode. This class is associated with the class map created earlier. • Enter the name of the class or enter the **class-default** keyword.

	Command or Action	Purpose
Step 5	**set cos** *cos-value* or **set cos dscp table** *table-map-name* **Example:** `Router(config-pmap-c)# set cos 2` or **Example:** `Router(config-pmap-c)# set cos dscp table table-map1`	(Optional) Sets the CoS value in the type of service (ToS) byte. **Note** The **set cos** command is an example of one of the **set** commands that can be used when marking traffic. Other **set** commands can be used. For a list of other **set** commands, see Table 1 on page 4. or (Optional) If a table map has been created earlier, sets the CoS value based on the DSCP value (or action) defined in the table map. **Note** The **set cos dscp table** *table-map-name* command is an example of one of the commands that can be used. For a list of other commands, see Table 3 on page 6.
Step 6	**end** **Example:** `Router(config-pmap-c)# end`	Returns to privileged EXEC mode.
Step 7	**show policy-map** or **show policy-map** *policy-map* **class** *class-name* **Example:** `Router# show policy-map` or **Example:** `Router# show policy-map policy1 class class1`	(Optional) Displays all configured policy maps. or (Optional) Displays the configuration for the specified class of the specified policy map. • Enter the policy map name and the class name.
Step 8	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

What to Do Next

Create and configure as many policy maps as you need for your network. To create and configure additional policy maps, repeat the steps in the "Creating a Policy Map for Applying a QoS Feature to Network Traffic" section on page 11. Then attach the policy maps to the appropriate interface, following the instructions in the "Attaching the Policy Map to an Interface" section on page 14.

Attaching the Policy Map to an Interface

After you create the policy map, you must attach it to an interface. Policy maps can be attached to either the input or output direction of the interface.

Note Depending on the needs of your network, policy maps can be attached to an interface, a subinterface, or an ATM permanent virtual circuit (PVC).

To attach the policy map, complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [**name-tag**]
4. **pvc** [*name*] *vpi/vci* [**ilmi | qsaal | smds | l2transport**]
5. **exit**
6. **service-policy** {**input | output**} *policy-map-name*
7. **end**
8. **show policy-map interface** *interface-name*
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`interface` *type number* `[name-tag]` **Example:** `Router(config)# interface serial4/0`	Configures an interface type and enters interface configuration mode. • Enter the interface type and number.

	Command or Action	Purpose
Step 4	**pvc** [*name*] *vpi/vci* [**ilmi** |**qsaal** |**smds** | **l2transport**] **Example:** `Router(config-if)# pvc cisco 0/16`	(Optional) Creates or assigns a name to an ATM permanent virtual circuit (PVC), specifies the encapsulation type on an ATM PVC, and enters ATM virtual circuit configuration mode. • Enter the PVC name, the ATM network virtual path identifier, and the network virtual channel identifier. **Note** This step is required only if you are attaching the policy map to an ATM PVC. If you are not attaching the policy map to an ATM PVC, advance to Step 6.
Step 5	**exit** **Example:** `Router(config-atm-vc)# exit`	(Optional) Returns to interface configuration mode. **Note** This step is required only if you are attaching the policy map to an ATM PVC and you completed Step 4. If you are not attaching the policy map to an ATM PVC, advance to Step 6.
Step 6	**service-policy** {**input** | **output**} *policy-map-name* **Example:** `Router(config-if)# service-policy input policy1`	Attaches a policy map to an input or output interface. • Enter the policy map name. **Note** Policy maps can be configured on ingress or egress routers. They can also be attached in the input or output direction of an interface. The direction (input or output) and the router (ingress or egress) to which the policy map should be attached varies according your network configuration. When using the **service-policy** command to attach the policy map to an interface, be sure to choose the router and the interface direction that are appropriate for your network configuration.
Step 7	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.
Step 8	**show policy-map interface** *type number* **Example:** `Router# show policy-map interface serial4/0`	(Optional) Displays traffic statistics of all classes configured for all service policies on the specified interface, subinterface, or PVC on the interface. When there are multiple instances of the same class in a policy-map, and this policy-map is attached to an interface, **show policy-map interface** <interface_name> **output class** <class-name> returns only the first instance. • Enter the interface type and number.
Step 9	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

Configuring QoS When Using IPsec VPNs

Note This task is required only if you are using IPsec Virtual Private Networks (VPNs). Otherwise, this task is not necessary. For information about IPsec VPNs, see the "Configuring Security for VPNs with IPsec" module.

To configure QoS when using IPsec VPNs, complete the following steps.

Restrictions

This task uses the **qos pre-classify** command to enable QoS preclassification for the packet. QoS preclassification is not supported for all fragmented packets. If a packet is fragmented, each fragment might received different preclassifications.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-num*
4. **exit**
5. **interface** *type number* [**name-tag**]
6. **qos pre-classify**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name seq-num* **Example:** `Router(config)# crypto map mymap 10`	Enters crypto map configuration mode and creates or modifies a crypto map entry. • Enter the crypto map name and sequence number.
Step 4	**exit** **Example:** `Router(config-crypto-map)# exit`	Returns to global configuration mode.
Step 5	**interface** *type number* [**name-tag**] **Example:** `Router(config)# interface serial4/0`	Configures an interface type and enters interface configuration mode. • Enter the interface type and number.
Step 6	**qos pre-classify** **Example:** `Router(config-if)# qos pre-classify`	Enables QoS preclassification.
Step 7	**end** **Example:** `Router(config-if)# end`	(Optional) Exits interface configuration mode andreturns to privileged EXEC mode.

Configuration Examples for Marking Network Traffic

This section contains the following examples:

Creating a Class Map for Marking Network Traffic: Example

The following is an example of creating a class map to be used for marking network traffic. In this example, a class called class1 has been created. The traffic with a Frame Relay DLCI value of 500 will be put in this class.

```
Router> enable
Router# configure terminal
Router(config)# class-map class1
Router(config-cmap)# match fr-dlci 500
Router(config-cmap)# end
```

Creating a Table Map for Marking Network Traffic: Example

In the following example, the **table-map** (value mapping) command has been used to create and configure a table map called table-map1. This table map will be used to establish a to-from relationship between one traffic-marking value and another.

In table-map1, a traffic-marking value of 0 will be mapped to a value of 1.

```
Router> enable
Router# configure terminal
Router(config)# table-map table-map1 map from 0 to 1
Router(config-tablemap)# end
```

Creating a Policy Map for Applying a QoS Feature to Network Traffic: Examples

Policy Map Configured to Use set Command

The following is an example of creating a policy map to be used for traffic marking. In this example, a policy map called policy1 has been created, and the **set dscp** command has been configured for class1.

```
Router> enable
Router# configure terminal
Router(config)# policy-map policy1
Router(config-pmap)# class class1
Router(config-pmap-c)# set dscp 2
Router(config-pmap-c)# end
```

Policy Map Configured to Use a Table Map

A policy map called policy1 has been created and configured to use table-map1 for setting the precedence value. In this example, the CoS value will be set according to the DSCP value defined in table-map1 created previously.

```
Router(config)# policy map policy1
Router(config-pmap)# class class-default
Router(config-pmap-c)# set cos dscp table table-map1
Router(config-pmap-c)# end
```

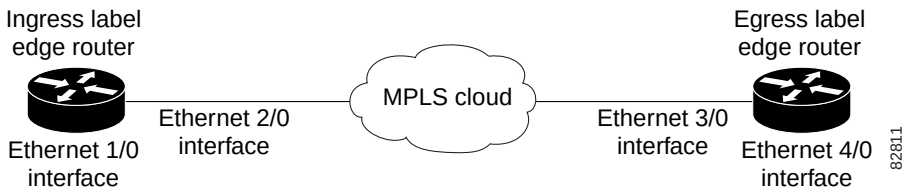
Note As an alternative to configuring the **set cos dscp table table-map1** command shown in the example, you could configure the command without specifying the **table** keyword and the applicable *table-map-name* argument (that is, you could configure the **set cos dscp** command). When the command is configured without the **table** keyword and applicable table map name, the values are copied from the specified categories. In this case, the DSCP value is copied and used to set the CoS value.

When the DSCP value is copied and used for the CoS value only the *first 3 bits* (that is, the class selector bits) of the DSCP value will be used to set the CoS value. For example, if the DSCP value is EF (101110), the first 3 bits of this DSCP value will be used to set the CoS value, resulting in a CoS value of 5 (101).

Policy Map Configured to Use a Table Map for Mapping MPLS EXP Values

This section contains an example of a policy map configured to map MPLS experimental (EXP) values. Figure 2 illustrates the network topology for this configuration example.

Figure 2 Network Topology for Mapping MPLS EXP Value



For this configuration example, traffic arrives at the input interface (an Ethernet 1/0 interface) of the ingress label edge router (LER). The precedence value is copied and used as the MPLS EXP value of the traffic when the MPLS label is imposed. This label imposition takes place at the ingress LER.

The traffic leaves the ingress LER through the output interface (an Ethernet 2/0 interface), traverses through the network backbone into the MPLS cloud, and enters the egress LER.

At the input interface of the egress LER (an Ethernet 3/0 interface), the MPLS EXP value is copied and used as the QoS group value. At the output interface of the egress LER (an Ethernet 4/0 interface), the QoS group value is copied and used as the precedence value.

To accomplish configuration described above, three separate policy maps were required—policy1, policy2, and policy3. Each policy map is configured to convert and propagate different traffic-marking values.

The first policy map, policy1, is configured to copy the precedence value of the traffic and use it as the MPLS EXP value during label imposition.

```
Router(config)# policy-map policy1
Router(config-pmap)# class class-default
Router(config-pmap-c)# set mpls experimental imposition precedence
Router(config-pmap-c)# end
```

When the traffic leaves the LER through the output interface (the Ethernet 2/0 interface), the MPLS EXP value is copied from the precedence value during MPLS label imposition. Copying the MPLS EXP value from the precedence value ensures that the MPLS EXP value reflects the appropriate QoS treatment. The traffic now proceeds through the MPLS cloud into the egress LER.

A second policy map called policy2 has been configured to copy the MPLS EXP value in the incoming MPLS traffic to the QoS group value. The QoS group value is used for internal purposes only. The QoS group value can be used with output queueing on the output interface of the egress router. The QoS group value can also be copied and used as the precedence value, as traffic leaves the egress LER through the output interface (the Ethernet 4/0 interface).

```
Router(config)# policy-map policy2
Router(config-pmap)# class class-default
Router(config-pmap-c)# set qos-group mpls experimental topmost
Router(config-pmap-c)# end
```

A third policy map called policy3 has been configured to copy the internal QoS group value (previously based on the MPLS EXP value) to the precedence value. The QoS group value will be copied to the precedence value as the traffic leaves the egress LER through the output interface.

```
Router(config)# policy-map policy3
Router(config-pmap)# class class-default
Router(config-pmap-c)# set precedence qos-group
Router(config-pmap-c)# end
```

Configuring these policy maps as shown (and attaching them to interfaces as shown in "Attaching the Policy Map to an Interface: Example" section on page 20), causes the appropriate quality of service treatment to be preserved for the traffic as the traffic progresses along an IP network, through an MPLS cloud, and back again into an IP network.

Note This configuration could also have been accomplished by first creating a table map (used to map one value to another) and then specifying the **table** keyword and *table-map-name* argument in each of the **set** commands (for example, **set precedence qos-group table tablemap1**).

In the MPLS configuration example, a table map was not created, and the **set** commands were configured without specifying the **table** keyword and *table-map-name* argument (for example, **set precedence qos-group)**.

When the **set** commands are configured without specifying the **table** keyword and *table-map-name* argument, the values are copied from the specified categories. In this case, the QoS group value was copied and used to set the precedence value.

When the DSCP value is copied and used for the MPLS EXP value, only the *first 3 bits* (that is, the class selector bits) of the DSCP value will be used to set the MPLS value.

Attaching the Policy Map to an Interface: Example

The following is an example of attaching the policy map to the interface. In this example, the policy map called policy1 has been attached in the input direction of the Serial4/0 interface.

```
Router> enable
Router# configure terminal
Router(config)# interface serial4/0
Router(config-if)# service-policy input policy1
Router(config-if)# end
```

Configuring QoS When Using IPsec VPNs: Example

The following is an example of configuring QoS when using IPsec VPNs. In this example, the **crypto map** command specifies the IPsec crypto map (mymap 10) to which the **qos pre-classify** command will be applied.

```
Router> enable
Router# configure terminal
Router(config)# crypto map mymap 10
Router(config-crypto-map)# exit
Router(config)# interface serial4/0
Router(config-if)# qos pre-classify
Router(config-if)# end
```

Additional References

The following sections provide references related to marking network traffic.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
MQC	"Applying QoS Features Using the MQC" module
CEF	"Cisco Express Forwarding Features Roadmap" module
Classifying network traffic	"Classifying Network Traffic" module
IPsec and VPNs	"Configuring Security for VPNs with IPsec" module
Committed Access Rate (CAR)	"Configuring Committed Access Rate" module

Standards

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

MIBs

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

RFCs

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

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Marking Network Traffic

Table 5 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Releases 12.2(1)T or later appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for specific commands was introduced, see the command reference documents.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 5 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 5 Feature Information for Marking Network Traffic

Feature Name	Software Releases	Feature Configuration Information
Enhanced Packet Marking	12.2(13)T	The Enhanced Packet Marking feature allows you to map and convert the marking of a packet from one value to another by using a kind of conversion chart called a table map. The table map establishes an equivalency from one value to another. For example, the table map can map and convert the class of service (CoS) value of a packet to the precedence value of the packet. This value mapping can be propagated for use on the network, as needed. The following sections provide information about this feature: • Information About Marking Network Traffic, page 2 • How to Mark Network Traffic, page 9
QoS Packet Marking	12.2(8)T	The QoS Packet Marking feature allows you to mark packets by setting the IP precedence bit or the IP differentiated services code point (DSCP) in the Type of Service (ToS) byte, and associate a local QoS group value with a packet. The following sections provide information about this feature: • Information About Marking Network Traffic, page 2 • How to Mark Network Traffic, page 9

Table 5 Feature Information for Marking Network Traffic (continued)

Feature Name	Software Releases	Feature Configuration Information
Class-Based Marking	12.2(2)T	The Class-Based Packet Marking feature provides users with a user-friendly command-line interface (CLI) for efficient packet marking by which users can differentiate packets based on the designated markings. The following sections provide information about this feature: • Information About Marking Network Traffic, page 2 • How to Mark Network Traffic, page 9
Quality of Service for Virtual Private Networks	12.2(2)T	The QoS for VPNs 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 the data is encrypted and tunneled. The QoS for VPN feature allows users to look inside the packet so that packet marking can be done based on original port numbers and based on source and destination IP addresses. This allows the service provider to treat mission critical or multi-service traffic with higher priority across their network. The following sections provide information about this feature: • Configuring QoS When Using IPsec VPNs, page 16 • Configuring QoS When Using IPsec VPNs: Example, page 21

Glossary

ATM—Asynchronous Transfer Mode. The international standard for cell relay in which multiple service types (such as voice, video, or data) are conveyed in fixed-length (53-byte) cells. Fixed-length cells allow cell processing to occur in hardware, thereby reducing transit delays. ATM is designed to take advantage of high-speed transmission media, such as E3, SONET, and T3.

CoS—class of service. An indication of how an upper-layer protocol requires a lower-layer protocol to treat its messages. A CoS definition comprises a virtual route number and a transmission priority field.

DLCI—data-link connection identifier. A value that specifies a PVC or a switched virtual circuit (SVC) in a Frame Relay network. In the basic Frame Relay specification, DLCIs are locally significant (connected devices might use different values to specify the same connection). In the Local Management Interface (LMI) extended specification, DLCIs are globally significant (DLCIs specify individual end devices).

IPsec—IP Security. A framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPsec provides these security services at the IP layer. IPsec uses Internet Key Exchange (IKE) to handle the negotiation of protocols and algorithms based on local policy and to generate the encryption and authentication keys to be used by IPsec. IPsec can protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

LER—label edge router. LERs are typically used in a Multiprotocol Label Switching network.

MPLS—Multiprotocol Label Switching. A switching method that forwards IP traffic using a label. This label instructs the routers and the switches in the network where to forward the packets based on preestablished IP routing information

PVC—permanent virtual circuit (or connection). A virtual circuit that is permanently established. PVCs save bandwidth associated with circuit establishment and tear down in situations where certain virtual circuits must exist all the time. In ATM terminology, called a permanent virtual connection.

VPN—Virtual Private Network. A network that enables traffic to travel securely over a public or shared network. An IPsec VPN uses encryption and tunneling, encapsulating private IP packets into IPsec-encrypted packets to protect information at the IP level.

QoS: Tunnel Marking for GRE Tunnels

First Published: December 31, 2007
Last Updated: April 18, 2008

The QoS: Tunnel Marking for GRE Tunnels feature introduces the capability to define and control the quality of service (QoS) for incoming customer traffic on the provider edge (PE) router in a service provider network.

Note For Cisco IOS Release 124(15)T2, the QoS: Tunnel Marking for GRE Tunnels feature is supported only on platforms equipped with a Cisco MGX Route Processor Module (RPM-XF).

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for QoS: Tunnel Marking for GRE Tunnels" section on page 15.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for QoS: Tunnel Marking for GRE Tunnels

- You must configure Cisco Express Forwarding (CEF) on the interface before GRE tunnel marking can be used.

  For information on CEF switching, see the "CEF Feature Roadmap" module.
- You must determine the topology and interfaces that need to be configured to mark incoming traffic.

Restrictions for QoS: Tunnel Marking for GRE Tunnels

- GRE tunnel marking is supported in input policy maps only and should not be configured for output policy maps.
- It is possible to configure GRE tunnel marking and the **ip tos** command at the same time. However, Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC) GRE tunnel marking has higher priority over IP ToS commands, meaning that tunnel marking always rewrites the IP header of the tunnel packet and overwrites the values set by **ip tos** commands. The priority of enforcement is as follows when these commands are used simultaneously:
  1. **set ip dscp tunnel** or **set ip precedence tunnel** (GRE tunnel marking)
  2. **ip tos reflect**
  3. **ip tos** *tos-value*

Note This is the designed behavior. We recommend that you configure only GRE tunnel marking and reconfigure any peers configured with the **ip tos** command to use GRE tunnel marking.

Information About QoS: Tunnel Marking for GRE Tunnels

To configure the QoS: Tunnel Marking for GRE Tunnels feature, you should understand the following concepts:

GRE Definition

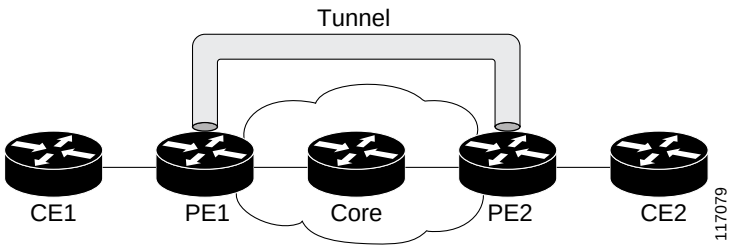
Generic Routing Encapsulation (GRE) is a tunneling protocol developed by Cisco that can encapsulate a wide variety of protocol packet types inside IP tunnels, creating a virtual point-to-point link to Cisco routers at remote points over an IP internetwork.

GRE Tunnel Marking Overview

The QoS: Tunnel Marking for GRE Tunnels feature allows you to define and control QoS for incoming customer traffic on the PE router in a service provider (SP) network. This feature lets you set (mark) either the IP precedence value or the differentiated services code point (DSCP) value in the header of an GRE tunneled packet. GRE tunnel marking can be implemented by using a QoS marking command, such as **set ip** {**dscp** | **precedence**} [**tunnel**], and it can also be implemented in QoS traffic policing. This feature simplifies administrative overhead previously required to control customer bandwidth by allowing you to mark the GRE tunnel header on the incoming interface on the PE routers.

Figure 1 shows traffic being received from the CE1 router through the incoming interface on the PE1 router on which tunnel marking occurs. The traffic is encapsulated (tunneled), and the tunnel header is marked on the PE1 router. The marked packets travel (tunnel) through the core and are decapsulated automatically on the exit interface of the PE2 router. This feature is designed to simplify classifying customer edge (CE) traffic and is configured only in the service provider network. This process is transparent to the customer sites. The CE1 and CE2 routers simply exist as a single network.

Figure 1 Sample Tunnel Marking Topology



GRE Tunnel Marking and the MQC

To configure the tunnel marking for GRE tunnels, you must configure a class map and a policy map and then attach that policy map to the appropriate interface. These three tasks can be accomplished by using the MQC.

For information on using the MQC, see the "Applying QoS Features Using the MQC" module.

GRE Tunnel Marking and DSCP or IP Precedence Values

GRE tunnel marking is configured with the **set ip precedence tunnel** or **set ip dscp tunnel** command on PE routers that carry incoming traffic from customer sites. GRE tunnel marking allows you to mark the header of a GRE tunnel by setting a DSCP value from 0 to 63 or an IP precedence value from 0 to 7 to control GRE tunnel traffic bandwidth and priority.

GRE traffic can also be marked under traffic policing with the **set-dscp-tunnel-transmit** and the **set-prec-tunnel-transmit** actions (or keywords) of the **police** command. The tunnel marking value is from 0 to 63 for the **set-dscp-tunnel-transmit** actions and from 0 to 7 for the **set-prec-tunnel-transmit** command. Under traffic policing, tunnel marking can be applied with "conform" and "exceed" action statements, allowing you to automatically apply a different value for traffic that does not conform to the expected traffic rate.

After the tunnel header is marked, GRE traffic is carried through the tunnel and across the service provider network. This traffic is decapsulated on the interface of the PE router that carries the outgoing traffic to the other customer site. The configuration of GRE tunnel marking is transparent to customer sites. All internal configuration is preserved.

It is important to distinguish between the **set ip precedence** and **set ip dscp** commands and the **set ip precedence tunnel** and **set ip dscp tunnel** commands.

- The **set ip precedence** and **set ip dscp** commands are used to set the IP precedence value or DSCP value in the header of an IP packet.
- The **set ip precedence tunnel** and **set ip dscp tunnel** commands are used to set (mark) the IP precedence value or DSCP value in the tunnel header that encapsulates the GRE traffic.

Benefits of GRE Tunnel Marking

GRE tunnel marking provides a simple mechanism to control the bandwidth of customer GRE traffic. The QoS: Tunnel Marking for GRE Tunnels feature is configured entirely within the service provider network and only on interfaces that carry incoming traffic on the PE routers.

How to Configure Tunnel Marking for GRE Tunnels

The QoS: Tunnel Marking for GRE Tunnels feature introduces the capability for a service provider to define and control customer traffic bandwidth and priority on the interfaces of PE routers that carry incoming traffic. This section contains the following procedures:

- Configuring a Class Map, page 4 (required)
- Creating a Policy Map, page 5 (required)
- Attaching the Policy Map to an Interface or a VC, page 8 (required)
- Verifying the Configuration of Tunnel Marking for GRE Tunnels, page 10 (optional)

Configuring a Class Map

To configure a class map, perform the following task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** [**match-all** | **match-any**] *class-map-name*
4. **match fr-de**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map** [**match-all** | **match-any**] *class-map-name* **Example:** `Router(config)# class-map MATCH_FRDE`	Specifies the name of the class map to be created and enters class-map configuration mode. The class map defines the criteria to use to differentiate the traffic. For example, you can use the class map to differentiate voice traffic from data traffic, based on a series of match criteria defined using the **match** command. • Enter class map name. **Note** If the **match-all** or **match-any** keyword is not specified, traffic must match all the match criteria to be classified as part of the traffic class.
Step 4	**match fr-de** **Example:** `Router(config-cmap)# match fr-de`	Enables packet matching on the basis of the specified class. You can enter one of the following three **match** commands to define the match criteria for GRE tunnel marking: • match atm clp • match cos • match fr-de **Note** This is only an example of one match criterion that you can configure with a **match** command. Other criteria include matching on the IP precedence, access group, or protocol. Enter the **match** command for the criterion that you want to specify. For more information about specifying match criteria using the MQC, see the "Applying QoS Features Using the MQC" module.
Step 5	**end** **Example:** `Router(config-cmap)# end`	(Optional) Returns to privileged EXEC mode.

Creating a Policy Map

To create a policy map and configure it to set either the precedence value or the DSCP value in the header of a GRE-tunneled packet, perform the following tasks.

GRE Tunnel Marking and Traffic Policing

Traffic policing allows you to control the maximum rate of traffic sent or received on an interface and to partition a network into multiple priority levels or class of service (CoS). If you use traffic policing in your network, you can also implement the GRE tunnel marking feature with the **set-dscp-tunnel-transmit** or **set-prec-tunnel-transmit** actions (or keywords) of the **police** command in policy-map class configuration mode. Under traffic policing, tunnel marking can be applied with "conform" and "exceed" action statements, allowing you to apply a different value automatically for traffic that does not conform to the expected traffic rate.

GRE Tunnel Marking Values

The range of the tunnel marking values for the **set ip dscp tunnel** and **set-dscp-tunnel-transmit** commands is from 0 to 63; and the range of values for the **set ip precedence tunnel** and **set-prec-tunnel-transmit** commands is from 0 to 7.

Restrictions

It is possible to configure GRE tunnel marking and the **ip tos** command at the same time. However, MQC (GRE) tunnel marking has higher priority over IP ToS commands, meaning that tunnel marking will always rewrite the IP header of the tunnel packet, overwriting the values set by **ip tos** commands. The order of enforcement is as follows when these commands are used simultaneously:

1. **set ip dscp tunnel** or **set ip precedence tunnel** (GRE tunnel marking)
2. **ip tos reflect**
3. **ip tos** *tos-value*

Note This is the designed behavior. We recommend that you configure only GRE tunnel marking and reconfigure any peers, configured with the **ip tos** command, to use GRE tunnel marking.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** {*class-name* | **class-default**}
5. **set ip dscp tunnel** *dscp-value*

   or

   set ip precedence tunnel *precedence-value*

   or

   police *bps* [*burst-normal*] [*burst-max*] **conform-action** *action* **exceed-action** *action* [**violate-action** *action*]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map-name* **Example:** `Router(config)# policy-map TUNNEL_MARKING`	Creates or modifies a policy map that can be attached toone or more interfaces to specify a service policy, and enters policy-map configuration mode. • Enter the policy map name.
Step 4	**class** {*class-name* | **class-default**} **Example:** `Router(config-pmap)# class MATCH_FRDE`	Specifies the name of the class whose policy you want to create or change or specifies the default class (commonly known as the class-default class) before you configure its policy. Also enters policy-map class configuration mode. • Enter the class name, or enter the **class-default** keyword.
Step 5	**set ip dscp tunnel** *dscp-value* **Example:** `Router(config-pmap-c)# set ip dscp tunnel 3` or **set ip precedence tunnel** *precedence-value* **Example:** `Router(config-pmap-c)# set ip precedence tunnel 3` or	Sets or marks the differentiated services code point (DSCP) value in the tunnel header of a GRE-tunneled packet on the ingress interface. The tunnel marking value is a number from 0 to 63 when configuring DSCP. • Enter the tunnel value. Sets or marks the IP precedence value in the tunnel header of a GRE-tunneled packet on the ingress interface. The tunnel marking value is a number from 0 to 7 when configuring IP precedence. • Enter the tunnel value.

	Command or Action	Purpose
	police *bps* [*burst-normal*] [*burst-max*] **conform-action** *action* **exceed-action** *action* [**violate-action** *action*] **Example:** `Router(config-pmap-c)# police 8000 conform-action set-dscp-tunnel-transmit 4 exceed-action set-dscp-tunnel-transmit 0` or `Router(config-pmap-c)# police 8000 conform-action set-prec-tunnel-transmit 4 exceed-action set-prec-tunnel-transmit 0`	Configures traffic policing on the basis of the bits per second (bps) specified and the actions specified. If you use traffic policing in your network, you can implement the GRE tunnel marking feature with the **set-dscp-tunnel-transmit** or **set-prec-tunnel-transmit** keywords of the police command instead of the **set ip dscp tunnel** or the **set ip precedence tunnel** commands. The tunnel marking value for the traffic policing commands is from 0 to 63 when using **set-dscp-tunnel-transmit** and from 0 to 7 when using **set-prec-tunnel-transmit**. • Enter the bps, any optional burst sizes, and the desired conform and exceed actions. • Enter the **set-dscp-tunnel-transmit** or **set-prec-tunnel-transmit** commands after the **conform-action** keyword. **Note** This is an example of one QoS feature that you can configure at this step. Other QoS features include Weighted Random Early Detection (WRED), Weighted Fair Queueing (WFQ), and traffic shaping. Enter the command for the specific QoS feature that you want to configure. For more information about QoS features, see the "Quality of Service Overview" module.
Step 6	**end** **Example:** `Router(config-pmap-c)# end`	(Optional) Returns to privileged EXEC mode.

Attaching the Policy Map to an Interface or a VC

To attach the policy map to an interface or a virtual circuit (VC), perform the following task.

Restrictions

Policy maps can be attached to main interfaces, subinterfaces, or ATM permanent virtual circuits (PVCs). Policy maps are attached to interfaces by using the **service-policy** command and specifying either the **input** or **output** keyword to indicate the direction of the interface. This feature is supported only on ingress interfaces with the **input** keyword and should not be configured on egress interfaces with the **output** keyword.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]

4. **pvc** [*name*] *vpi/vci* [**ilmi** | **qsaal** | **smds**]
5. **service-policy** {**input** | **output**} *policy-map-name*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [*name-tag*] **Example:** `Router(config)# interface serial 0`	Configures the specified interface type and enters interface configuration mode. • Enter the interface type.
Step 4	**pvc** [*name*] *vpi/vci* [**ilmi**	**qsaal**	**smds**] **Example:** `Router(config-if)# pvc cisco 0/16 ilmi`	(Optional) Creates or assigns a name to an ATM PVC, specifies the encapsulation type on an ATM PVC, and enters ATM virtual circuit configuration mode.
Step 5	**service-policy** {**input**	**output**} *policy-map-name* **Example:** `Router(config-if)# service-policy input policy1` or **Example:** `Router(config-if-atm-vc)# service-policy input policy1`	Specifies the name of the policy map to be attached to the *input or output* direction of the interface. • Policy maps can be configured on ingress or egress routers. They can also be attached in the input or output direction of an interface. The direction (input or output) and the router (ingress or egress) to which the policy map should be attached vary according your network configuration. • Enter the **input** keyword followed by the policy map name. **Note** For this feature, only the incoming interface configured with the **input** keyword is supported.
Step 6	**end** **Example:** `Router(config-if)# end` or **Example:** `Router(config-if-atm-vc)# end`	(Optional) Returns to privileged EXEC mode.

Verifying the Configuration of Tunnel Marking for GRE Tunnels

To verify that the Tunnel Marking for GRE Tunnels feature is configured as intended and that either the IP precedence or DSCP value is set as expected, perform the following task.

SUMMARY STEPS

1. **enable**
2. **show policy-map interface** *interface-name*
3. **show policy-map** *policy-map*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`show policy-map interface` *interface-name* **Example:** `Router# show policy-map interface serial4/0`	(Optional) Displays the packet statistics of all classes that are configured for all service policies either on the specified interface or subinterface or on a specific PVC on the interface. • Enter the interface name.
Step 3	`show policy-map` *policy-map* **Example:** `Router# show policy-map policy1`	(Optional) Displays the configuration of all classes for a specified service policy map or all classes for all existing policy maps. • Enter a policy map name.
Step 4	`exit` **Example:** `Router# exit`	(Optional) Returns to user EXEC mode.

Troubleshooting Tips

The commands in the "Verifying the Configuration of Tunnel Marking for GRE Tunnels" section allow you to verify that you achieved the intended configuration and that the feature is functioning correctly. If, after using the **show** commands listed above, you find that the configuration is not functioning as expected, perform these operations to troubleshoot the configuration.

- Use the **show running-config** command and analyze the output of the command.
- If the policy map does not appear in the output of the **show running-config** command, enable the **logging console** command.
- Attach the policy map to the interface again.

Configuration Examples for QoS: Tunnel Marking for GRE Tunnels

This section provides the following configuration examples:

Configuring Tunnel Marking for GRE Tunnels: Examples

The following is an example of a GRE tunnel marking configuration. In this example, a class map called "MATCH_FRDE" has been configured to match traffic based on the Frame Relay DE bit.

```
Router> enable
Router# configure terminal
Router(config)# class-map MATCH_FRDE
Router(config-cmap)# match fr-de
Router(config-cmap)# end
```

In this part of the example configuration, a policy map called "TUNNEL_MARKING" has been created and the **set ip dscp tunnel** command has been configured in the policy map. You could use the **set ip precedence tunnel** command instead of the **set ip dscp tunnel** command if you do not use DSCP in your network.

```
Router(config)# policy-map TUNNEL_MARKING
Router(config-pmap)# class MATCH_FRDE
Router(config-pmap-c)# set ip dscp tunnel 3
Router(config-pmap-c)# end
```

Note This next part of the example configuration is not required to configure this feature if you use the **set ip dscp tunnel** or **set ip precedence tunnel** commands to enable GRE tunnel marking. This example shows how GRE tunnel marking can be enabled under traffic policing.

In this part of the example configuration, the policy map called "TUNNEL_MARKING" has been created and traffic policing has also been configured by using the **police** command and specifying the appropriate policing actions. The **set-dscp-tunnel-transmit** command can be used instead of the **set-prec-tunnel-transmit** command if you use DSCP in your network.

```
Router(config)# policy-map TUNNEL_MARKING
Router(config-pmap)# class class-default
Router(config-pmap-c)# police 8000 conform-action set-prec-tunnel-transmit 4 exceed-action
set-prec-tunnel-transmit 0
Router(config-pmap-c)# end
```

In the final part of the example configuration, the policy map is attached to serial interface 0 in the inbound (input) direction by specifying the **input** keyword of the **service-policy** command.

```
Router(config)# interface serial 0
Router(config-if)# service-policy input TUNNEL_MARKING
Router(config-if)# end
```

Verifying the Tunnel Marking for GRE Tunnels Configuration: Examples

This section contains sample output from the **show policy-map interface** command and the **show policy-map** command. The output from these commands can be used to verify and monitor the feature configuration in your network.

The following is sample output from the **show policy-map interface** command. In this sample output, the character string "ip dscp tunnel 3" indicates that GRE tunnel marking has been configured to set the DSCP value in the header of a GRE-tunneled packet.

```
Router# show policy-map interface

 Serial0

Service-policy input: tunnel

    Class-map: frde (match-all)
      0 packets, 0 bytes
      30 second offered rate 0 bps, drop rate 0 bps
      Match: fr-de
      QoS Set
        ip dscp tunnel 3
          Packets marked 0

    Class-map: class-default (match-any)
      13736 packets, 1714682 bytes
      30 second offered rate 0 bps, drop rate 0 bps
      Match: any
        13736 packets, 1714682 bytes
        30 second rate 0 bps
```

The following is sample output from the **show policy-map** command. In this sample output, the character string "ip precedence tunnel 4" indicates that the GRE tunnel marking feature has been configured to set the IP precedence value in the header of an GRE-tunneled packet.

```
Router# show policy-map

Policy Map TUNNEL_MARKING
    Class MATCH_FRDE
      set ip precedence tunnel 4
```

Additional References

The following sections provide references related to the QoS: Tunnel Marking for GRE Tunnels feature.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
MQC	"Applying QoS Features Using the MQC" module
Tunnel marking for Layer 2 Tunnel Protocol Version 3 (L2TPv3) tunnels	"QoS: Tunnel Marking for L2TPv3 Tunnels" module
DSCP	"Overview of DiffServ for Quality of Service" module

Standards

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

MIBs

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

RFCs

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

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Quality of Service Solutions Command Reference* at http://www.cisco.com/en/US/docs/ios/qos/command/reference/qos_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

- **match atm-clp**
- **match cos**
- **match fr-de**
- **police**
- **police (two rates)**
- **set ip dscp tunnel**
- **set ip precedence tunnel**
- **show policy-map**
- **show policy-map interface**

Feature Information for QoS: Tunnel Marking for GRE Tunnels

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for QoS: Tunnel Marking for GRE Tunnels

Feature Name	Releases	Feature Information
QoS: Tunnel Marking for GRE Tunnels	12.4(15)T2 12.2(33)SRC 12.2(33)SB	The QoS: Tunnel Marking for GRE Tunnels feature introduces the capability to define and control the QoS for incoming customer traffic on the PE router in a service provider network. **Note** For Cisco IOS Release 12.4(15)T2, the QoS: Tunnel Marking for GRE Tunnels feature is supported only on platforms equipped with a Cisco MGX Route Processor Module (RPM-XF). The following commands were introduced or modified: **match atm-clp**, **match cos**, **match fr-de**, **police**, **police (two rates)**, **set ip dscp tunnel**, **set ip precedence tunnel**, **show policy-map**, **show policy-map interface**.

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

Classifying Network Traffic

First Published: May 02, 2005
Last Updated: May 29, 2009

Classifying network traffic allows you to organize traffic (that is, packets) into traffic classes or categories on the basis of whether the traffic matches specific criteria. Classifying network traffic is the foundation for enabling many quality of service (QoS) features on your network. This module contains conceptual information and the configuration tasks for classifying network traffic.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "Feature Information for Classifying Network Traffic" section on page 15.

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

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Classifying Network Traffic

In order to mark network traffic, Cisco Express Forwarding (CEF) must be configured on both the interface receiving the traffic and the interface sending the traffic.

Information About Classifying Network Traffic

To classify network traffic, you should understand the following concepts:

Purpose of Classifying Network Traffic

Classifying network traffic allows you to organize traffic (that is, packets) into traffic classes or categories on the basis of whether the traffic matches specific criteria. Classifying network traffic is the foundation for enabling other QoS features such as traffic shaping and traffic policing on your network.

The goal of network traffic classification is to group traffic based on user-defined criteria so that the resulting groups of network traffic can then be subjected to specific QoS treatments. The QoS treatments might include faster forwarding by intermediate routers and switches or reduced probability of the traffic being dropped due to lack of buffering resources.

Identifying and categorizing network traffic into traffic classes (that is, classifying packets) enables distinct handling for different types of traffic, effectively separating network traffic into different categories. This classification can be associated with a variety of match criteria such as the IP Precedence value, differentiated services code point (DSCP) value, class of service (CoS) value, source and destination MAC addresses, input interface, or protocol type. You classify network traffic by using class maps and policy maps with the Modular Quality of Service Command-Line Interface (MQC). For example, you can configure class maps and policy maps to classify network traffic on the basis of the QoS group, Frame Relay DLCI number, Layer 2 packet length, or other criteria that you specify.

Benefits of Classifying Network Traffic

Classifying network traffic allows you to see what kinds of traffic you have, organize the various kinds of network traffic into traffic classes, and treat some types of traffic differently than others. Identifying and organizing network traffic is the foundation for applying the appropriate QoS feature to that traffic, enabling you to allocate network resources to deliver optimal performance for different types of traffic. For example, high-priority network traffic or traffic matching specific criteria can be singled out for special handling, and thus, help to achieve peak application performance.

MQC and Network Traffic Classification

To configure network traffic classification, you use the Modular Quality of Service Command-Line Interface (MQC).

The MQC is a CLI structure that allows you to complete the following tasks:

- Specify the matching criteria used to define a traffic class.
- Create a traffic policy (policy map). The traffic policy defines the QoS policy actions to be taken for each traffic class.
- Apply the policy actions specified in the policy map to aninterface, subinterface, or ATM permanent virtual circuit (PVC) by using the **service-policy** command.

For more information about the MQC, see the "Applying QoS Features Using the MQC" module.

Network Traffic Classification match Commands and Match Criteria

Network traffic classification allows you to group or categorize traffic on the basis of whether the traffic meets one or more specific criteria. For example, network traffic with a specific IP precedence can be placed into one traffic class, while traffic with a specific DSCP value can be placed into another traffic class. The network traffic within that traffic class can be given the appropriate QoS treatment, which you can configure in a policy map later.

You specify the criteria used to classify traffic with a **match** command. Table 1 lists the available **match** commands and the corresponding match criterion.

Table 1 match Commands and Corresponding Match Criterion

match Commands[1]	Match Criterion
match access group	Access control list (ACL) number
match any	Any match criteria
match class-map	Traffic class name
match cos	Layer 2 class of service (CoS) value
match destination-address mac	MAC address
match discard-class	Discard class value
match dscp	DSCP value
match field	Fields defined in the protocol header description files (PHDFs)
match fr-de	Frame Relay discard eligibility (DE) bit setting
match fr-dlci	Frame Relay data-link connection identifier (DLCI) number
match input-interface	Input interface name
match ip rtp	Real-Time Transport Protocol (RTP) port
match mpls experimental	Multiprotocol Label Switching (MPLS) experimental (EXP) value
match mpls experimental topmost	MPLS EXP value in the topmost label
match not	Single match criterion value to use as an unsuccessful match criterion
match packet length (class-map)	Layer 3 packet length in the IP header

Table 1 match Commands and Corresponding Match Criterion (continued)

match Commands[1]	Match Criterion
match port-type	Port type
match precedence	IP precedence values
match protocol	Protocol type
match protocol (NBAR)	Protocol type known to network-based application recognition (NBAR)
match protocol citrix	Citrix protocol
match protocol fasttrack	FastTrack peer-to-peer traffic
match protocol gnutella	Gnutella peer-to-peer traffic
match protocol http	Hypertext Transfer Protocol
match protocol rtp	RTP traffic
match qos-group	QoS group value
match source-address mac	Source Media Access Control (MAC) address
match start	Datagram header (Layer 2) or the network header (Layer 3)
match tag (class-map)	Tag type of class map
match vlan (QoS)	Layer 2 virtual local-area network (VLAN) identification number

1. Cisco IOS **match** commands can vary by release and platform. For instance, as of Cisco IOS Release 12.2(31)SB2, the **match vlan** (QoS) command is supported on Cisco 10000 series routers only. For more information, see the command documentation for the Cisco IOS release and platform that you are using.

Traffic Classification Compared with Traffic Marking

Traffic classification and traffic marking are closely related and can be used together. Traffic marking can be viewed as an additional action, specified in a policy map, to be taken on a traffic class.

Traffic classification allows you to organize into traffic classes on the basis of whether the traffic matches specific criteria. For example, all traffic with a CoS value of 2 is grouped into one class, and traffic with DSCP value of 3 is grouped into another class. The match criterion is user-defined.

After the traffic is organized into traffic classes, traffic marking allows you to mark (that is, set or change) an attribute for the traffic belonging to that specific class. For instance, you may want to change the CoS value from 2 to 1, or you may want to change the DSCP value from 3 to 2.

The match criteria used by traffic classification are specified by configuring a **match** command in a class map. The marking action taken by traffic marking is specified by configuring a **set** command in a policy map. These class maps and policy maps are configured using the MQC.

Table 2 compares the features of traffic classification and traffic marking.

Table 2 Traffic Classification Compared with Traffic Marking

	Traffic Classification	Traffic Marking
Goal	Groups network traffic into specific traffic classes on the basis of whether the traffic matches the user-defined criteria.	After the network traffic is grouped into traffic classes, modifies the attributes for the traffic in a particular traffic class.
Configuration Mechanism	Uses class maps and policy maps in the MQC.	Uses class maps and policy maps in the MQC.
CLI	In a class map, uses **match** commands (for example, **match cos**) to define the traffic matching criteria.	Uses the traffic classes and matching criteria specified by traffic classification. In addition, uses **set** commands (for example, **set cos**) in a policy map to modify the attributes for the network traffic. If a table map was created, uses the **table** keyword and *table-map-name* argument with the **set** commands (for example, **set cos precedence table** *table-map-name*) in the policy map to establish the to-from relationship for mapping attributes.

How to Classify Network Traffic

This section contains the following procedures:

- Creating a Class Map for Classifying Network Traffic, page 5 (required)
- Creating a Policy Map for Applying a QoS Feature to Network Traffic, page 6 (required)
- Attaching the Policy Map to an Interface, page 8 (required)
- Configuring QoS When Using IPsec VPNs, page 10 (optional)

Creating a Class Map for Classifying Network Traffic

In this procedure, you create a class map to define traffic classes. Within the class map, the appropriate **match** command is used to specify the matching criteria for the traffic classes.

To create the class map and specify the matching criteria, complete the following steps.

Note In the following task, the **match fr-dlci** command is shown in Step 4. The **match fr-dlci** command matches traffic on the basis of the Frame Relay DLCI number. The **match fr-dlci** command is just an example of one of the **match** commands that can be used. For a list of other **match** commands, see Table 1 on page 3.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** *class-map-name* [**match-all** | **match-any**]

4. **match fr-dlci** *dlci-number*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map** *class-map-name* [**match-all** | **match-any**] **Example:** `Router(config)# class-map class1`	Creates a class map to be used for matching traffic to a specified class, and enters class-map configuration mode. • Enter the class map name.
Step 4	**match fr-dlci** *dlci-number* **Example:** `Router(config-cmap)# match fr-dlci 500`	(Optional) Specifies the match criteria in a class map. **Note** The **match fr-dlci** command classifies traffic on the basis of the Frame RelayDLCI number. The **match fr-dlci** command is just an example of one of the **match** commands that can be used. For a list of other **match** commands, see Table 1 on page 3.
Step 5	**end** **Example:** `Router(config-cmap)# end`	(Optional) Returns to privileged EXEC mode.

Creating a Policy Map for Applying a QoS Feature to Network Traffic

In this procedure, you create and configure a policy map to use the class map. The policy map applies the appropriate QoS feature to the network traffic based on the traffic classification.

To create and configure a policy map, complete the following steps.

Note In the following task, the **bandwidth** command is shown at Step 5. The **bandwidth** command configures the QoS feature class-based weighted fair queuing (CBWFQ). CBWFQ is just an example of a QoS feature that can be configured. Use the appropriate command for the QoS feature you want to use.

Note Configuring bandwidth on policies that have the class-default class is supported on physical interfaces such as Gigabit Ethernet (GigE), Serial, Mobile Location Protocol (MLP), and Multilink Frame-Relay (MFR), but it is notsupported on logical interfaces such as Virtual Access Interface (VAI), Subinterface, and Frame-Relay on Virtual Circuits (FR-VC).

SUMMARY STEPS

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

   or

   show policy-map *policy-map* **class** *class-name*
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map-name* **Example:** `Router(config)# policy-map policy1`	Specifies the name of the policy map to be created and enters policy-map configuration mode. • Enter the policy map name.
Step 4	**class** {*class-name* | **class-default**} **Example:** `Router(config-pmap)# class class1`	Specifies the name of the class and enters policy-map class configuration mode. This class is associated with the class map created earlier. • Enter the name of the class or enter the **class-default** keyword.
Step 5	**bandwidth** {*bandwidth-kbps* | **remaining percent** *percentage* | **percent** *percentage*} **Example:** `Router(config-pmap-c)# bandwidth percent 50`	(Optional) Specifies or modifies the bandwidth allocated for a class belonging to a policy map. • Enter the amount of bandwidth as a number of kbps, a relative percentage of bandwidth, or an absolute amount of bandwidth. **Note** The **bandwidth** command configures the QoS feature class-based weighted fair queuing (CBWFQ). CBWFQ is just an example of a QoS feature that can be configured. Use the appropriate command for the QoS feature that you want to use.

	Command or Action	Purpose
Step 6	**end** **Example:** `Router(config-pmap-c)# end`	Returns to privileged EXEC mode.
Step 7	**show policy-map** or **show policy-map** *policy-map* **class** *class-name* **Example:** `Router# show policy-map` or **Example:** `Router# show policy-map policy1 class class1`	(Optional) Displays all configured policy maps. or (Optional) Displays the configuration for the specified class of the specified policy map. • Enter the policy map name and the class name.
Step 8	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

What to Do Next

Create and configure as many policy maps as you need for your network. To create and configure additional policy maps, repeat the steps in the "Creating a Policy Map for Applying a QoS Feature to Network Traffic" section on page 6. Then attach the policy maps to the appropriate interface, following the instructions in the "Attaching the Policy Map to an Interface" section on page 8.

Attaching the Policy Map to an Interface

After you create the policy map, you must attach it to an interface. Policy maps can be attached to either the input or output direction of the interface.

Note Depending on the needs of your network, policy maps can be attached to an interface, a subinterface, or an ATM PVC.

To attach the policy map, complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [**name-tag**]
4. **pvc** [*name*] *vpi/vci* [**ilmi** | **qsaal** | **smds** | **l2transport**]
5. **exit**

6. **service-policy** {**input** | **output**} *policy-map-name*
7. **end**
8. **show policy-map interface** *type number*
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [**name-tag**] **Example:** `Router(config)# interface serial4/0`	Configures an interface type and enters interface configuration mode. • Enter the interface type and number.
Step 4	**pvc** [*name*] *vpi/vci* [**ilmi** |**qsaal** |**smds** | **l2transport**] **Example:** `Router(config-if)# pvc cisco 0/16`	(Optional) Creates or assigns a name to an ATM PVC, specifies the encapsulation type on an ATM PVC, and enters ATM virtual circuit configuration mode. • Enter the PVC name, the ATM network virtual path identifier, and the network virtual channel identifier. **Note** This step is required only if you are attaching the policy map to an ATM PVC. If you are not attaching the policy map to an ATM PVC, advance to Step 6.
Step 5	**exit** **Example:** `Router(config-atm-vc)# exit`	(Optional) Returns to interface configuration mode. **Note** This step is required only if you are attaching the policy map to an ATM PVC and you completed Step 4. If you are not attaching the policy map to an ATM PVC, advance to Step 6.
Step 6	**service-policy** {**input** | **output**} *policy-map-name* **Example:** `Router(config-if)# service-policy input policy1`	Attaches a policy map to an input or output interface. • Enter the policy map name. **Note** Policy maps can be configured on ingress or egress routers. They can also be attached in the input or output direction of an interface. The direction (input or output) and the router (ingress or egress) to which the policy map should be attached varies according your network configuration. When using the **service-policy** command to attach the policy map to an interface, be sure to choose the router and the interface direction that are appropriate for your network configuration.

	Command or Action	Purpose
Step 7	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.
Step 8	**show policy-map interface** *type number* **Example:** `Router# show policy-map interface serial4/0`	(Optional) Displays the traffic statistics of all traffic classes that are configured for all service policies either on the specified interface or subinterface or on a specific PVC on the interface. • Enter the type and number.
Step 9	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

Configuring QoS When Using IPsec VPNs

Note This task is required only if you are using IPsec Virtual Private Networks (VPNs). Otherwise, this task is not necessary. For information about IPsec VPNs, see the "Configuring Security for VPNs with IPsec" module.

To configure QoS when using IPsec VPNs, complete the following steps.

Restrictions

This task uses the **qos pre-classify** command to enable QoS preclassification for the packet. QoS preclassification is not supported for all fragmented packets. If a packet is fragmented, each fragment might receive different preclassifications.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-num*
4. **exit**
5. **interface** *type number* [**name-tag**]
6. **qos pre-classify**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name seq-num* **Example:** `Router(config)# crypto map mymap 10`	Enters crypto map configuration mode and creates or modifies a crypto map entry. • Enter the crypto map name and sequence number.
Step 4	**exit** **Example:** `Router(config-crypto-map)# exit`	Returns to global configuration mode.
Step 5	**interface** *type number* [**name-tag**] **Example:** `Router(config)# interface serial4/0`	Configures an interface type and enters interface configuration mode. • Enter the interface type and number.
Step 6	**qos pre-classify** **Example:** `Router(config-if)# qos pre-classify`	Enables QoS preclassification.
Step 7	**end** **Example:** `Router(config-if)# end`	(Optional) Returns to privileged EXEC mode.

Configuration Examples for Classifying Network Traffic

This section contains the following examples:

Creating a Class Map for Classifying Network Traffic: Example

The following is an example of creating a class map to be used for traffic classification. In this example, a traffic class called class1 has been created. Traffic with a Frame Relay DLCI value of 500 will be put in this traffic class.

```
Router> enable
Router# configure terminal
Router(config)# class-map class1
Router(config-cmap)# match fr-dlci 500
Router(config-cmap)# end
```

Note This example uses the **match fr-dlci** command. The **match fr-dlci** command is just an example of one of the **match** commands that can be used. For a list of other **match** commands, see Table 1 on page 3.

Creating a Policy Map for Applying a QoS Feature to Network Traffic: Example

The following is an example of creating a policy map to be used for traffic classification. In this example, a policy map called policy1 has been created, and the **bandwidth** command has been configured for class1. The **bandwidth** command configures the QoS feature CBWFQ.

```
Router> enable
Router# configure terminal
Router(config)# policy-map policy1
Router(config-pmap)# class class1
Router(config-pmap-c)# bandwidth percent 50
Router(config-pmap-c)# end
Router# show policy-map policy1 class class1
Router# exit
```

Note This example uses the **bandwidth** command. The **bandwidth** command configures the QoS feature class-based weighted fair queuing (CBWFQ). CBWFQ is just an example of a QoS feature that can be configured. Use the appropriate command for the QoS feature that you want to use.

Attaching the Policy Map to an Interface: Example

The following is an example of attaching the policy map to an interface. In this example, the policy map called policy1 has been attached in the input direction of serial interface 4/0.

```
Router> enable
Router# configure terminal
Router(config)# interface serial4/0
Router(config-if)# service-policy input policy1
Router(config-if)# end
Router# show policy-map interface serial4/0
Router# exit
```

Configuring QoS When Using IPsec VPNs: Example

The following is an example of configuring QoS when using IPsec VPNs. In this example, the **crypto map** command specifies the IPsec crypto map mymap 10, to which the **qos pre-classify** command is applied.

```
Router> enable
Router# configure terminal
Router(config)# crypto map mymap 10
Router(config-crypto-map)# exit
Router(config)# interface serial4/0
Router(config-if)# qos pre-classify
Router(config-if)# end
```

Additional References

The following sections provide references related to classifying network traffic.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
MQC	"Applying QoS Features Using the MQC" module
Marking network traffic	"Marking Network Traffic" module
IPsec and VPNs	"Configuring Security for VPNs with IPsec" module
NBAR	"Classifying Network Traffic Using NBAR" module
CAR	"Configuring Committed Access Rate" module

Standards

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

MIBs

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

RFCs

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

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Classifying Network Traffic

Table 3 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(1) or a later release appear in the table.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 3 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 3 Feature Information for Classifying Network Traffic

Feature Name	Releases	Feature Information
Packet Classification Based on Layer 3 Packet Length	12.2(13)T	This feature provides the added capability of matching and classifying network traffic on the basis of the Layer3 length in the IP packet header. The Layer 3 length is the IP datagram plus the IP header. This new match criteria is in addition to the other match criteria, such as the IP precedence, differentiated services code point (DSCP) value, class of service (CoS), currently available. The following sections provide information about this feature: • Information About Classifying Network Traffic, page 2 • How to Classify Network Traffic, page 5
Packet Classification Using Frame Relay DLCI Number	12.2(13)T	The Packet Classification Using the Frame Relay DLCI Number feature allows customers to match and classify traffic based on the Frame Relay data-link connection identifier (DLCI) number associated with a packet. This new match criteria is in addition to the other match criteria, such as the IP Precedence, differentiated services code point (DSCP) value, class of service (CoS), currently available. The following sections provide information about this feature: • Information About Classifying Network Traffic, page 2 • How to Classify Network Traffic, page 5

Table 3 Feature Information for Classifying Network Traffic (continued)

Feature Name	Releases	Feature Information
Quality of Service for Virtual Private Networks	12.2(2)T	The QoS for VPNs 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 the data is encrypted and tunneled. The QoS for VPN feature allows users to look inside the packet so that packet classification can be done based on original port numbers and based on source and destination IP addresses. This allows the service provider to treat mission critical or multi-service traffic with higher priority across their network. The following sections provide information about this feature: • Configuring QoS When Using IPsec VPNs, page 10 • Configuring QoS When Using IPsec VPNs: Example, page 13
QoS: Match VLAN **Note** As of Cisco IOS Release 12.2(31)SB2, the QoS: Match VLAN feature is supported on Cisco 10000 series routers only.	12.2(31)SB2	The QoS: Match VLAN feature allows you to classify network traffic on the basis of the Layer 2 virtual local-area network (VLAN) identification number. The following sections provide information about this feature: • Information About Classifying Network Traffic, page 2 • How to Classify Network Traffic, page 5 The following commands were introduced or modified by this feature: **match vlan** (QoS), **show policy-map interface**.

Glossary

ATM—Asynchronous Transfer Mode. The international standard for cell relay in which multiple service types (such as voice, video, or data) are conveyed in fixed-length (53-byte) cells. Fixed-length cells allow cell processing to occur in hardware, thereby reducing transit delays. ATM is designed to take advantage of high-speed transmission media, such as E3, SONET, and T3.

CoS—class of service. An indication of how an upper-layer protocol requires a lower-layer protocol to treat its messages. A CoS definition comprises a virtual route number and a transmission priority field.

DLCI—data-link connection identifier. A value that specifies a PVC or a switched virtual circuit (SVC) in a Frame Relay network. In the basic Frame Relay specification, DLCIs are locally significant (connected devices might use different values to specify the same connection). In the Local Management Interface (LMI) extended specification, DLCIs are globally significant (DLCIs specify individual end devices).

IPsec—IP security. A framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPsec provides these security services at the IP layer. IPsec uses Internet Key Exchange (IKE) to handle the negotiation of protocols and algorithms based on local policy and to generate the encryption and authentication keys to be used by IPsec. IPsec can protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

MPLS—Multiprotocol Label Switching. A switching method that forwards IP traffic using a label. This label instructs the routers and the switches in the network where to forward the packets based on preestablished IP routing information.

PVC—permanent virtual circuit (or connection). A virtual circuit that is permanently established. PVCs save bandwidth associated with circuit establishment and teardown in situations where certain virtual circuits must exist all the time. In ATM terminology, called a permanent virtual connection.

VLAN—virtual LAN. A group of devices on one or more LANs that are configured (using management software) so that they can communicate as if they were attached to the same wire, when in fact they are located on a number of different LAN segments. Because VLANs are based on logical connections instead of physical connections, they are extremely flexible.

VPN—Virtual Private Network. A network that enables traffic to travel securely over a public or shared network. An IPsec VPN uses encryption and tunneling, encapsulating private IP packets into IPsec-encrypted packets to protect information at the IP level.

Classifying Network Traffic Using NBAR

Classifying Network Traffic Using NBAR Features Roadmap

First Published: April 4, 2006
Last Updated: December 5, 2008

This features roadmap lists the Cisco IOS features related to Network-Based Application Recognition (NBAR) that are documented in the *Cisco IOS Quality of Service Solutions Configuration Guide;* the roadmap also maps the features to the documents in which they appear. The roadmap is organized so that you can select your release train and see the features in that release. Find the feature name that you are searching for and click on the URL in the "Where Documented" column to access the document containing that feature.

Many legacy features have been incorporated into the configuration files, and these features may not have entries in this roadmap. In addition, information in this roadmap supports other software releases or platforms. For the latest feature information and caveats, see the release notes for your platform and software release.

Feature and Release Support

Table 1 lists NBAR-related feature support for the following Cisco IOS software release trains:

- Cisco IOS Releases 12.2T, 12.3, and 12.3T
- Cisco IOS Release 12.2ZY
- Cisco IOS Releases 12.4 and 12.4T
- Cisco IOS XE Release 2

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS, Catalyst OS, and Cisco IOS XE software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Table 1 lists the most recent release of each software train first and the features in alphabetical order within the release.

Table 1 Supported NBAR-Related Features

Release	Feature Name	Feature Description	Where Documented
Cisco IOS Releases 12.2T, 12.3, and 12.3T			
12.3(4)T	NBAR Extended Inspection for HTTP Traffic	Allows NBAR to scan TCP ports that are not well known and identify HTTP traffic that traverses these ports.	"Classifying Network Traffic Using NBAR" "Configuring NBAR Using the MQC"
	NBAR PDLM Versioning	Enables the ability to verify the Cisco IOS and NBAR Packet Description Language Module (PDLM) versions for ensuring software compatibility.	"Classifying Network Traffic Using NBAR" "Adding Application Recognition Modules"
	NBAR User-Defined Custom Application Classification	Provides the ability to identify TCP- or UDP-based applications by using a character string or value. The character string or value is used to match traffic within the packet payload.	"Classifying Network Traffic Using NBAR" "Creating a Custom Protocol"
12.2(15)T	NBAR Protocol Discovery MIBs	NBAR Protocol Discovery MIBs expand the capabilities of NBAR Protocol Discovery by providing the following new Protocol Discovery functionality through SNMP: • Enable or disable Protocol Discovery per interface. • Display Protocol Discovery statistics. • Configure and view multiple top-n tables that list protocols by bandwidth usage. • Configure thresholds based on traffic of particular NBAR-supported protocols or applications that report breaches and send notifications when these thresholds are crossed.	*Network-Based Application Recognition Protocol Discovery Management Information Base*
	NBAR Real-Time Transport Protocol Payload Classification	Enables stateful identification of real-time audio and video traffic.	"Classifying Network Traffic Using NBAR" "Configuring NBAR Using the MQC"

Table 1 Supported NBAR-Related Features (continued)

Release	Feature Name	Feature Description	Where Documented
Cisco IOS Release 12.2ZY			
12.2(18)ZYA1	Non-intrusive Protocol Discovery	Non-intrusive Protocol Discovery enables the Catalyst 6500 series switch that is equipped with a Supervisor 32/programmable intelligent services accelerator (PISA) toperform protocol discovery in out-of-band (that is, offline) mode. In offline mode, a copy of the network traffic is used to discover the application protocols that are operating on an interface.	"Classifying Network Traffic Using NBAR"
12.2(18)ZYA	NBAR—Network-Based Application Recognition	Integrates NBAR and Firewall Service Module (FWSM) functionality on the Catalyst 6500 series switch that is equipped with a Supervisor 32/programmable intelligent services accelerator (PISA). Provides support for Layer 2 Etherchannels and supports additional protocols.	"Classifying Network Traffic Using NBAR" "Enabling Protocol Discovery" "Configuring NBAR Using the MQC" The following commands were modified: **ip nbar protocol-discovery, match protocol (NBAR)**, **show ip nbar protocol-discovery**.
12.2(18)ZY2	NBAR—Network-Based Application Recognition (Hardware Accelerated Quality of Service (QoS) for NBAR Classification on PISA)	Enhances NBAR functionality on the Catalyst 6500 series switch that is equipped with a Supervisor 32/programmable intelligent services accelerator (PISA).	"Classifying Network Traffic Using NBAR"
12.2(18)ZY	NBAR—Network-Based Application Recognition (Hardware Accelerated NBAR)	Enables NBAR functionality on the Catalyst 6500 series switch that is equipped with a Supervisor 32/PISA.	"Classifying Network Traffic Using NBAR" "Configuring NBAR Using the MQC"

Table 1 Supported NBAR-Related Features (continued)

Release	Feature Name	Feature Description	Where Documented
Cisco IOS Releases 12.4 and 12.4T			
12.4(4)T	QoS: DirectConnect PDLM	Provides support for the DirectConnect PDLM and protocol. The DirectConnect protocol can now be recognized when using theModular Quality of Service (QoS) Command-Line Interface (CLI) (MQC) to classify traffic.	"Classifying Network Traffic Using NBAR" "Configuring NBAR Using the MQC" "Adding Application Recognition Modules"
	QoS: Skype Classification	Provides support for the Skype protocol. The Skype protocol can now be recognized when using the MQC to classify traffic. **Note** For Cisco IOS Release 12.4(4)T, Cisco supports only Skype version 1. Other versions of Skype are supported in later Cisco IOS releases.	"Classifying Network Traffic Using NBAR" "Configuring NBAR Using the MQC"
12.4(2)T	NBAR—BitTorrent PDLM	Provides support for the BitTorrent PDLM and protocol. The BitTorrent protocol can now be recognized when using the MQC to classify traffic.	"Classifying Network Traffic Using NBAR" "Configuring NBAR Using the MQC" "Adding Application Recognition Modules"
	NBAR—Citrix ICA Published Applications	Enables NBAR to classify traffic on the basis of the Citrix Independent Computing Architecture (ICA) published application name and tag number.	"Classifying Network Traffic Using NBAR" "Configuring NBAR Using the MQC"
	NBAR—Multiple Matches Per Port	Provides the ability for NBAR to distinguish between values of an attribute within the traffic stream of a particular application on a TCP or UDP port.	"Classifying Network Traffic Using NBAR" "Configuring NBAR Using the MQC" "Creating a Custom Protocol"
Cisco IOS XE Release 2			
Cisco IOS XE Release 2.1	NBAR—Network-Based Application Recognition	Provides support for the Protocol Discovery MIB and stateful identification of real-time audio and video traffic (Real-time Transport Protocol Payload Classification). Also provides support for additional protocols.	"Classifying Network Traffic Using NBAR" The following command was modified: **match protocol (NBAR)**.

Classifying Network Traffic Using NBAR

First Published: April 4, 2006
Last Updated: June 24, 2009

Network-Based Application Recognition (NBAR) is a classification engine that recognizes and classifies a wide variety of protocols and applications. When NBAR recognizes and classifies a protocol or application, the network can be configured to apply the appropriate quality of service (QoS) for that application or traffic with that protocol.

This module contains overview information about classifying network traffic using NBAR. The processes for configuring NBAR are documented in separate modules.

Note This module includes information for both NBAR and Distributed Network-Based Application Recognition (dNBAR). dNBAR is NBAR used on the Cisco 7500 router with a Versatile Interface Processor (VIP) and on the Catalyst 6500 family of switches with a FlexWAN module or serial interface processor (SIP). The implementation of NBAR and dNBAR is identical. Therefore, unless otherwise noted, the term NBAR is used throughout this module to describe both NBAR and dNBAR. The term dNBAR is used only when applicable.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Using NBAR

CEF

Before you configure NBAR, you must enable Cisco Express Forwarding (CEF). For more information on CEF, see the "CEF Feature Roadmap" module.

Note This prerequisite does not apply if you are using Cisco IOS Release 12.2(18)ZYA.

Stateful Switchover Support

NBAR is currently not supported with Stateful Switchover (SSO). This restriction applies to the Catalyst 6500 switches and to the Cisco 7500 and Cisco 7600 series routers.

Memory Requirements for dNBAR

To use dNBAR on a Cisco 7500 series router, you must be using a slot controller (or VIP processor) that has 64 MB of DRAM or more. Therefore, before configuring dNBAR on your Cisco 7500 series router, review the DRAM specifications for your particular slot controller or VIP processor.

Restrictions for Using NBAR

NBAR does not support the following:

- More than 24 concurrent URLs, hosts, or Multipurpose Internet Mail Extension (MIME) type matches.

  Note For Cisco IOS Release 12.2(18)ZYA, the maximum number of concurrent URLs, hosts, or MIME type matches is 56.

- Matching beyond the first 400 bytes in a packet payload in Cisco IOS releases before Cisco IOS Release 12.3(7)T. In Cisco IOS Release 12.3(7)T, this restriction was removed, and NBAR now supports full payload inspection. The only exception is that NBAR can inspect custom protocol traffic for only 255 bytes into the payload.
- Non-IP traffic.
- MPLS-labeled packets. NBAR classifies IP packets only. You can, however, use NBAR to classify IP traffic before the traffic is handed over to MPLS. Use the Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC) to set the IP differentiated services code point (DSCP) field on the NBAR-classified packets and make Multiprotocol Label Switching (MPLS) map the DSCP setting to the MPLS experimental (EXP) setting inside the MPLS header.
- Multicast and other non-CEF switching modes.
- Fragmented packets.
- Pipelined persistent HTTP requests.
- URL/host/MIME classification with secure HTTP.
- Asymmetric flows with stateful protocols.
- Packets that originate from or that are destined to the router running NBAR.

NBAR is not supported on the following logical interfaces:

- Fast Etherchannel

Note Fast Etherchannels *are* supported in Cisco IOS Release 12.2(18)ZYA.

- Dialer interfaces until Cisco IOS Release 12.2(4)T
- Interfaces where tunneling or encryption is used

Note You cannot use NBAR to classify output traffic on a WAN link where tunneling or encryption is used. Therefore, you should configure NBAR on other interfaces of the router (such as a LAN link) to perform input classification before the traffic is switched to the WAN link.

Layer 2 NBAR Restrictions

The phrase "Layer 2 NBAR" refers to NBAR functionality used with Layer 2 interfaces (such as switchports, trunks, or Etherchannels).

Layer 2 NBAR functionality can also be used with service modules such as a Firewall Service Module (FWSM) and an Intrusion Detection Service Module (IDSM) with the following restriction. Layer 2 NBAR is not supported on Layer 2 interfaces that are configured as part of a service module (such as FWSM and IDSM) when those service modules are configured in inline mode (that is, network traffic is in a direct path through the service module).

Note This restriction does not apply to NBAR functionality that is used with Layer 3 interfaces.

However, Layer 2 NBAR *is* supported in non-inline mode with service modules even when using Switched Port Analyzer (SPAN), Remote SPAN (RSPAN), or VLAN Access Control List (VACL) Capture functionality to send traffic to a service module.

For more information about the FWSM and its connection features, see the "Configuring Advanced Connection Features" module of the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Configuration Guide.*

For more information about the IDSM, see the "Configuring IDSM-2" module of the *Configuring the Cisco Intrusion Prevention System Sensor Using the Command Line Interface.*

For more information about SPAN or RSPAN, see the "Configuring SPAN and RSPAN" module of the *Catalyst 6500 Series Software Configuration Guide*.

For more information about VACL Capture, see the "VACL Capture for Granular Traffic Analysis with Cisco Catalyst 6000/6500 Running Cisco IOS Software" module.

Information About Using NBAR

Before classifying network traffic using NBAR, you should understand the following concepts:

- NBAR Functionality, page 4
- NBAR Benefits, page 5
- NBAR and Classification of HTTP Traffic, page 6

NBAR Functionality

NBAR is a classification engine that recognizes and classifies a wide variety of protocols and applications, including web-based and other difficult-to-classify applications and protocols that use dynamic TCP/UDP port assignments.

When NBAR recognizes and classifies a protocol or application, the network can be configured to apply the appropriate QoS for that application or traffic with that protocol. The QoS is applied using the Modular Quality of Service Command-Line Interface (MQC).

Note For more information about NBAR and its relationship with the MQC, see the "Configuring NBAR Using the MQC" module.

Examples of the QoS features that can be applied to the network traffic (using the MQC) after NBAR has recognized and classified the application or protocol include the following:

- Class-Based Marking
- Class-Based Weighted Fair Queuing (CBWFQ)
- Low Latency Queuing (LLQ)
- Traffic Policing
- Traffic Shaping

Note For Cisco IOS Release 12.2(18)ZYA on the Catalyst 6500 series switch (that is equipped with a Supervisor 32/programmable intelligent services accelerator [PISA]), only the QoS features listed below can be configured. These features can be configured (using the MQC) after NBAR has recognized and classified the application or protocol.

- Traffic Classification
- Traffic Marking
- Traffic Policing

Note For more information about the QoS features, see the "Quality of Service Overview" module. For more information about the Catalyst 6500 series switch and QoS, see the "Configuring QoS" module of the *Catalyst 6500 Series Software Configuration Guide.*

NBAR introduces several classification features that identify applications and protocols from Layer 4 through Layer 7. These classification features include the following:

- Statically assigned TCP and UDP port numbers.
- Non-TCP and non-UDP IP protocols.
- Dynamically assigned TCP and UDP port numbers.

  This kind of classification requires stateful inspection; that is, the ability to inspect a protocol across multiple packets during packet classification.
- Subport classification or classification based on deep-packet inspection.

  Deep-packet classification is classification performed at a finer level of granularity. For instance, if a packet is already classified as HTTP traffic, it may be further classified by HTTP traffic with a specific URL.

Note Access control lists (ACLs) can also be used for classifying static port protocols. However, NBAR is easier to configure, and NBAR can provide classification statistics that are not available when ACLs are used.

NBAR includes a Protocol Discovery feature that provides an easy way to discover application protocols that are operating on an interface. For more information about Protocol Discovery, see the "Enabling Protocol Discovery" module.

Note NBAR classifies network traffic by application or protocol. Network traffic can be classified without using NBAR. For information about classifying network traffic without using NBAR, see the "Classifying Network Traffic" module.

NBAR Benefits

Improved Network Management

Identifying and classifying network traffic is an important first step in implementing QoS. A network administrator can more effectively implement QoS in a networking environment after identifying the amount and the variety of applications and protocols that are running on a network.

NBAR gives network administrators the ability to see the variety of protocols and the amount of traffic generated by each protocol. After gathering this information, NBAR allows users to organize traffic into classes. These classes can then be used to provide different levels of service for network traffic, thereby allowing better network management by providing the right level of network resources for network traffic.

NBAR and Classification of HTTP Traffic

This section includes information about the following topics:

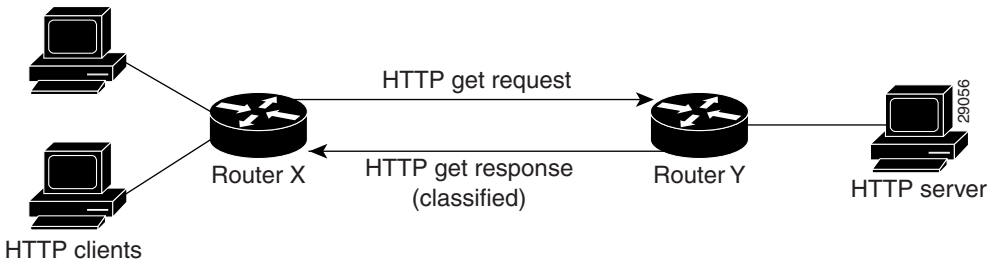
Classification of HTTP Traffic by URL, Host, or MIME

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

Classification of HTTP traffic by URL, host, or Multipurpose Internet Mail Extension (MIME) type is an example of subport classification. NBAR classifies HTTP traffic by text within the URL or host fields of a request using regular expression matching. HTTP client request matching in NBAR supports most HTTP request methods such as GET, PUT, HEAD, POST, DELETE, OPTIONS, CONNECT, and TRACE. The NBAR engine then converts the specified match string into a regular expression.

Figure 1 illustrates a network topology with NBAR in which Router Y is the NBAR-enabled router.

Figure 1 Network Topology with NBAR



When specifying a URL for classification, include only the portion of the URL that follows the www.*hostname.domain* in the **match** statement. For example, for the URL www.cisco.com/latest/whatsnew.html, include only /latest/whatsnew.html with the **match** statement (for instance, **match protocol http url /latest/whatsnew.html**).

Note For Cisco IOS Release 12.2(18)ZY2 (and later) on the Cisco Catalyst 6500 series switch that is equipped with a Supervisor 32/PISA, up to 56 parameters or sub classifications can be specified with the **match protocol http** command. These parameters or sub classifications can be a combination of any of the available match choices, such as host matches, MIME matches, server matches, and URL matches. For other Cisco IOS releases and platforms, the maximum is 24 parameters or sub classifications.

Host specification is identical to URL specification. NBAR performs a regular expression match on the host field contents inside an HTTP packet and classifies all packets from that host. For example, for the URL www.cisco.com/latest/whatsnew.html, include only www.cisco.com.

For MIME type matching, the MIME type can contain any user-specified text string. A list of the IANA-supported MIME types can be found at the following URL:

http://www.iana.org/assignments/media-types/

When matching by MIME type, NBAR matches a packet containing the MIME type and all subsequent packets until the next HTTP transaction.

NBAR supports URL and host classification in the presence of persistent HTTP. NBAR does not classify packets that are part of a pipelined request. With pipelined requests, multiple requests are pipelined to the server before previous requests are serviced. Pipelined requests are a less commonly used type of persistent HTTP request.

In Cisco IOS Release 12.3(4)T, the NBAR Extended Inspection for HTTP Traffic feature was introduced. This feature allows NBAR to scan TCP ports that are not well known and to identify HTTP traffic that traverses these ports. HTTP traffic classification is no longer limited to the well-known and defined TCP ports.

Classification of HTTP Traffic Using the HTTP Header Fields

In Cisco IOS Release 12.3(11)T, NBAR introduced expanded ability for users to classify HTTP traffic using information in the HTTP header fields.

HTTP works using a client/server model. HTTP clients open connections by sending a request message to an HTTP server. The HTTP server then returns a response message to the HTTP client (this response message is typically the resource requested in the request message from the HTTP client). After delivering the response, the HTTP server closes the connection and the transaction is complete.

HTTP header fields are used to provide information about HTTP request and response messages. HTTP has numerous header fields. For additional information on HTTP headers, see section 14 of RFC 2616: *Hypertext Transfer Protocol—HTTP/1.1*. This RFC can be found at the following URL:

http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html

NBAR is able to classify the following HTTP header fields:

- For request messages (client to server), the following HTTP header fields can be identified using NBAR:
  - User-Agent
  - Referer
  - From
- For response messages (server to client), the following HTTP header fields can be identified using NBAR:
  - Server
  - Location
  - Content-Encoding
  - Content-Base

Note Use of the Content-Base field has not been implemented by the HTTP community. (See RFC 2616 for details.) Therefore, the Content-Base field is not identified by NBAR on the Catalyst 6500 series switch that is equipped with a Supervisor 32/PISA.

Within NBAR, the **match protocol http c-header-field** command is used to specify that NBAR identify request messages (the “c” in the **c-header-field** portion of the command is for client). The **match protocol http s-header-field** command is used to specify response messages (the “s” in the **s-header-field** portion of the command is for server).

Note For Cisco IOS Release 12.2(18)ZY2 (and later) on the Cisco Catalyst 6500 series switch that is equipped with a Supervisor 32/PISA, the **c-header-field** and **s-header-field** keywords and associated arguments are not available. The same functionality is achieved by using the individual keywords and arguments. For more information, see the syntax of the **match protocol http** command in the *Cisco IOS Quality of Service Solutions Command Reference*.

Examples

In the following example, any request message that contains "somebody@cisco.com" in the User-Agent, Referer, or From fields will be classified by NBAR. Typically, a term with a format similar to "somebody@cisco.com" would be found in the From header field of the HTTP request message.

```
class-map match-all class1
 match protocol http c-header-field "somebody@cisco.com"
```

In the following example, any request message that contains "http://www.cisco.com/routers" in the User-Agent, Referer, or From fields will be classified by NBAR. Typically, a term with a format similar to "http://www.cisco.com/routers" would be found in the Referer header field of the HTTP request message.

```
class-map match-all class2
 match protocol http c-header-field "http://www.cisco.com/routers"
```

In the following example, any request message that contains "CERN-LineMode/2.15" in the User-Agent, Referer, or From header fields will be classified by NBAR. Typically, a term with a format similar to "CERN-LineMode/2.15" would be found in the User-Agent header field of the HTTP request message.

```
class-map match-all class3
 match protocol http c-header-field "CERN-LineMode/2.15"
```

In the following example, any response message that contains "CERN/3.0" in the Content-Base (if available), Content-Encoding, Location, or Server header fields will be classified by NBAR. Typically, a term with a format similar to "CERN/3.0" would be found in the Server header field of the response message.

```
class-map match-all class4
 match protocol http s-header-field "CERN/3.0"
```

In the following example, any response message that contains "http://www.cisco.com/routers" in the Content-Base (if available), Content-Encoding, Location, or Server header fields will be classified by NBAR. Typically, a term with a format similar to "http://www.cisco.com/routers" would be found in the Content-Base (if available) or Location header field of the response message.

```
class-map match-all class5
 match protocol http s-header-field "http://www.cisco.com/routers"
```

In the following example, any response message that contains "gzip" in the Content-Base (if available), Content-Encoding, Location, or Server header fields will be classified by NBAR. Typically, the term "gzip" would be found in the Content-Encoding header field of the response message.

```
class-map match-all class6
 match protocol http s-header-field "gzip"
```

Combinations of Classification of HTTP Headers and URL, Host, or MIME Type to Identify HTTP Traffic

Note that combinations of URL, Host, MIME type, and HTTP headers can be used during NBAR configuration. These combinations provide customers with more flexibility to classify specific HTTP traffic based on their network requirements.

Examples

In the following example, HTTP header fields are combined with a URL to classify traffic. In this example, traffic with a User-Agent field of "CERN-LineMode/3.0" and a Server field of "CERN/3.0," along with URL "www.cisco.com/routers," will be classified using NBAR:

```
class-map match-all c-http
 match protocol http c-header-field "CERN-LineMode/3.0"
 match protocol http s-header-field "CERN/3.0"
 match protocol http url "www.cisco.com/routers"
```

NBAR and Classification of Citrix ICA Traffic

NBAR can classify Citrix Independent Computing Architecture (ICA) traffic and perform subport classification of Citrix traffic based on the published application name or ICA tag number.

This section includes information about the following topics:

- Classification of Citrix ICA Traffic by Published Application Name, page 9
- Classification of Citrix ICA Traffic by ICA Tag Number, page 10

Classification of Citrix ICA Traffic by Published Application Name

NBAR can monitor Citrix ICA client requests for a published application destined to a Citrix ICA Master browser. After the client requests the published application, the Citrix ICA Master browser directs the client to the server with the most available memory. The Citrix ICA client then connects to this Citrix ICA server for the application.

Note For Citrix to monitor and classify traffic by the published application name, Server Browser Mode on the Master browser must be used.

In Server Browser Mode, NBAR statefully tracks and monitors traffic and performs a regular expression search on the packet contents for the published application name specified by the **match protocol citrix** command. The published application name is specified by using the **app** keyword and the *application-name-string* argument of the **match protocol citrix** command. For more information about the **match protocol citrix** command, see the *Cisco IOS Quality of Service Solutions Command Reference*.

The Citrix ICA session triggered to carry the specified application is cached, and traffic is classified appropriately for the published application name.

Citrix ICA Client Modes

Citrix ICA clients can be configured in various modes. NBAR cannot distinguish among Citrix applications in all modes of operation. Therefore, network administrators might need to collaborate with Citrix administrators to ensure that NBAR properly classifies Citrix traffic.

A Citrix administrator can configure Citrix to publish Citrix applications individually or as the entire desktop. In the Published Desktop mode of operation, all applications within the published desktop of a client use the same TCP session. Therefore, differentiation among applications is impossible, and NBAR can be used to classify Citrix applications only as aggregates (by looking at port 1494).

The Published Application mode for Citrix ICA clients is recommended when you use NBAR. In Published Application mode, a Citrix administrator can configure a Citrix client in either seamless or non-seamless (windows) modes of operation. In nonseamless mode, each Citrix application uses a separate TCP connection, and NBAR can be used to provide interapplication differentiation based on the name of the published application.

Seamless mode clients can operate in one of two submodes: session sharing or nonsession sharing. In seamless session sharing mode, all clients share the same TCP connection, and NBAR cannot differentiate among applications. Seamless sharing mode is enabled by default on some software releases. In seamless nonsession sharing mode, each application for each particular client uses a separate TCP connection. NBAR can provide interapplication differentiation in seamless nonsession sharing mode.

Note NBAR operates properly in Citrix ICA secure mode. Pipelined Citrix ICA client requests are not supported.

Classification of Citrix ICA Traffic by ICA Tag Number

Citrix uses one TCP session each time an application is opened. In the TCP session, a variety of Citrix traffic may be intermingled in the same session. For example, print traffic may be intermingled with interactive traffic, causing interruption and delay for a particular application. Most people would prefer that printing be handled as a background process and that printing not interfere with the processing of higher-priority traffic.

To accommodate this preference, the Citrix ICA protocol includes the ability to identify Citrix ICA traffic based on the ICA tag number of the packet. The ability to identify, tag, and prioritize Citrix ICA traffic is referred to as ICA Priority Packet Tagging. With ICA Priority Packet Tagging, Citrix ICA traffic is categorized as high, medium, low, and background, depending on the ICA tag of the packet.

When ICA traffic priority tag numbers are used, and the priority of the traffic is determined, QoS features can be implemented to determine how the traffic will be handled. For example, QoS traffic policing can be configured to transmit or drop packets with a specific priority.

Citrix ICA Packet Tagging

The Citrix ICA tag is included in the first two bytes of the Citrix ICA packet, after the initial negotiations are completed between Citrix client and server. These bytes are not compressed or encrypted.

The first two bytes of the packet (byte 1 and byte 2) contain the byte count and the ICA priority tag number. Byte 1 contains the low-order byte count, and the first two bits of byte 2 contain the priority tags. The other six bits contain the high-order byte count.

The ICA priority tag value can be a number from 0 to 3. The number indicates the packet priority, with 0 being the highest priority and 3 being the lowest priority.

To prioritize Citrix traffic by the ICA tag number of the packet, you specify the tag number using the **ica-tag** keyword and the *ica-tag-value* argument of the **match protocol citrix** command. For more information about the **match protocol citrix** command, see the *Cisco IOS Quality of Service Solutions Command Reference*.

NBAR and RTP Payload Type Classification

RTP is a packet format for multimedia data streams. It can be used for media-on-demand as well as for interactive services such as Internet telephony. RTP consists of a data and a control part. The control part is called Real-Time Transport Control Protocol (RTCP). RTCP is a separate protocol that is supported by NBAR. It is important to note that the NBAR RTP Payload Type 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 that provides 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. RTP is discussed in RFC 1889 (*A Transport Protocol for Real-Time Applications*) and RFC 1890 (*RTP Profile for Audio and Video Conferences with Minimal Control*).

The RTP payload type is the data transported by RTP in a packet, for example audio samples or compressed video data.

NBAR RTP Payload Type Classification not only allows one to statefully identify real-time audio and video traffic but can also differentiate on the basis of audio and video codecs to provide more granular QoS. The RTP Payload Type Classification feature, therefore, looks deep into the RTP header to classify RTP packets.

NBAR and Classification of Custom Protocols and Applications

NBAR supports the use of custom protocols to identify custom applications. Custom protocols support static port-based protocols and applications that NBAR does not currently support. You can add to the set of protocols and application types that NBAR recognizes by creating custom protocols.

Custom protocols extend the capability of NBAR Protocol Discovery to classify and monitor additional static port applications and allows NBAR to classify nonsupported static port traffic.

Note For more information about specifying user-defined (custom) protocols, see the "Creating a Custom Protocol" module.

NBAR and Classification of Peer-to-Peer File-Sharing Applications

The following are the most common peer-to-peer file-sharing applications supported by NBAR:

- BitTorrent
- DirectConnect
- eDonkey
- eMule
- FastTrack
- Grokster
- JTella
- Kazaa (as well as Kazaa Lite and Kazaa Lite Resurrection)
- Morpheus
- Win MX

Gnutella Also Supported

Gnutella is another file-sharing protocol that became classifiable using NBAR in Cisco IOS Release 12.1(12c)E.

Applications that use the Gnutella protocol include Bearshare, Gnewtellium, Gnucleus, Gtk-Gnutella, Limewire, Mutella, Phex, Qtella, Swapper, and Xolo.

The **match protocol gnutella file-transfer** *regular-expression* and **match protocol fasttrack file-transfer** *regular-expression* commands are used to enable Gnutella and FastTrack classification in a traffic class. The **file-transfer** keyword indicates that a regular expression variable will be used to identify specific Gnutella or FastTrack traffic. The *regular-expression* variable can be expressed as "*" to indicate that all FastTrack or Gnutella traffic be classified by a traffic class.

In the following example, all FastTrack traffic is classified into class map nbar:

```
class-map match-all nbar
 match protocol fasttrack file-transfer "*"
```

Similarly, all Gnutella traffic is classified into class map nbar in the following example:

```
class-map match-all nbar
 match protocol gnutella file-transfer "*"
```

Wildcard characters in a regular expression can also be used to identify specified Gnutella and FastTrack traffic. These regular expression matches can be used to match on the basis of filename extension or a particular string in a filename.

In the following example, all Gnutella files that have the .mpeg extension will be classified into class map nbar.

```
class-map match-all nbar
 match protocol gnutella file-transfer "*.mpeg"
```

In the following example, only Gnutella traffic that contains the characters "cisco" is classified:

```
class-map match-all nbar
 match protocol gnutella file-transfer "*cisco*"
```

The same examples can be used for FastTrack traffic:

```
class-map match-all nbar
 match protocol fasttrack file-transfer "*.mpeg"
```

or

```
class-map match-all nbar
 match protocol fasttrack file-transfer "*cisco*"
```

NBAR and Classification of Streaming Protocols

In Cisco IOS Release 12.3(7)T, NBAR introduced support for Real Time Streaming Protocol (RTSP). RTSP is the protocol used for applications with steaming audio, such as the following:

- Apple QuickTime
- RealAudio (RealSystems G2)
- Windows Media Services

NBAR and AutoQoS

Earlier Cisco IOS releases included two features that allow you to automate the deployment of QoS on your network: AutoQoS—Voice over IP (VoIP); and AutoQoS for the Enterprise. Both of these AutoQoS features take advantage of the traffic classification functionality of NBAR.

Note Cisco IOS Release 12.2(18)ZY (and later) does not support the AutoQoS—Voice over IP (VoIP) feature on the Catalyst 6500 series switch.

AutoQoS—VoIP

This feature was available with Cisco IOS Release 12.2(15)T. The AutoQoS—VoIP feature allows you to automate the deployment of QoS on your network and provides a means for simplifying the implementation and provisioning of QoS for VoIP traffic. For more information about the AutoQoS—VoIP feature and how it uses NBAR, see the "AutoQoS—VoIP" module.

AutoQoS for the Enterprise

This feature was available with Cisco IOS Release 12.3(11)T. The AutoQoS for the Enterprise feature allows you to automate the deployment of QoS in a general business environment, particularly for midsize companies and branch offices of larger companies. It expands on the functionality available with the AutoQoS—VoIP feature. For more information about the AutoQoS for the Enterprise feature and how it uses NBAR, see the "AutoQoS for the Enterprise" module.

NBAR and FWSM Integration

With Cisco IOS Release 12.2(18)ZYA, the functionality of NBAR to recognize protocols and applications has been integrated with the Firewall Service Module (FWSM) on the Catalyst 6500 series switch. Available with this release are the following commands that can be used for classifying and tagging traffic to the FWSM:

- **ip nbar protocol-tagging**
- **show ip nbar protocol-tagging**

For more information about the FWSM and its connection features, see the "Configuring Advanced Connection Features" module of the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Configuration Guide.*

For more information about FWSM commands (including the two commands listed above), see the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Service Module Command Reference Guide.*

NBAR and TelePresence PDLM

Cisco IOS Release 12.2(18)ZYA2 NBAR introduced support for the Cisco TelePresence PDLM.

Cisco TelePresence integrates advanced audio, high-definition video and interactive elements with the power of the underlying network to deliver an immersive meeting experience.

The Telepresence PDLM uses NBAR to identify TelePresence media and TelePresence control traffic over the network. TelePresence media traffic and TelePresence control traffic are treated differently by QoS and so must be classified separately. TelePresence media traffic must have a low latency. TelePresence control traffic does not require a low latency but should not be dropped.

NBAR-Supported Protocols

The **match protocol** (NBAR) command is used to classify traffic on the basis of protocols supported by NBAR. NBAR is capable of classifying the following types of protocols:

- Non-UDP and non-TCP IP protocols
- TCP and UDP protocols that use statically assigned port numbers
- TCP and UDP protocols that use statically assigned port numbers but still require stateful inspection
- TCP and UDP protocols that dynamically assign port numbers and therefore require stateful inspection

Table 1 lists the NBAR-supported protocols available in Cisco IOS software, sorted by category. The table also provides information about the protocol type, the well-known port numbers (if applicable), the syntax for entering the protocol in NBAR, and the Cisco IOS release in which the protocol was initially supported. This table is updated when a protocol is added to a new Cisco IOS release train.

Many peer-to-peer file-sharing applications not listed in this table can be classified using FastTrack or Gnutella. See the "NBAR and Classification of Peer-to-Peer File-Sharing Applications" section on page 11 for additional information.

RTSP can be used to classify various types of applications that use streaming audio. See the "NBAR and Classification of Streaming Protocols" section on page 12 for additional information.

Note Support for some protocols can be added to NBAR using application recognition modules (also known as Packet Description Language Modules [PDLMs]). For more information about PDLMs, see the "Adding Application Recognition Modules" module.

Note Table 1 includes the NBAR-supported protocols available with the 12.2(18)ZY and 12.2(18)ZYA releases. Protocols included in the 12.2(18)ZY and 12.2(18)ZYA releases are supported on the Catalyst 6500 series switch that is equipped with a Supervisor 32/PISA.

Table 1 NBAR-Supported Protocols

Category	Protocol	Type	Well-Known Port Number	Description	Syntax	Cisco IOS Release
Enterprise Application	Citrix ICA	TCP/ UDP	TCP: 1494, 2512, 2513, 2598 UDP: 1604	Citrix ICA traffic	citrix citrix app citrix ica-tag	12.1(2)E 12.1(5)T 12.2(18)ZYA1
	PCAnywhere	TCP/ UDP	TCP: 5631, 65301 UDP: 22, 5632	Symantic PCAnywhere	pcanywhere	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	Novadigm	TCP/ UDP	3460–3465	Novadigm Enterprise Desktop Manager (EDM)	novadigm	12.1(2)E 12.1(5)T 12.2(18)ZYA1
	SAP	TCP	3300–3315 (sap-pgm. pdlm) 3200–3215 (sap-app. pdlm) 3600–3615 (sap-msg. pdlm)	Application server to application server traffic (sap-pgm.pdlm) Client to application server traffic (sap-app.pdlm) Client to message server traffic (sap-msg.pdlm)	sap	12.1E 12.2T 12.3 12.3T 12.2(18)ZYA1
	Exchange[1]	TCP	135	MS-RPC for Exchange	exchange	12.1(1)E 12.1(5)T 12.2(18)ZY 12.2(18)ZYA 12.2(18)ZYA1
	MAPI	TCP	135	Messaging Application Programming Interface	mapi	12.2(18)ZYA 12.2(18)ZYA1

Table 1 NBAR-Supported Protocols (continued)

Category	Protocol	Type	Well-Known Port Number	Description	Syntax	Cisco IOS Release
Routing Protocol	BGP	TCP/ UDP	179	Border Gateway Protocol	bgp	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	EGP	IP	8	Exterior Gateway Protocol	egp	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	EIGRP	IP	88	Enhanced Interior Gateway Routing Protocol	eigrp	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	OSPF	IP	89	Open Shortest Path First	ospf	12.3(8)T 12.2(18)ZYA1
	RIP	UDP	520	Routing Information Protocol	rip	12.1(1)E 12.1(5)T 12.2(18)ZYA1
Database	SQL*NET	TCP/ UDP	1521	SQL*NET for Oracle	sqlnet	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	MS- SQLServer	TCP	1433	Microsoft SQL Server Desktop Videoconferencing	sqlserver	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	CIFS	TCP	139, 445	Common Internet File System	cifs	12.2(18)ZYA 12.2(18)ZYA1
Health	DiCom	TCP	Dynamically Assigned	Digital Imaging and Communications in Medicine	dicom	12.2(18)ZYA 12.2(18)ZYA1
	HL7	TCP	Dynamically Assigned	Health Level Seven	hl7	12.2(18)ZYA 12.2(18)ZYA1
Financial	FIX	TCP	Dynamically Assigned	Financial Information Exchange	fix	12.2(18)ZYA 12.2(18)ZYA1

Table 1 NBAR-Supported Protocols (continued)

Category	Protocol	Type	Well-Known Port Number	Description	Syntax	Cisco IOS Release
Security and Tunneling	GRE	IP	47	Generic Routing Encapsulation	gre	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	IPINIP	IP	4	IP in IP	ipinip	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	IPsec	IP	50, 51	IP Encapsulating Security Payload/ Authentication-Header	ipsec	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	L2TP	UDP	1701	L2F/L2TP Tunnel	l2tp	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	MS-PPTP	TCP	1723	Microsoft Point-to-Point Tunneling Protocol for VPN	pptp	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	SFTP	TCP	990	Secure FTP	secure-ftp	12.1(1)E 12.1(5)T 12.2(18)ZYA1

Table 1 NBAR-Supported Protocols (continued)

Category	Protocol	Type	Well-Known Port Number	Description	Syntax	Cisco IOS Release
Security and Tunneling (Continued)	SHTTP	TCP	443	Secure HTTP	secure-http	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	SIMAP	TCP/ UDP	585, 993	Secure IMAP	secure-imap	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	SIRC	TCP/ UDP	994	Secure IRC	secure-irc	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	SLDAP	TCP/ UDP	636	Secure LDAP	secure-ldap	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	SNNTP	TCP/ UDP	563	Secure NNTP	secure-nntp	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	SPOP3	TCP/ UDP	995	Secure POP3	secure-pop3	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	STELNET	TCP	992	Secure Telnet	secure-telnet	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	SOCKS	TCP	1080	Firewall Security Protocol	socks	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	SSH	TCP	22	Secured Shell	ssh	12.1(1)E 12.1(5)T 12.2(18)ZYA1
Network Management	ICMP	IP	1	Internet Control Message Protocol	icmp	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	SNMP	TCP/ UDP	161, 162	Simple Network Management Protocol	snmp	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	Syslog	UDP	514	System Logging Utility	syslog	12.1(1)E 12.1(5)T 12.2(18)ZYA1

Table 1 NBAR-Supported Protocols (continued)

Category	Protocol	Type	Well-Known Port Number	Description	Syntax	Cisco IOS Release
Network Mail Services	IMAP	TCP/ UDP	143, 220	Internet Message Access Protocol	imap	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	POP3	TCP/ UDP	110	Post Office Protocol	pop3	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	Notes	TCP/ UDP	1352	Lotus Notes	notes	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	SMTP	TCP	25	Simple Mail Transfer Protocol	smtp	12.1(1)E 12.1(5)T 12.2(18)ZYA1
Directory	DHCP/ BOOTP	UDP	67, 68	Dynamic Host Configuration Protocol/Bootstrap Protocol	dhcp	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	Finger	TCP	79	Finger User Information Protocol	finger	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	DNS	TCP/ UDP	53	Domain Name System	dns	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	Kerberos	TCP/ UDP	88, 749	Kerberos Network Authentication Service	kerberos	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	LDAP	TCP/ UDP	389	Lightweight Directory Access Protocol	ldap	12.1(1)E 12.1(5)T 12.2(18)ZYA1

Table 1 NBAR-Supported Protocols (continued)

Category	Protocol	Type	Well-Known Port Number	Description	Syntax	Cisco IOS Release
Streaming Media	CU-SeeMe	TCP/ UDP	TCP: 7648, 7649 UDP: 24032	Desktop Video Conferencing	cuseeme	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	Netshow	TCP/ UDP	Dynamically Assigned	Microsoft Netshow	netshow	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	RealAudio	TCP/ UDP	Dynamically Assigned	RealAudio Streaming Protocol	realaudio	12.1(1)E 12.1(5)T
	StreamWorks	UDP	Dynamically Assigned	Xing Technology Stream Works Audio and Video	streamwork	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	VDOLive	TCP/ UDP	Static (7000) with inspection	VDOLive Streaming Video	vdolive	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	RTSP	TCP/ UDP	Dynamically Assigned	Real Time Streaming Protocol	rtsp	12.3(11)T 12.2(18)ZYA1
	MGCP	TCP/ UDP	2427, 2428, 2727	Media Gateway Control Protocol	mgcp	12.3(7)T 12.2(18)ZYA1
	YouTube[2]	TCP	Both static (80) and dynamically assigned	Online Video-Sharing Website	youtube	12.2(18)ZYA 12.2(18)ZYA1

Table 1 NBAR-Supported Protocols (continued)

Category	Protocol	Type	Well-Known Port Number	Description	Syntax	Cisco IOS Release
Internet	FTP	TCP	Dynamically Assigned	File Transfer Protocol	ftp	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	Gopher	TCP/ UDP	70	Internet Gopher Protocol	gopher	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	HTTP	TCP	80[3]	Hypertext Transfer Protocol	http	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	IRC	TCP/ UDP	194	Internet Relay Chat	irc	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	Telnet	TCP	23	Telnet Protocol	telnet	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	TFTP	UDP	Static (69) with inspection	Trivial File Transfer Protocol	tftp	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	NNTP	TCP/ UDP	119	Network News Transfer Protocol	nntp	12.1(1)E 12.1(5)T 12.2(18)ZYA1
Signaling	RSVP	UDP	1698, 1699	Resource Reservation Protocol	rsvp	12.1(1)E 12.1(5)T 12.2(18)ZYA1
RPC	NFS	TCP/ UDP	2049	Network File System	nfs	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	Sunrpc	TCP/ UDP	Dynamically Assigned	Sun Remote Procedure Call	sunrpc	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	MSN-messenger	TCP	1863	MSN Messenger Chat Messages[4]	msn-messenger	12.2(18)ZYA 12.2(18)ZYA1
	Yahoo-messenger	TCP	5050, 5101	Yahoo Messenger Chat Messages	yahoo-messenger	12.2(18)ZYA 12.2(18)ZYA1
	AOL-messenger	TCP	5190, 443	AOL Instant Messenger Chat Messages	aol-messenger	12.2(18)ZYA 12.2(18)ZYA1
Non-IP and LAN/ Legacy	NetBIOS	TCP/ UDP	137, 138, 139	NetBIOS over IP (MS Windows)	netbios	12.1(1)E 12.1(5)T 12.2(18)ZYA1

Table 1 NBAR-Supported Protocols (continued)

Category	Protocol	Type	Well-Known Port Number	Description	Syntax	Cisco IOS Release
Misc.	NTP	TCP/ UDP	123	Network Time Protocol	ntp	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	Printer	TCP/ UDP	515	Printer	printer	12.1(2)E 12.1(5)T 12.2(18)ZYA1
	X Windows	TCP	6000–6003	X11, X Windows	xwindows	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	r-commands	TCP	Dynamically Assigned	rsh, rlogin, rexec	rcmd	12.1(1)E 12.1(5)T 12.2(18)ZYA1
	AppleQTC	TCP/ UDP	458	Apple Quick Time	appleqtc	12.2(18)ZYA 12.2(18)ZYA1
	Chargen	TCP/ UDP	19	Character Generator	chargen	12.2(18)ZYA 12.2(18)ZYA1
	ClearCase	TCP/ UDP	371	Clear Case Protocol Software Informer	clearcase	12.2(18)ZYA 12.2(18)ZYA1
	Corba	TCP/ UDP	683, 684	Corba Internet Inter-Orb Protocol (IIOP)	corba-iiop	12.2(18)ZYA 12.2(18)ZYA1
	Daytime	TCP/ UDP	13	Daytime Protocol	daytime	12.2(18)ZYA 12.2(18)ZYA1
	Doom	TCP/ UDP	666	Doom	doom	12.2(18)ZYA 12.2(18)ZYA1
	Echo	TCP/ UDP	7	Echo Protocol	echo	12.2(18)ZYA 12.2(18)ZYA1
	IBM DB2	TCP/ UDP	523	IBM Information Management	ibm-db2	12.2(18)ZYA 12.2(18)ZYA1
	IPX	TCP/ UDP	213	Internet Packet Exchange	ipx	12.2(18)ZYA 12.2(18)ZYA1
	ISAKMP	TCP/ UDP	500	Internet Security Association and Key Management	isakmp	12.2(18)ZYA 12.2(18)ZYA1
	ISI-GL	TCP/ UDP	55	Interoperable Self Installation Graphics Language	isi-gl	12.2(18)ZYA 12.2(18)ZYA1
	KLogin	TCP	543	KLogin	klogin	12.2(18)ZYA 12.2(18)ZYA1
	KShell	TCP	544	KShell	kshell	12.2(18)ZYA 12.2(18)ZYA1

Table 1 NBAR-Supported Protocols (continued)

Category	Protocol	Type	Well-Known Port Number	Description	Syntax	Cisco IOS Release
Misc. (Continued)	LockD	TCP/ UDP	4045	LockD	lockd	12.2(18)ZYA 12.2(18)ZYA1
	Microsoft-DS	TCP/ UDP	445	Microsoft Directory Services	microsoftds	12.2(18)ZYA 12.2(18)ZYA1
	Nickname	TCP/ UDP	43	Nickname	nicname	12.2(18)ZYA 12.2(18)ZYA1
	NPP	TCP/ UDP	92	Network Payment Protocol	npp	12.2(18)ZYA 12.2(18)ZYA1
	ORASRV	TCP	1525	ORASRV	ora-srv	12.2(18)ZYA 12.2(18)ZYA1
	RTelnet	TCP/ UDP	107	Remote Telnet Service	rtelnet	12.2(18)ZYA 12.2(18)ZYA1
	RCP	TCP/ UDP	469	Rate Control Protocol	rcp	12.2(18)ZYA 12.2(18)ZYA1
	SQLExec	TCP/ UDP	9088	SQL Exec	sqlexec	12.2(18)ZYA 12.2(18)ZYA1
	Systat	TCP/ UDP	11	System Statistics	systat	12.2(18)ZYA 12.2(18)ZYA1
	TACACS	TCP/ UDP	49, 65	Terminal Access Controller Access-Control System	tacacs	12.2(18)ZYA 12.2(18)ZYA1
	Time	TCP/ UDP	37	Time	time	12.2(18)ZYA 12.2(18)ZYA1
	VNC	UDP	5800, 5900, 5901	Virtual Network Computing	vnc	12.2(18)ZYA 12.2(18)ZYA1
	Whois++	TCP/ UDP	63	Whois++	whois++	12.2(18)ZYA 12.2(18)ZYA1
	XDMCP	UDP	177	X Display Manager Control Protocol	xdmcp	12.2(18)ZYA 12.2(18)ZYA1

Table 1 NBAR-Supported Protocols (continued)

Category	Protocol	Type	Well-Known Port Number	Description	Syntax	Cisco IOS Release
Voice	H.323	TCP	Dynamically Assigned	H.323 Teleconferencing Protocol	h323	12.3(7)T 12.2(18)ZYA1
	RTCP	TCP/ UDP	Dynamically Assigned	Real-Time Control Protocol	rtcp	12.1E 12.2T 12.3 12.3T 12.3(7)T 12.2(18)ZYA1
	RTP	TCP/ UDP	Dynamically Assigned	Real-Time Transport Protocol Payload Classification	rtp	12.2(8)T 12.2(18)ZYA1
	Cisco-phone[5]	UDP	5060	Cisco IP Phones and PC-Based Unified Communicators	cisco-phone	12.2(18)ZYA 12.2(18)ZYA1
	SIP	TCP/ UPD	5060	Session Initiation Protocol	sip	12.3(7)T 12.2(18)ZYA1
	SCCP/ Skinny	TCP	2000, 2001, 2002	Skinny Client Control Protocol	skinny	12.3(7)T 12.2(18)ZYA1
	Skype[6]	TCP/ UDP	Dynamically Assigned	Peer-to-Peer VoIP Client Software	skype	12.4(4)T
	TelePresence	TCP/ UDP	Dynamically Assigned	Cisco TelePresence System	telepresence-media telepresence-control	12.2(18)ZYA2

Table 1 NBAR-Supported Protocols (continued)

Category	Protocol	Type	Well-Known Port Number	Description	Syntax	Cisco IOS Release
Peer-to-Peer File-Sharing Applications	BitTorrent	TCP	Dynamically Assigned, or 6881–6889	BitTorrent File Transfer Traffic	bittorrent	12.4(2)T 12.2(18)ZYA1
	Direct Connect	TCP/ UDP	411	Direct Connect File Transfer Traffic	directconnect	12.4(4)T 12.2(18)ZYA1
	eDonkey/ eMule	TCP	4662	eDonkey File-Sharing Application eMule traffic is also classified as eDonkey traffic in NBAR.	edonkey	12.3(11)T 12.2(18)ZYA1
	FastTrack	N/A	Dynamically Assigned	FastTrack	fasttrack	12.1(12c)E 12.2(18)ZYA1
	Gnutella	TCP	Dynamically Assigned	Gnutella	gnutella	12.1(12c)E 12.2(18)ZYA1
	KaZaA	TCP/ UPD	Dynamically Assigned	KaZaA Note that earlier KaZaA version 1 traffic can be classified using FastTrack.	kazaa2	12.2(8)T 12.2(18)ZYA1
	WinMX	TCP	6699	WinMX Traffic	winmx	12.3(7)T 12.2(18)ZYA1

1. For Release 12.2(18)ZYA, Cisco supports Exchange 03 and 07 only. MS client access is recognized, but web client access is not recognized.
2. For Release 12.2(18)ZYA, access to YouTube via HTTP only will be recognized.
3. In Release 12.3(4)T, the NBAR Extended Inspection for Hypertext Transfer Protocol (HTTP) Traffic feature was introduced. This feature allows NBAR to scan TCP ports that are not well known and to identify HTTP traffic that is traversing these ports.
4. For Release 12.2(18)ZYA, messages (“chat”) from Yahoo, MSN, and AOL are recognized. Messages from Lotus and SameTime are not recognized. Video and voice from Instant Messaging are also not recognized.
5. For Release 12.2(18)ZYA, only SIP and Skinny telephone connections (cisco-phone traffic connections) are recognized. H.323 telephone connections are not recognized.
6. Skype was introduced in Cisco IOS Release 12.4(4)T. As a result of this introduction, Skype is now native in (included with) the Cisco IOS software and uses the NBAR infrastructure new to Cisco IOS Release 12.4(4)T. Cisco software supports Skype 1.0, 2.5, and 3.0. Note that certain hardware platforms do not support Skype. For instance, Skype is not supported on the Catalyst 6500 series switch that is equipped with a Supervisor/PISA.

NBAR Memory Management

NBAR uses approximately 150 bytes of DRAM for each traffic flow that requires stateful inspection. (See Table 1 for a list of protocols supported by NBAR that require stateful inspection.)

When NBAR is configured, it allocates 1 MB of DRAM to support up to 5000 concurrent traffic flows. NBAR checks to see if more memory is required to handle additional concurrent stateful traffic flows. If such a need is detected, NBAR expands its memory usage in increments of 200 to 400 Kb.

Note This expansion of memory by NBAR does not apply if a PISA is in use.

NBAR Protocol Discovery

NBAR includes a feature called Protocol Discovery. Protocol discovery provides an easy way to discover the application protocols that are operating on an interface. For more information about protocol discovery, see the "Enabling Protocol Discovery" module.

Note With Cisco IOS Release 12.2(18)ZYA, intended for use on the Catalyst 6500 series switch that is equipped with a Supervisor 32/PISA, Protocol Discovery supports Layer 2 Etherchannels.

Non-intrusive Protocol Discovery

Cisco IOS Release 12.2(18)ZYA1 includes a feature called Non-intrusive Protocol Discovery. The Non-intrusive Protocol Discovery feature enables the Catalyst 6500 series switch that is equipped with a Supervisor 32/PISA to perform protocol discovery in out-of-band (that is, offline) mode. In offline mode, a copy of the network traffic is used to discover the application protocols that are operating on an interface, leaving the network traffic undisturbed and available for other purposes.

Non-intrusive Protocol Discovery is closely associated with a feature called Intelligent Traffic Redirect (ITR). ITR allows network administrators to optimize system performance by identifying the specific traffic that needs to be redirected to the Supervisor 32/PISA for deep-packet inspection.

Non-intrusive Protocol Discovery is achieved by enabling ITR on an interface on which protocol discovery has been enabled. For more information about the commands used to enable ITR, see the *Catalyst Supervisor Engine 32 PISA IOS Command Reference*. For more information about protocol discovery, see the "Enabling Protocol Discovery" module.

Note For the Non-intrusive Protocol Discovery feature to function properly, no other "intrusive" features (for example, Flexible Packet Matching [FPM]) can be in use on the interface in either the input or output direction. An intrusive feature is one that some how manipulates the packets (such as modifying a statistic or a packet counter). If such a feature is in use, the actual traffic (and not a copy of the traffic) is redirected.

NBAR Protocol Discovery MIB

The NBAR Protocol Discovery Management Information Base (MIB) expands the capabilities of NBAR Protocol Discovery by providing the following new functionality through Simple Network Management Protocol (SNMP):

- Enable or disable Protocol Discovery per interface.
- Display Protocol Discovery statistics.
- Configure and view multiple top-n tables that list protocols by bandwidth usage.
- Configure thresholds based on traffic of particular NBAR-supported protocols or applications that report breaches and send notifications when these thresholds are crossed.

For more information about the NBAR Protocol Discovery MIB, see the "Network-Based Application Recognition Protocol Discovery Management Information Base" module.

NBAR Configuration Processes

Configuring NBAR consists of the following processes:

- Enabling Protocol Discovery (required)

  When you configure NBAR, the first process is to enable Protocol Discovery.

- Configuring NBAR using the MQC (optional)

  After you enable Protocol Discovery, you have the option to configure NBAR using the functionality of the MQC.

- Adding application recognition modules (also known as Packet Description Language Modules [PDLMs]) (optional)

  Adding PDLMs extends the functionality of NBAR by enabling NBAR to recognize additional protocols on your network.

- Creating custom protocols (optional)

  Custom protocols extend the capability of NBAR Protocol Discovery to classify and monitor additional static port applications and allow NBAR to classify nonsupported static port traffic.

Where to Go Next

Begin configuring NBAR by first enabling Protocol Discovery. To enable Protocol Discovery, see the "Enabling Protocol Discovery" module.

Additional References

The following sections provide references related to classifying network traffic using NBAR.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
QoS features and functionality	"Quality of Service Overview" module
QoS features and functionality on the Catalyst 6500 series switch	"Configuring PFC QoS" module of the *Catalyst Supervisor Engine 32 PISA Cisco IOS Software Configuration Guide*, Release 12.2ZY
Classifying network traffic if not using NBAR	"Classifying Network Traffic" module
FWSM and its connection features	Configuring Advanced Connection Features" module of the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Configuration Guide*
FWSM commands	*Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Service Module Command Reference Guide*
IDSM	"Configuring IDSM-2" module of the *Configuring the Cisco Intrusion Prevention System Sensor Using the Command Line Interface*
SPAN or RSPAN	"Configuring SPAN and RSPAN" module of the *Catalyst 6500 Series Software Configuration Guide*
VACL Capture	"VACL Capture for Granular Traffic Analysis with Cisco Catalyst 6000/6500 Running Cisco IOS Software" module
Catalyst 6500 series switch and QoS	"Configuring QoS" module of the *Catalyst 6500 Series Software Configuration Guide*
Commands used to enable ITR on the Catalyst 6500 series switch equipped with a Supervisor 32/PISA	*Catalyst Supervisor Engine 32 PISA IOS Command Reference*.
FPM	"Flexible Packet Matching" module
FPM eXtensible Markup Language (XML) Configuration	"Flexible Packet Matching XML Configuration" module
Marking network traffic	"Marking Network Traffic" module
CISCO-NBAR-PROTOCOL-DISCOVERY MIB	"Network-Based Application Recognition Protocol Discovery Management Information Base" module
CEF	"Cisco Express Forwarding Features Roadmap" module
AutoQoS,[1] AutoQos for the Enterprise, VoIP traffic	"AutoQoS—VoIP" module; "AutoQos for the Enterprise" module
NBAR Protocol Discovery MIB	"Network-Based Application Recognition Protocol Discovery Management Information Base" module
Enabling Protocol Discovery	"Enabling Protocol Discovery" module
Configuring NBAR using the MQC	"Configuring NBAR Using the MQC" module

Related Topic	Document Title
Adding application recognition modules (also known as PDLMs)	"Adding Application Recognition Modules" module
Creating a custom protocol	"Creating a Custom Protocol" module

1. Cisco IOS Release 12.2(18)ZY does not support either the AutoQoS—Voice over IP (VoIP) feature or the AutoQoS for the Enterprise feature on the Catalyst 6500 series switch.

Standards

Standards	Title
ISO 0009	*File Transfer Protocol (FTP)*
ISO 0013	*Domain Names - Concepts and Facilities*
ISO 0033	*The TFTP Protocol (Revision 2)*
ISO 0034	*Routing Information Protocol*
ISO 0053	*Post Office Protocol - Version 3*
ISO 0056	*RIP Version 2*

MIBs

MIBs	MIBs Link
CISCO-NBAR-PROTOCOL-DISCOVERY MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 742	*NAME/FINGER Protocol*
RFC 759	*Internet Message Protocol*
RFC 768	*User Datagram Protocol*
RFC 792	*Internet Control Message Protocol*
RFC 793	*Transmission Control Protocol*
RFC 821	*Simple Mail Transfer Protocol*
RFC 827	*Exterior Gateway Protocol*
RFC 854	*Telnet Protocol Specification*
RFC 888	*"STUB" Exterior Gateway Protocol*
RFC 904	*Exterior Gateway Protocol Formal Specification*
RFC 951	*Bootstrap Protocol*

RFC	Title
RFC 959	*File Transfer Protocol*
RFC 977	*Network News Transfer Protocol*
RFC 1001	*Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Concepts and Methods*
RFC 1002	*Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Detailed Specifications*
RFC 1057	*RPC: Remote Procedure Call*
RFC 1094	*NFS: Network File System Protocol Specification*
RFC 1112	*Host Extensions for IP Multicasting*
RFC 1157	*Simple Network Management Protocol*
RFC 1282	*BSD Rlogin*
RFC 1288	*The Finger User Information Protocol*
RFC 1305	*Network Time Protocol*
RFC 1350	*The TFTP Protocol (Revision 2)*
RFC 1436	*The Internet Gopher Protocol*
RFC 1459	*Internet Relay Chat Protocol*
RFC 1510	*The Kerberos Network Authentication Service*
RFC 1542	*Clarifications and Extensions for the Bootstrap Protocol*
RFC 1579	*Firewall-Friendly FTP*
RFC 1583	*OSPF Version 2*
RFC 1657	*Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol*
RFC 1701	*Generic Routing Encapsulation*
RFC 1730	*Internet Message Access Protocol—Version 4*
RFC 1771	*A Border Gateway Protocol 4 (BGP-4)*
RFC 1777	*Lightweight Directory Access Protocol*
RFC 1831	*RPC: Remote Procedure Call Protocol Specification Version 2*
RFC 1889	*A Transport Protocol for Real-Time Applications*
RFC 1890	*RTP Profile for Audio and Video Conferences with Minimal Control*
RFC 1928	*SOCKS Protocol Version 5*
RFC 1939	*Post Office Protocol—Version 3*
RFC 1945	*Hypertext Transfer Protocol—HTTP/1.0*
RFC 1964	*The Kerberos Version 5 GSS-API Mechanism*
RFC 2045	*Multipurpose Internet Mail Extension (MIME) Part One: Format of Internet Message Bodies*
RFC 2060	*Internet Message Access Protocol—Version 4 rev1*
RFC 2068	*Hypertext Transfer Protocol—HTTP/1.1*
RFC 2131	*Dynamic Host Configuration Protocol*

RFC	Title
RFC 2205	*Resource ReSerVation Protocol (RSVP)—Version 1 Functional Specification*
RFC 2236	*Internet Group Management Protocol, Version 2*
RFC 2251	*Lightweight Directory Access Protocol (v3)*
RFC 2252	*Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions*
RFC 2253	*Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names*
RFC 2326	*Real Time Streaming Protocol (RTSP)*
RFC 2401	*Security Architecture for the Internet Protocol*
RFC 2406	*IP Encapsulating Security Payload*
RFC 2453	*RIP Version 2*
RFC 2616	*Hypertext Transfer Protocol—HTTP/1.1* **Note** This RFC updates RFC 2068.

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Glossary

encryption—Encryption is the application of a specific algorithm to data so as to alter the appearance of the data, making it incomprehensible to those who are not authorized to see the information.

HTTP—Hypertext Transfer Protocol. The protocol used by web browsers and web servers to transfer files, such as text and graphic files.

IANA—Internet Assigned Numbers Authority. An organization operated under the auspices of the Internet Society (ISOC) as a part of the Internet Architecture Board (IAB). IANA delegates authority for IP address-space allocation and domain-name assignment to the InterNIC and other organizations. IANA also maintains a database of assigned protocol identifiers used in the TCP/IP stack, including autonomous system numbers.

LAN—local-area network. A high-speed, low-error data network that covers a relatively small geographic area (up to a few thousand meters). LANs connect workstations, peripherals, terminals, and other devices in a single building or other geographically limited area. LAN standards specify cabling and signaling at the physical and data link layers of the Open System Interconnection (OSI) model. Ethernet, FDDI, and Token Ring are widely used LAN technologies.

MIME—Multipurpose Internet Mail Extension. The standard for transmitting nontext data (or data that cannot be represented in plain ASCII code) in Internet mail, such as binary, foreign language text (such as Russian or Chinese), audio, and video data. MIME is defined in RFC 2045: *Multipurpose Internet Mail Extension (MIME) Part One: Format of Internet Message Bodies*.

MPLS—Multiprotocol Label Switching. A switching method that forwards IP traffic using a label. This label instructs the routers and the switches in the network where to forward the packets based on preestablished IP routing information.

MQC—Modular Quality of Service Command-Line Interface. A command-line interface that allows you to define traffic classes, create and configure traffic policies (policy maps), and then attach the policy maps to interfaces. The policy maps are used to apply the appropriate quality of service (QoS) to network traffic.

dNBAR—Distributed Network-Based Application Recognition. dNBAR is NBAR used on the Cisco 7500 router with a Versatile Interface Processor (VIP) and on the Catalyst 6500 family of switches with a FlexWAN module or serial interface processor (SIP). The implementation of NBAR and dNBAR is identical.

NBAR—Network-Based Application Recognition. A classification engine that recognizes and classifies a wide variety of protocols and applications. When NBAR recognizes and classifies a protocol or application, the network can be configured to apply the appropriate quality of service (QoS) for that application or traffic with that protocol.

PDLM—Packet Description Language Module. A file that contains Packet Description Language statements used to define the signature of one or more application protocols.

Protocol Discovery—A feature included with NBAR. Protocol Discovery provides a way to discover the application protocols that are operating on an interface.

QoS—quality of service. A measure of performance for a transmission system that reflects its transmission quality and service availability.

RTCP—RTP Control Protocol. A protocol that monitors the QoS of an IPv6 Real-Time Transport Protocol (RTP) connection and conveys information about the ongoing session.

RTSP—Real Time Streaming Protocol. A means for enabling the controlled delivery of real-time data, such as audio and video. Sources of data can include both live data feeds, such as live audio and video, and stored content, such as prerecorded events. RTSP is designed to work with established protocols, such as Real-Time Transport Protocol (RTP) and HTTP.

stateful protocol—A protocol that uses TCP and UDP port numbers that are determined at connection time.

static protocol—A protocol that uses well-defined (predetermined) TCP and UDP ports for communication.

subport classification—The classification of network traffic by information that is contained in the packet payload; that is, information found beyond the TCP or UDP port number.

TCP—Transmission Control Protocol. A connection-oriented transport layer protocol that provides reliable full-duplex data transmission. TCP is part of the TCP/IP protocol stack.

tunneling—Tunneling is an architecture that is designed to provide the services necessary to implement any standard point-to-point encapsulation scheme.

UDP—User Datagram Protocol. A connectionless transport layer protocol in the TCP /IP protocol stack. UDP is a simple protocol that exchanges datagrams without acknowledgments or guaranteed delivery, requiring that error processing and retransmission be handled by other protocols. UDP is defined in RFC 768: *User Datagram Protocol*.

WAN—wide-area network. A data communications network that serves users across a broad geographic area and often uses transmission devices provided by common carriers.

Enabling Protocol Discovery

First Published: April 4, 2006
Last Updated: August 7, 2008

Network-Based Application Recognition (NBAR) includes a feature called Protocol Discovery. Protocol Discovery provides an easy way to discover the application protocols that are operating on an interface. When you configure NBAR, the first task is to enable Protocol Discovery.

This module contains concepts and tasks for enabling the Protocol Discovery feature.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Enabling Protocol Discovery" section on page 7.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Enabling Protocol Discovery

Before enabling Protocol Discovery, read the information in the "Classifying Network Traffic Using NBAR" module.

Information About Protocol Discovery

Before enabling Protocol Discovery, you should understand the following concept:

- Protocol Discovery Functionality, page 2

Protocol Discovery Functionality

NBAR determines which protocols and applications are currently running on your network. NBAR includes a feature called Protocol Discovery. Protocol Discovery provides an easy way of discovering the application protocols that are operating on an interface so that appropriate quality of service (QoS) features can be applied. With Protocol Discovery, you can discover any protocol traffic that is supported by NBAR and obtain statistics that are associated with that protocol.

Protocol Discovery maintains the following per-protocol statistics for enabled interfaces:

- Total number of input packets and bytes
- Total number of output packets and bytes
- Input bit rates
- Output bit rates

The statistics can then be used when you later define classes and traffic policies (sometimes known as policy maps) for each traffic class. The traffic policies (policy maps) are used to apply specific QoS features and functionality to the traffic classes.

How to Configure Protocol Discovery

This section contains the following tasks:

- Enabling Protocol Discovery on an Interface, page 2 (required)
- Reporting Protocol Discovery Statistics, page 3 (optional)

Enabling Protocol Discovery on an Interface

To enable Protocol Discovery on an interface, perform the following steps.

ip nbar protocol-discovery Command and Layer 2/3 Etherchannel Support

The **ip nbar protocol-discovery** command is used to enable Protocol Discovery on an interface. With Cisco IOS Release 12.2(18)ZYA, intended for use on the Cisco 6500 series switch that is equipped with a Supervisor 32/PISA, the **ip nbar protocol-discovery** command is supported on both Layer 2 and Layer 3 Etherchannels.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **ip nbar protocol-discovery**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`interface` *`type number`* [*`name-tag`*] **Example:** `Router(config)# interface ethernet 2/4`	Configures an interface type and enters interface configuration mode. • Enter the interface type and the interface number.
Step 4	`ip nbar protocol-discovery` **Example:** `Router(config-if)# ip nbar protocol-discovery`	Configures NBAR to discover traffic for all protocols known to NBAR on a particular interface.
Step 5	`end` **Example:** `Router(config-if)# end`	(Optional) Exits interface configuration mode.

Reporting Protocol Discovery Statistics

To display a report of the Protocol Discovery statistics per interface, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show policy-map interface** *type number*
3. **show ip nbar protocol-discovery** [**interface** *type number*] [**stats** {**byte-count** | **bit-rate** | **packet-count** | **max-bit-rate**}] [**protocol** *protocol-name* | **top-n** *number*]
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show policy-map interface** *type number* **Example:** `Router# show policy-map interface Fastethernet 6/0`	(Optional) Displays the packet and class statistics for all policy maps on the specified interface. • Enter the interface type and the interface number.
Step 3	**show ip nbar protocol-discovery** [**interface** *type number*] [**stats** {**byte-count** | **bit-rate** | **packet-count** | **max-bit-rate**}] [**protocol** *protocol-name* | **top-n** *number*] **Example:** `Router# show ip nbar protocol-discovery interface Fastethernet 6/0`	Displays the statistics gathered by the NBAR Protocol Discovery feature. • (Optional) Enter keywords and arguments to fine-tune the statistics displayed.
Step 4	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

Configuration Examples for Enabling Protocol Discovery

This section provides the following configuration examples:

- Enabling Protocol Discovery on an Interface: Example, page 4
- Reporting Protocol Discovery Statistics: Example, page 5

Enabling Protocol Discovery on an Interface: Example

In the following sample configuration, Protocol Discovery is enabled on Ethernet interface 2/4.

```
Router> enable
Router# configure terminal
Router(config)# interface ethernet 2/4
Router(config-if)# ip nbar protocol-discovery
Router(config-if)# end
```

Reporting Protocol Discovery Statistics: Example

The following example displays output from the **show ip nbar protocol-discovery** command for the five most active protocols on an Ethernet interface:

```
Router# show ip nbar protocol-discovery top-n 5

 Ethernet2/0
                            Input                    Output
                            -----                    ------
  Protocol                  Packet Count             Packet Count
                            Byte Count               Byte Count
                            30sec Bit Rate (bps)     30sec Bit Rate (bps)
                            30sec Max Bit Rate (bps) 30sec Max Bit Rate (bps)
--------------------------- ------------------------ ------------------------
    rtp                       3272685                  3272685
                            242050604                242050604
                              768000                   768000
                              2002000                  2002000
    gnutella                  513574                   513574
                              118779716                118779716
                              383000                   383000
                              987000                   987000
    ftp                       482183                   482183
                              37606237                 37606237
                              121000                   121000
                              312000                   312000
    http                      144709                   144709
                              32351383                 32351383
                              105000                   105000
                              269000                   269000
    netbios                   96606                    96606
                              10627650                 10627650
                              36000                    36000
                              88000                    88000
    unknown                   1724428                  1724428
                              534038683                534038683
                              2754000                  2754000
                              4405000                  4405000
    Total                     6298724                  6298724
                              989303872                989303872
                              4213000                  4213000
                              8177000                  8177000
```

Where to Go Next

After you enable Protocol Discovery, you have the option to configure NBAR using the Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC). To configure NBAR using the MQC, see the "Configuring NBAR Using the MQC" module.

Additional References

The following sections provide references related to enabling Protocol Discovery.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
Concepts and information about NBAR	"Classifying Network Traffic Using NBAR" module
Configuring NBAR using the MQC	"Configuring NBAR Using the MQC" module
Adding application recognition modules (also known as PDLMs)	"Adding Application Recognition Modules" module
Creating a custom protocol	"Creating a Custom Protocol" module

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Enabling Protocol Discovery

Table 1 lists the release history for this feature.

For information on a feature in this technology that is not documented here, see the "Classifying Network Traffic Using NBAR Features Roadmap" module.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Enabling Protocol Discovery

Feature Name	Releases	Feature Information
NBAR—Network-Based Application Recognition	12.2(18)ZYA	Integrates NBAR and Firewall Service Module (FWSM) functionality on the Catalyst 6500 series switch that is equipped with a Supervisor 32/programmable intelligent services accelerator (PISA). The following commands were modified: **ip nbar protocol-discovery, show ip nbar protocol-discovery**.

Configuring NBAR Using the MQC

First Published: April 4, 2006
Last Updated: August 7, 2008

After you enable Protocol Discovery, you can configure Network-Based Application Recognition (NBAR) using the functionality of the Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC). The MQC uses traffic classes and traffic policies (policy maps) to apply QoS features to classes of traffic and applications recognized by NBAR.

This module contains concepts and tasks for configuring NBAR using the MQC.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Configuring NBAR Using the MQC" section on page 13.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Configuring NBAR Using the MQC

- Before configuring NBAR using the MQC, read the information inthe "Classifying Network Traffic Using NBAR" module.
- As applicable, enable Protocol Discovery and use it to obtain statistics about the protocols and applications that are used in your network. You will need this information when using the MQC.

Note This prerequisite assumes that you do not already have this information about the protocols and applications in use in your network.

Information About Configuring NBAR Using the MQC

Before configuring NBAR using the MQC, you should understand the following concepts:

- NBAR and the MQC Functionality, page 2
- NBAR and the match protocol Commands, page 3

NBAR and the MQC Functionality

To configure NBAR using the MQC, you must define a traffic class, configure a traffic policy (policy map), and then attach that traffic policy to the appropriate interface. These three tasks can be accomplished by using the MQC. The MQC is a command-line interface that allows you to define traffic classes, create and configure traffic policies (policy maps), and then attach these traffic policies to interfaces.

In the MQC, the **class-map** command is used to define a traffic class (which is then associated with a traffic policy). The purpose of a traffic class is to classify traffic.

Using the MQC to configure NBAR consists of the following:

- Defining a traffic class with the **class-map** command.
- Creating a traffic policy by associating the traffic class with one or more QoS features (using the **policy-map** command).
- Attaching the traffic policy to the interface with the **service-policy** command.

A traffic class contains three major elements: a name, one or more **match** commands, and, if more than one **match** command exists in the traffic class, an instruction on how to evaluate these **match** commands (that is, **match-all** or **match-any**). The traffic class is named in the **class-map** command line; for example, if you enter the **class-map cisco** command while configuring the traffic class in the CLI, the traffic class would be named "cisco."

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.

Note For NBAR, the **match protocol** commands are used to specify the match criteria, as described in the "NBAR and the match protocol Commands" section on page 3.

NBAR and the match protocol Commands

NBAR recognizes specific network protocols and network applications that are used in your network. Once a protocol or application is recognized by NBAR, you can use the MQC to group the packets associated with those protocols or applications into classes. These classes are grouped on the basis of whether the packets conform to certain criteria.

For NBAR, the criterion is whether the packet matches a specific protocol or application known to NBAR. Using the MQC, network traffic with one network protocol (citrix, for example) can be placed into one traffic class, while traffic that matches a different network protocol (gnutella, for example) can be placed into another traffic class. Later, the network traffic within each class can be given the appropriate QoS treatment by using a traffic policy (policy map).

You specify the criteria used to classify traffic by using a **match protocol** command. Table 1 lists some of the available **match protocol** commands and the corresponding protocol or traffic type recognized and supported by NBAR.

Note For a more complete list of the protocol types supported by NBAR, see the "Classifying Network Traffic Using NBAR" module.

Table 1 match protocol Commands and Corresponding Protocol or Traffic Type

match protocol Command[1]	Protocol Type
match protocol (NBAR)	Protocol type supported by NBAR
match protocol citrix	Citrix protocol
match protocol fasttrack	FastTrack peer-to-peer traffic
match protocol gnutella	Gnutella peer-to-peer traffic
match protocol http	Hypertext Transfer Protocol
match protocol rtp	Real-Time Transport Protocol traffic

1. Cisco IOS **match protocol** commands can vary by release. For more information, see the command documentation for the Cisco IOS release that you are using.

How to Configure NBAR Using the MQC

This section contains the following tasks:

- Configuring a Traffic Class, page 4 (required)
- Configuring a Traffic Policy, page 5 (required)
- Attaching a Traffic Policy to an Interface or Subinterface, page 6 (required)
- Verifying the NBAR Traffic Classes, Traffic Policies, and Protocol-to-Port Mappings, page 8 (optional)

Configuring a Traffic Class

Traffic classes can be used to organize packets into groups based on a user-specified criteria. For example, traffic classes can be configured to match packets on the basis of the protocol type or application recognized by NBAR. In this task, the traffic class is configured to match on the basis of the Citrix protocol type.

To configure a traffic class, perform the following steps.

Note The **match protocol citrix** command is shown in Step 4. The **match protocol citrix** command is just an example of one of the **match protocol** commands that can be used. For a complete list of **match protocol** commands, see the command documentation for the Cisco IOS release that you are using.

Restrictions

Typically, a single traffic class contains one or more **match** commands that can be used to organize packets into groups on the basis of a protocol type or application. You can create as many traffic classes as needed. However, for Cisco IOS Release 12.2(18)ZY, the following restrictions apply:

- A single traffic class can be configured to match a maximum of 8 protocols or applications.
- Multiple traffic classes can be configured to match a cumulative maximum of 95 protocols or applications.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** [**match-all** | **match-any**] *class-map-name*
4. **match protocol citrix**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`class-map [match-all | match-any] class-map-name` **Example:** `Router(config)# class-map cmap1`	Creates a class map to be used for matching packets to a specified class and enters class-map configuration mode. • Enter the name of the class map.

	Command or Action	Purpose
Step 4	`match protocol citrix` **Example:** `Router(config-cmap)# match protocol citrix`	Configures NBAR to match Citrix traffic. **Note** The **match protocol citrix** command is just an example of one of the **match protocol** commands that can be used. For a complete list of **match protocol** commands, see the command documentation for the Cisco IOS release that you are using. **Note** For Cisco IOS Release 12.2(18)ZY, a maximum of 8 **match protocol** commands can be configured in a single traffic class.
Step 5	**end** **Example:** `Router(config-cmap)# end`	(Optional) Returns to privileged EXEC mode.

Configuring a Traffic Policy

Traffic that matches a user-specified criterion can be organized into a specific class that can, in turn, receive specific user-defined QoS treatment when that class is included in a policy map.

To configure a traffic policy, perform the following steps.

Note The **bandwidth** command is shown in Step 5. The **bandwidth** command configures the QoS feature class-based weighted fair queuing (CBWFQ). CBWFQ is just an example of a QoS feature that can be configured. Use the appropriate command for the QoS feature that you want to use.

As of Cisco IOS Release 12.2(18)ZY, CBWFQ is not supported on the Catalyst 6500 series switch that is equipped with a Supervisor 32/programmable intelligent services accelerator (PISA).

Restrictions

For Cisco IOS Release 12.2(18)ZY, an existing traffic policy (policy map) cannot be modified if the traffic policy is already attached to the interface. To remove the policy map from the interface, use the **no** form of the **service-policy** command.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map-name* **Example:** `Router(config)# policy-map policy1`	Creates or modifies a policy map that can be attached toone or more interfaces and enters policy-map configuration mode. • Enter the name of the policy map.
Step 4	**class** {*class-name* | **class-default**} **Example:** `Router(config-pmap)# class class1`	Specifies the name of the class whose policy you want to create or change and enters policy-map class configuration mode. • Enter the specific class name or enter the **class-default** keyword.
Step 5	**bandwidth** {*bandwidth-kbps* | **remaining percent** *percentage* | **percent** *percentage*} **Example:** `Router(config-pmap-c)# bandwidth percent 50`	(Optional) Specifies or modifies the bandwidth allocated for a class belonging to a policy map. • Enter the amount of bandwidth as a number of kbps, a relative percentage of bandwidth, or an absolute amount of bandwidth. **Note** The **bandwidth** command configures the QoS feature class-based weighted fair queuing (CBWFQ). CBWFQ is just an example of a QoS feature that can be configured. Use the appropriate command for the QoS feature that you want to use. **Note** As of Cisco IOS Release 12.2(18)ZY, CBWFQ is not supported on the Catalyst 6500 series switch that is equipped with a Supervisor 32/PISA.
Step 6	**end** **Example:** `Router(config-pmap-c)# end`	(Optional) Returns to privileged EXEC mode.

Attaching a Traffic Policy to an Interface or Subinterface

After a policy map is created, the next step is to attach the traffic policy (sometimes called a policy map) to an interface or subinterface. Traffic policies can be attached to either the input or output direction of the interface or subinterface.

Note Depending on the needs of your network, you may need to attach the traffic policy to an ATM PVC, a Frame Relay data-link connection identifier (DLCI), or other type of interface.

To attach a traffic policy (policy map) to an interface, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **pvc** [*name*] *vpi/vci* [**ilmi** | **qsaal** | **smds** | **l2transport**]
5. **exit**
6. **service-policy** {**input** | **output**} *policy-map-name*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`interface` *`type number`* `[`*`name-tag`*`]` **Example:** `Router(config)# interface ethernet 2/4`	Configures an interface type and enters interface configuration mode. • Enter the interface type and the interface number.
Step 4	`pvc [`*`name`*`] `*`vpi/vci`*` [ilmi | qsaal | smds | l2transport]` **Example:** `Router(config-if)# pvc cisco 0/16`	(Optional) Creates or assigns a name to an ATM permanent virtual circuit (PVC), specifies the encapsulation type on an ATM PVC, and enters ATM virtual circuit configuration mode. • Enter the PVC name, the ATM network virtual path identifier, and the network virtual channel identifier. **Note** This step is required only if you are attaching the policy map to an ATM PVC. If you are not attaching the policy map to an ATM PVC, advance to Step 6.
Step 5	`exit` **Example:** `Router(config-atm-vc)# exit`	(Optional) Returns to interface configuration mode. **Note** This step is required only if you are attaching the policy map to an ATM PVC and you completed Step 4. If you are not attaching the policy map to an ATM PVC, advance to Step 6.

	Command or Action	Purpose
Step 6	**service-policy** {**input** | **output**} *policy-map-name* **Example:** `Router(config-if)# service-policy input policy1`	Attaches a policy map (traffic policy) to an input or output interface. • Specify either the **input** or **output** keyword, and enter the policy map name. **Note** Policy maps can be configured on ingress or egress routers. They can also be attached in the input or output direction of an interface. The direction (input or output) and the router (ingress or egress) to which the policy map should be attached vary according your network configuration. When using the **service-policy** command to attach the policy map to an interface, be sure to choose the router and the interface direction that are appropriate for your network configuration. **Note** After you use the **service-policy** command, you may see two messages similar to the following: `%PISA-6-NBAR_ENABLED: feature accelerated on input direction of: [interface name and type]` `%PISA-6-NBAR_ENABLED: feature accelerated on output direction of: [interface name and type]` While both of these messages appear, NBAR is enabled in the direction specified by the **input** or **output** keyword *only*.
Step 7	**end** **Example:** `Router(config-if)# end`	(Optional) Returns to privileged EXEC mode.

Verifying the NBAR Traffic Classes, Traffic Policies, and Protocol-to-Port Mappings

After you create the traffic classes and traffic policies (policy maps), you may want to verify that the end result is the one you intended. That is, you may want to verify whether your traffic is being classified correctly and whether it is receiving the QoS treatment as intended. You may also want to verify that the protocol-to-port mappings are correct.

To verify the NBAR traffic classes, traffic policies, and protocol-to-port mappings, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show class-map** [*class-map-name*]

3. **show policy-map** [*policy-map*]
4. **show policy-map interface** *interface-name*
5. **show ip nbar port-map** [*protocol-name*]
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show class-map** [*class-map-name*] **Example:** `Router# show class-map`	(Optional) Displays all class maps and their matching criteria. • (Optional) Enter the name of a specific class map.
Step 3	**show policy-map** [*policy-map*] **Example:** `Router# show policy-map`	(Optional) Displays the configuration of all classes for a specified service policy map or all classes for all existing policy maps. • (Optional) Enter the name of a specific policy map.
Step 4	**show policy-map interface** *type number* **Example:** `Router# show policy-map interface Fastethernet 6/0`	(Optional) Displays the packet and class statistics for all policy maps on the specified interface. • Enter the interface type and the interface number.
Step 5	**show ip nbar port-map** [*protocol-name*] **Example:** `Router# show ip nbar port-map`	(Optional) Displays the current protocol-to-port mappings in use by NBAR. • (Optional) Enter a specific protocol name.
Step 6	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

Configuration Examples for Configuring NBAR Using the MQC

This section provides the following configuration examples:

Configuring a Traffic Class: Example

In the following example, a class called cmap1 has been configured. All traffic that matches the citrix protocol will be placed in the cmap1 class.

```
Router> enable
Router# configure terminal
Router(config)# class-map cmap1
Router(config-cmap)# match protocol citrix
Router(config-cmap)# end
```

Configuring a Traffic Policy: Example

In the following example, a traffic policy (policy map) called policy1 has been configured. Policy1 contains a class called class1, within which CBWFQ has been enabled.

```
Router> enable
Router# configure terminal
Router(config)# policy-map policy1
Router(config-pmap)# class class1
Router(config-pmap-c)# bandwidth percent 50
Router(config-pmap-c)# end
```

Note In the above example, the **bandwidth** command is used to enable Class-Based Weighted Fair Queuing (CBWFQ). CBWFQ is only an example of one QoS feature that can be applied in a policy map. Use the appropriate command for the QoS feature that you want to use.

As of Cisco IOS Release 12.2(18)ZY, CBWFQ is not supported on the Catalyst 6500 series switch that is equipped with a Supervisor 32/PISA.

Attaching a Traffic Policy to an Interface or Subinterface: Example

In the following example, the traffic policy (policy map) called policy1 has been attached to Ethernet interface 2/4 in the input direction of the interface.

```
Router> enable
Router# configure terminal
Router(config)# interface ethernet 2/4
Router(config-if)# service-policy input policy1
Router(config-if)# end
```

Verifying the NBAR Protocol-to-Port Mappings: Example

The following is sample output of the **show ip nbar port-map** command. This command displays the current protocol-to-port mappings in use by NBAR. Use the display to verify that these mappings are correct.

```
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
```

If the **ip nbar port-map** command has been used, the **show ip nbar port-map** command displays the ports assigned to the protocol.

If the **no ip nbar port-map** command has been used, the **show ip nbar port-map** command displays the default ports. To limit the display to a specific protocol, use the *protocol-name* argument of the **show ip nbar port-map** command.

Where to Go Next

To add application recognition modules (also known as Packet Description Language Modules or PDLMs) to your network, see the "Adding Application Recognition Modules" module.

To classify network traffic on the basis of a custom protocol, see the "Creating a Custom Protocol" module.

Additional References

The following sections provide references related to configuring NBAR using the MQC.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
QoS features and functionality on the Catalyst 6500 series switch	"Configuring PFC QoS" chapter of the *Catalyst Supervisor Engine 32 PISA Cisco IOS Software Configuration Guide*, Release 12.2ZY
MQC, traffic policies (policy maps), and traffic classes	"Applying QoS Features Using the MQC" module
CBWFQ	"Configuring Weighted Fair Queueing" module
Concepts and information about NBAR	"Classifying Network Traffic Using NBAR" module
Information about enabling Protocol Discovery	"Enabling Protocol Discovery" module
Information about adding application recognition modules (also known as PDLMs)	"Adding Application Recognition Modules" module
Creating a custom protocol	"Creating a Custom Protocol" module

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Configuring NBAR Using the MQC

Table 2 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(1) or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the "Classifying Network Traffic Using NBAR Features Roadmap" module.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 2 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 2 Feature Information for Configuring NBAR Using the MQC

Feature Name	Releases	Feature Information
QoS: DirectConnect PDLM	12.4(4)T	Provides support for the DirectConnect protocol and Packet Description Language Module (PDLM). The DirectConnect protocol can now be recognized when using the MQC to classify traffic. The following sections provide information about the QoS: DirectConnect PDLM feature: • Information About Configuring NBAR Using the MQC, page 2 • How to Configure NBAR Using the MQC, page 3
QoS: Skype Classification	12.4(4)T	Provides support for the Skype protocol. The Skype protocol can now be recognized when using the MQC to classify traffic. **Note** Cisco currently supports Skype Version 1 only. The following sections provide information about the QoS: Skype Classification feature: • Information About Configuring NBAR Using the MQC, page 2 • How to Configure NBAR Using the MQC, page 3

Table 2 Feature Information for Configuring NBAR Using the MQC (continued)

Feature Name	Releases	Feature Information
NBAR—BitTorrent PDLM	12.4(2)T	Provides support for the BitTorrent PDLM and protocol. The BitTorrent protocol can now be recognized when using the MQC to classify traffic. The following sections provide information about the NBAR–BitTorrent PDLM feature: • Information About Configuring NBAR Using the MQC, page 2 • How to Configure NBAR Using the MQC, page 3
NBAR—Citrix ICA Published Applications	12.4(2)T	Enables NBAR to classify traffic on the basis of the Citrix Independent Computing Architecture (ICA) published application name and tag number. The following sections provide information about the NBAR–Citrix ICA Published Applications feature: • Information About Configuring NBAR Using the MQC, page 2 • How to Configure NBAR Using the MQC, page 3
NBAR—Multiple Matches Per Port	12.4(2)T	Provides the ability for NBAR to distinguish between values of an attribute within the traffic stream of a particular application on a TCP or UDP port. The following sections provide information about the NBAR–Multiple Matches Per Port feature: • Information About Configuring NBAR Using the MQC, page 2 • How to Configure NBAR Using the MQC, page 3
NBAR Extended Inspection for HTTP Traffic	12.3(4)T	Allows NBAR to scan TCP ports that are not well known and identify HTTP traffic that traverses these ports. The following sections provide information about the NBAR Extended Inspection for HTTP Traffic feature: • Information About Configuring NBAR Using the MQC, page 2 • How to Configure NBAR Using the MQC, page 3
NBAR Real-Time Transport Protocol Payload Classification	12.2(15)T	Enables stateful identification of real-time audio and video traffic. The following section provides information about the NBAR Real-Time Transport Protocol Payload Classification feature: • Information About Configuring NBAR Using the MQC, page 2 • How to Configure NBAR Using the MQC, page 3

Table 2 Feature Information for Configuring NBAR Using the MQC (continued)

Feature Name	Releases	Feature Information
NBAR—Network-Based Application Recognition	12.2(18)ZYA	Integrates NBAR and Firewall Service Module (FWSM) functionality on the Catalyst 6500 series switch that is equipped with a Supervisor 32/programmable intelligent services accelerator (PISA). Additional protocols are now recognized by NBAR. The following sections provide information about the NBAR feature: • Information About Configuring NBAR Using the MQC, page 2 • How to Configure NBAR Using the MQC, page 3 The following command was modified: **match protocol (NBAR)**.
NBAR—Network-Based Application Recognition (Hardware Accelerated NBAR)	12.2(18)ZY	Enables NBAR functionality on the Catalyst 6500 series switch that is equipped with a Supervisor 32/programmable intelligent services accelerator (PISA). The following section provides information about the NBAR—Network-Based Application Recognition (Hardware Accelerated NBAR) feature: • Information About Configuring NBAR Using the MQC, page 2 • How to Configure NBAR Using the MQC, page 3

Congestion Management

Congestion Management Overview

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

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

This chapter discusses these four types of queueing, which constitute the congestion management QoS features:

- FIFO (first-in, first-out). FIFO entails no concept of priority or classes of traffic. With FIFO, transmission of packets out the interface occurs in the order the packets arrive.
- Weighted fair queueing (WFQ). WFQ offers dynamic, fair queueing that divides bandwidth across queues of traffic based on weights. (WFQ ensures that all traffic is treated fairly, given its weight.) To understand how WFQ works, consider the queue for a series of File Transfer Protocol (FTP) packets as a queue for the collective and the queue for discrete interactive traffic packets as a queue for the individual. Given the weight of the queues, WFQ ensures that for all FTP packets sent as a collective an equal number of individual interactive traffic packets are sent.)

  Given this handling, WFQ ensures satisfactory response time to critical applications, such as interactive, transaction-based applications, that are intolerant of performance degradation. For serial interfaces at E1 (2.048 Mbps) and below, flow-based WFQ is used by default. When no other queueing strategies are configured, all other interfaces use FIFO by default.

  There are four types of WFQ:

  - Flow-based WFQ (WFQ)
  - Distributed WFQ (DWFQ)

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

  - Class-based WFQ (CBWFQ)
  - Distributed class-based WFQ (DCBWFQ)
- Custom queueing (CQ). With CQ, bandwidth is allocated proportionally for each different class of traffic. CQ allows you to specify the number of bytes or packets to be drawn from the queue, which is especially useful on slow interfaces.
- Priority queueing (PQ). With PQ, packets belonging to one priority class of traffic are sent before all lower priority traffic to ensure timely delivery of those packets.

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

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

Why Use Congestion Management?

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

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

- Traffic prioritization is especially important for delay-sensitive, interactive transaction-based applications—for instance, desktop video conferencing—that require higher priority than do file transfer applications. However, use of WFQ ensures that all traffic is treated fairly, given its weight, and in a dynamic manner. For example, WFQ addresses the requirements of the interactive application without penalizing the FTP application.
- Prioritization is most effective on WAN links where the combination of bursty traffic and relatively lower data rates can cause temporary congestion.
- Depending on the average packet size, prioritization is most effective when applied to links at T1/E1 bandwidth speeds or lower.
- If users of applications running across your network notice poor response time, you should consider using congestion management features. Congestion management features are dynamic, tailoring themselves to the existing network conditions. However, consider that if a WAN link is constantly congested, traffic prioritization may *not* resolve the problem. Adding bandwidth might be the appropriate solution.
- If there is no congestion on the WAN link, there is no reason to implement traffic prioritization.

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

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

  After you assess your needs, review the available congestion management queueing mechanisms described in this chapter and determine which approach best addresses your requirements and goals.
- Configure the interface for the kind of queueing strategy you have chosen, and observe the results.

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

See the following section "Deciding Which Queueing Policy to Use" for elaboration of the differences among the various queueing mechanisms.

Deciding Which Queueing Policy to Use

This section looks briefly at some of the differences between the types of queueing and includes a table that compares the main queueing strategies.

FIFO queueing performs no prioritization of data packets on user data traffic. It entails no concept of priority or classes of traffic. When FIFO is used, ill-behaved sources can consume available bandwidth, bursty sources can cause delays in time-sensitive or important traffic, and important traffic may be dropped because less important traffic fills the queue.

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

- CQ guarantees some level of service to all traffic because you can allocate bandwidth to all classes of traffic. You can define the size of the queue by determining its configured packet-count capacity, thereby controlling bandwidth access.
- PQ guarantees strict priority in that it ensures that one type of traffic will be sent, possibly at the expense of all others. For PQ, a low priority queue can be detrimentally affected, and, in the worst case, never allowed to send its packets if a limited amount of bandwidth is available or if the transmission rate of critical traffic is high.

In deciding whether to use WFQ or one of the other two queueing types, consider these differences among WFQ and PQ and CQ:

- WFQ does not require configuration of access lists to determine the preferred traffic on a serial interface. Rather, the fair queue algorithm dynamically sorts traffic into messages that are part of a conversation.
- Low-volume, interactive traffic gets fair allocation of bandwidth with WFQ, as does high-volume traffic such as file transfers.
- Strict priority queueing can be accomplished with WFQ by using the IP RTP Priority, Frame Relay IP RTP Priority, low latency queueing (LLQ), distributed low latency queueing, low latency queueing for Frame Relay, or Frame Relay PVC Interface Priority Queueing features. Strict PQ allows delay-sensitive data such as voice to be dequeued and sent before packets in other queues are dequeued.

Table 1 compares the salient features of flow-based WFQ, CBWFQ and DCBWFQ, CQ, and PQ.

Table 1 Queueing Comparison

	Flow-Based WFQ	CBWFQ/DCBWFQ	CQ	PQ
Number of Queues	Configurable number of queues (256 user queues, by default)	One queue per class, up to 64 classes	16 user queues	4 queues
Kind of Service	• Ensures fairness among all traffic flows based on weights • Strict priority queueing is available through use of the IP RTP Priority or Frame Relay IP RTP Priority features	• Provides class bandwidth guarantee for user-defined traffic classes • Provides flow-based WFQ support for nonuser-defined traffic classes • Strict priority queueing is available through use of the IP RTP Priority, Frame Relay IP RTP Priority, LLQ, Distributed LLQ, and LLQ for Frame Relay features	• Round-robin service	• High priority queues are serviced first • Absolute prioritization; ensures critical traffic of highest priority through use of the Frame Relay PVC Interface Priority Queueing feature
Configuration	No configuration required	Requires configuration	Requires configuration	Requires configuration

FIFO Queueing

In its simplest form, FIFO queueing—also known as first-come, first-served (FCFS) queueing—involves buffering and forwarding of packets in the order of arrival.

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 sent out an interface in the order in which they arrive.

When FIFO is used, ill-behaved sources can consume all the bandwidth, bursty sources can cause delays in time-sensitive or important traffic, and important traffic can be dropped because less important traffic fills the queue.

When no other queueing strategies are configured, all interfaces except serial interfaces at E1 (2.048 Mbps) and below use FIFO by default. (Serial interfaces at E1 and below use WFQ by default.)

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

Weighted Fair Queueing

This section discusses the four types of WFQ described in the following sections:

- Flow-Based Weighted Fair Queueing
- Distributed Weighted Fair Queueing
- Class-Based Weighted Fair Queueing
- Distributed Class-Based Weighted Fair Queueing

This section also discusses the six related features described in the following sections:

- IP RTP Priority
- Frame Relay IP RTP Priority
- Frame Relay PVC Interface Priority Queueing
- Low Latency Queueing
- Distributed Low Latency Queueing
- Low Latency Queueing for Frame Relay

Table 2 summarizes the differences among WFQ, DWFQ, CBWFQ, and DCBWFQ.

Table 2 WFQ, DWFQ, CBWFQ, and DCBWFQ Comparison

WFQ	DWFQ	CBWFQ	DCBWFQ
Flow-based WFQ: • Weighted, when packets are classified; for example, Resource Reservation Protocol (RSVP) • Fair queued (FQ), when packets are not classified (for example, best-effort traffic)	Flow-based DWFQ: • FQ, not weighted Class-based DWFQ: • Weighted • QoS-group-based • Type of Service (ToS)-based	Class-based WFQ: • Weighted • Bandwidth allocation can be specified for a specific class of traffic	Class-based distributed WFQ: • Weighted • Bandwidth allocation can be specified for a specific class of traffic
Runs on standard Cisco IOS platforms	Runs on Versatile Interface Processor (VIP) (faster performance)	Runs on standard Cisco IOS platforms	Runs on VIP (faster performance)

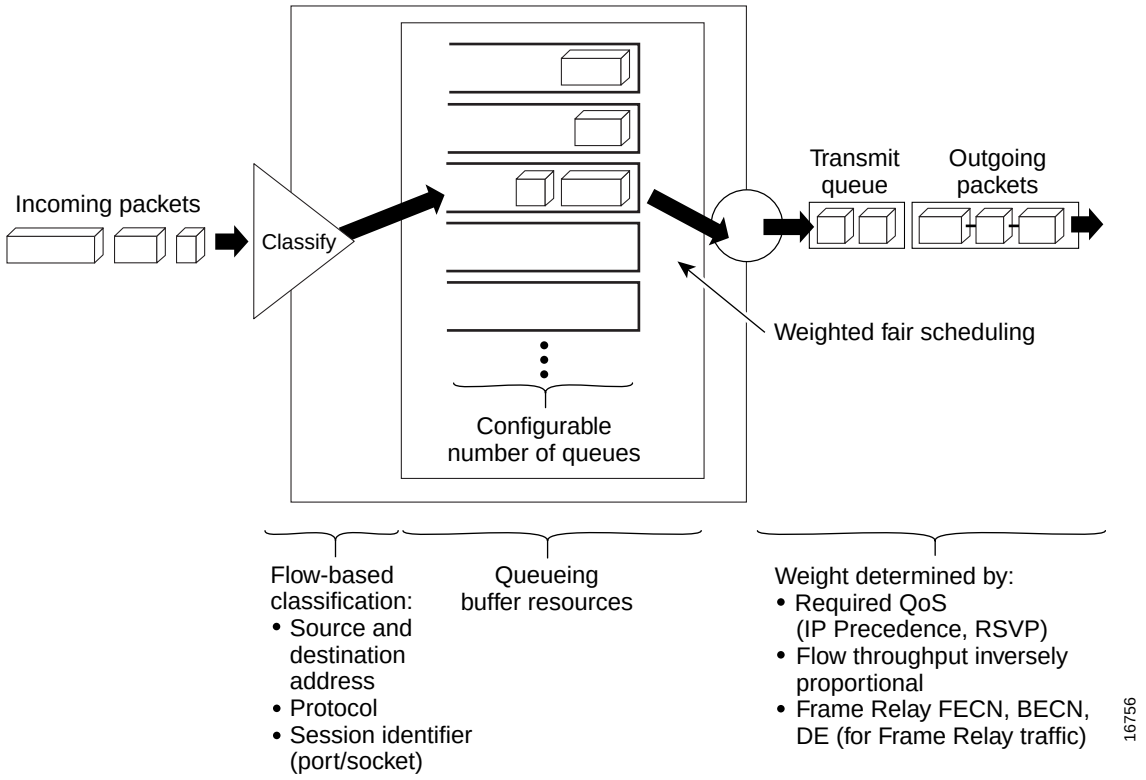
For DWFQ and DCBWFQ, all queueing is transacted by the VIP. On the VIP, all packets are sent directly out the interface. A Route Switch Processor (RSP) resides on the same platform as the VIP. The RSP handles all tasks associated with system maintenance and routing. The VIP and the RSP each handle some scheduling. The dual-processor support accounts for the faster speed of DWFQ and DCBWFQ over WFQ running on standard Cisco IOS platforms.

For information on how to configure WFQ, DWFQ, CBWFQ, and DCBWFQ, see the "Configuring Weighted Fair Queueing" module. For information on how to configure per-VC WFQ and CBWFQ, see the "Configuring IP to ATM Class of Service" module.

Flow-Based Weighted Fair Queueing

WFQ is a dynamic scheduling method that provides fair bandwidth allocation to all network traffic. WFQ applies priority, or weights, to identified traffic to classify traffic into conversations and determine how much bandwidth each conversation is allowed relative to other conversations. WFQ is a flow-based algorithm that simultaneously schedules interactive traffic to the front of a queue to reduce response time and fairly shares the remaining bandwidth among high-bandwidth flows. In other words, WFQ allows you to give low-volume traffic, such as Telnet sessions, priority over high-volume traffic, such as FTP sessions. WFQ gives concurrent file transfers balanced use of link capacity; that is, when multiple file transfers occur, the transfers are given comparable bandwidth. Figure 1 shows how WFQ works.

Figure 1 Weighted Fair Queueing



WFQ overcomes a serious limitation of FIFO queueing. When FIFO is in effect, traffic is sent in the order received without regard for bandwidth consumption or the associated delays. As a result, file transfers and other high-volume network applications often generate series of packets of associated data. These related packets are known as packet trains. Packet trains are groups of packets that tend to move together through the network. These packet trains can consume all available bandwidth, depriving other traffic of bandwidth.

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

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

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

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

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

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

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

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

Restrictions

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

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

WFQ and IP Precedence

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

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

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

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

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

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

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

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

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

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

WFQ and RSVP

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

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

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

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

Distributed Weighted Fair Queueing

DWFQ is a special high-speed version of WFQ that runs on the VIP. It is supported on the following routers with a VIP2-40 or greater interface processor:

- Cisco 7000 series with RSP7000
- Cisco 7500 series

A VIP2-50 interface processor is recommended when the aggregate line rate of the port adapters on the VIP is greater than DS3. A VIP2-50 card is required for OC-3 rates.

To use DWFQ, distributed Cisco Express Forwarding (dCEF) switching must be enabled on the interface. For more information on CEF, see the "Cisco Express Forwarding Features Roadmap" module.

Note The VIP-distributed WFQ implementation differs from WFQ that runs on all other platforms.

There are two forms of distributed WFQ:

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

  Each flow corresponds to a separate output queue. When a packet is assigned to a flow, it is placed in the queue for that flow. During periods of congestion, DWFQ allocates an equal share of the bandwidth to each active queue.

  Flow-based DWFQ is also called fair queueing because all flows are equally weighted and allocated equal bandwidth. In the current implementation of DWFQ, weights are not assigned to flows. With DWFQ, well-behaved hosts are protected from ill-behaved hosts.
- Class-based. In this form, packets are assigned to different queues based on their QoS group or the IP precedence in the ToS field.

QoS groups allow you to customize your QoS policy. A QoS group is an internal classification of packets used by the router to determine how packets are treated by certain QoS features, such as DWFQ and committed access rate (CAR). Use a CAR policy or the QoS Policy Propagation via Border Gateway Protocol (BGP) feature to assign packets to QoS groups.

If you want to classify packets based only on the two low-order IP Precedence bits, use ToS-based DWFQ. Specify a weight for each class. In periods of congestion, each group is allocated a percentage of the output bandwidth equal to the weight of the class. For example, if a class is assigned a weight of 50, packets from this class will be allocated at least 50 percent of the outgoing bandwidth during periods of congestion. When the interface is not congested, queues can use any available bandwidth.

The "Drop Policy" section describes the drop policy used by both forms.

Drop Policy

DWFQ keeps track of the number of packets in each queue and the total number of packets in all queues.

When the total number of packets is below the aggregate limit, queues can buffer more packets than the individual queue limit.

When the total number of packets reaches the aggregate limit, the interface starts enforcing the individual queue limits. Any new packets that arrive for a queue that has exceeded its individual queue limit are dropped. Packets that are already in the queue will not be dropped, even if the queue is over the individual limit.

In some cases, the total number of packets in all queues put together may exceed the aggregate limit.

Restrictions

Use DWFQ with IP traffic. All non-IP traffic is treated as a single flow and, therefore, placed in the same queue.

DWFQ has the following restrictions:

- Can be configured on interfaces, but not subinterfaces.
- Is not supported with the ATM encapsulations AAL5-MUX and AAL5-NLPID.
- Is not supported on Fast EtherChannel, tunnel interfaces, or other logical (virtual) interfaces such as Multilink PPP (MLP).
- Cannot be configured on the same interface as RSP-based PQ, CQ, or WFQ.

Class-Based Weighted Fair Queueing

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

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

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

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

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

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

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

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

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

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

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

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

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

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

- Attaching policies to interfaces (service policies).

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

CBWFQ Bandwidth Allocation

The sum of all bandwidth allocation on an interface cannot exceed 75 percent of the total available interface bandwidth. The remaining 25 percent is used for other overhead, including Layer 2 overhead, routing traffic, and best-effort traffic. Bandwidth for the CBWFQ class-default class, for instance, is taken from the remaining 25 percent. However, under aggressive circumstances in which you want to configure more than 75 percent of the interface bandwidth to classes, you can override the 75 percent

maximum sum allocated to all classes or flows using the **max-reserved-bandwidth** command. If you want to override the default 75 percent, exercise caution and ensure that you allow enough remaining bandwidth to support best-effort and control traffic, and Layer 2 overhead.

When ATM is used you must account for the fact that ATM cell tax overhead is not included. For example, consider the case where a class needs guaranteed bandwidth on an ATM permanent virtual circuit (PVC). Suppose the average packet size for the class is 256 bytes and the class needs 100 kbps (which translates to 49 packets per second) of guaranteed bandwidth. Each 256-byte packet would be split into six cells to be sent on a VC, giving a total of 6 * 53 = 318 bytes. In this case, the ATM cell tax overhead would be 62 bytes or 49 * 62 * 8 = 24.34 kbps. When configuring CBWFQ in this example, ensure that the sum of all the configured class bandwidths is less than the VC bandwidth by at least 24.34 kbps to ensure desired payload guarantee for the configured classes (in this example, there is only one class). If you have several classes, the sum of all the class overheads should be estimated and added to the sum of all the configured class bandwidths. This total should be less than the VC bandwidth to ensure the required payload guarantees.

Why Use CBWFQ?

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

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

CBWFQ and RSVP

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

Restrictions

Configuring CBWFQ on a physical interface is only possible if the interface is in the default queueing mode. Serial interfaces at E1 (2.048 Mbps) and below use WFQ by default—other interfaces use FIFO by default. Enabling CBWFQ on a physical interface overrides the default interface queueing method. Enabling CBWFQ on an ATM PVC does not override the default queueing method.

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

Traffic shaping and policing are not currently supported with CBWFQ.

CBWFQ is supported on variable bit rate (VBR) and available bit rate (ABR) ATM connections. It is not supported on unspecified bit rate (UBR) connections.

CBWFQ is not supported on Ethernet subinterfaces.

Distributed Class-Based Weighted Fair Queueing

As explained earlier, WFQ offers dynamic, fair queueing that divides bandwidth across queues of traffic based on weights. WFQ ensures that all traffic is treated fairly, given its weight. For more information about WFQ, see the "Weighted Fair Queueing" section of this module.

The DCBWFQ feature extends the standard WFQ functionality to provide support for user-defined traffic classes on the VIP. These user-defined traffic classes are configured in the Modular Quality of Service Command-Line Interface (Modular QoS CLI) feature. For information on how to configure QoS with the Modular QoS CLI, see the "Applying QoS Features Using the MQC" module.

The maximum number of packets allowed to accumulate in a traffic class queue is called the queue limit and is specified with the **queue-limit** command when you create a service policy with the **policy-map** command. Packets belonging to a traffic class are subject to the guaranteed bandwidth allocation and the queue limits that characterize the traffic class.

After a queue has reached its configured queue limit, enqueuing of additional packets to the traffic class causes tail drop or WRED drop to take effect, depending on how the service policy is configured. (Tail drop is a means of avoiding congestion that treats all traffic equally and does not differentiate between classes of service. Queues fill during periods of congestion. When the output queue is full and tail drop is in effect, packets are dropped until the congestion is eliminated and the queue is no longer full).

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

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

RSVP Interaction with DCBWFQ

When RSVP and DCBWFQ are configured, RSVP and DCBWFQ act independently of one another. RSVP and DCBWFQ allocate bandwidth among their traffic classes and flows according to unallocated bandwidth available at the underlying point of congestion.

When an RSVP flow is created, the VIP queueing system reserves the unit of bandwidth allocation in an RSVP queue, similar to the way a traffic class queue is allotted to a DCBWFQ traffic class. DCBWFQ traffic classes are unaffected by the RSVP flows.

Benefits

Bandwidth Allocation

DCBWFQ allows you to specify the amount of guaranteed bandwidth to be allocated for a traffic class. Taking into account available bandwidth on the interface, you can configure up to 64 traffic classes and control bandwidth allocation among them. If excess bandwidth is available, the excess bandwidth is divided among the traffic classes in proportion to their configured bandwidths.

Flow-based WFQ allocates bandwidth equally among all flows.

Coarser Granularity and Scalability

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

Restrictions

Using the bandwidth Command on VIP Default Traffic Class

On a VIP, all traffic that does not match a user-defined traffic class is classified as part of the default traffic class. The implicit bandwidth allocated to the default traffic class on a VIP is equal to the link bandwidth minus all of the user-defined bandwidth given to the user-defined traffic classes (with the **bandwidth** command). At least 1 percent of the link bandwidth is always reserved for the default traffic class.

Because the bandwidth of the default traffic class for a VIP is implicit (the default traffic class receives all remaining bandwidth not given to the user-defined traffic classes), the **bandwidth** command cannot be used with the default traffic class when you configure a VIP.

Using the match protocol Command on a VIP

Do not use the **match protocol** command to create a traffic class with a non-IP protocol as a match criterion. The VIP does not support matching of non-IP protocols.

PA-A3-8T1IMA Modules

DCBWFQ is not supported on Cisco 7500 series routers with PA-A3-8T1IMA modules.

Prerequisites

WFQ

Attaching a service policy to an interface disables WFQ on that interface if WFQ is configured for the interface. For this reason, you should ensure that WFQ is not enabled on such an interface.

For information on WFQ, see the "Configuring Weighted Fair Queueing" module.

ACLs

You can specify a numbered access list as the match criterion for any traffic class that you create. For this reason, you should know how to configure access lists.

Modular QoS CLI

You can configure DCBWFQ using the Modular QoS CLI.

For information on configuring QoS features with the Modular QoS CLI, see the "Applying QoS Features Using the MQC" module.

IP RTP Priority

The IP RTP Priority feature provides a strict priority queueing scheme for delay-sensitive data such as voice. Voice traffic can be identified by its Real-Time Transport Protocol (RTP) port numbers and classified into a priority queue configured by the **ip rtp priority** command. The result is that voice is serviced as strict priority in preference to other nonvoice traffic.

Note Although this section focuses mainly on voice traffic, IP RTP Priority is useful for any RTP traffic.

The IP RTP Priority feature extends and improves on the functionality offered by the **ip rtp reserve** command by allowing you to specify a range of UDP/RTP ports whose traffic is guaranteed strict priority service over any other queues or classes using the same output interface. Strict priority means that if packets exist in the priority queue, they are dequeued and before packets in other queues are dequeued. We recommend that you use the **ip rtp priority** command instead of the **ip rtp reserve** command for voice configurations.

The IP RTP Priority feature does not require that you know the port of a voice call. Rather, the feature gives you the ability to identify a range of ports whose traffic is put into the priority queue. Moreover, you can specify the entire voice port range—16384 to 32767—to ensure that all voice traffic is given strict priority service. IP RTP Priority is especially useful on links whose speed is less than 1.544 Mbps.

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

- When used in conjunction with WFQ, the **ip rtp priority** command provides strict priority to voice, and WFQ scheduling is applied to the remaining queues.
- When used in conjunction with CBWFQ, the **ip rtp priority** command provides strict priority to voice. CBWFQ can be used to set up classes for other types of traffic (such as SNA) that needs dedicated bandwidth and needs to be treated better than best effort and not as strict priority; the nonvoice traffic is serviced fairly based on the weights assigned to the enqueued packets. CBWFQ can also support flow-based WFQ within the default CBWFQ class if so configured.

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

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

IP RTP Priority Bandwidth Allocation

If you want to understand its behavior and properly use the IP RTP Priority feature, it is important to consider its admission control and policing characteristics. When you use the **ip rtp priority** command to configure the priority queue for voice, you specify a strict bandwidth limitation. This amount of bandwidth is guaranteed to voice traffic enqueued in the priority queue. (This is the case whether you use the IP RTP Priority feature with CBWFQ or WFQ.)

Note IP RTP Priority does not have per-call admission control. The admission control is on an aggregate basis. For example, if configured for 96 kbps, IP RTP Priority guarantees that 96 kbps is available for reservation. It does not ensure that only four calls of 24 kbps are admitted. A fifth call of 24 kbps could be admitted, but because the five calls will only get 96 kbps, the call quality will be deteriorated. (Each call would get 96/5 = 19.2 kbps.) In this example, it is the responsibility of the user to ensure that only four calls are placed at one time.

IP RTP Priority closely polices use of bandwidth for the priority queue, ensuring that the allocated amount is not exceeded in the event of congestion. In fact, IP RTP Priority polices the flow every second. IP RTP Priority prohibits transmission of additional packets once the allocated bandwidth is consumed. If it discovers that the configured amount of bandwidth is exceeded, IP RTP Priority drops packets, an event that is poorly tolerated by voice traffic. (Enable debugging to watch for this condition.) Close policing allows for fair treatment of other data packets enqueued in other CBWFQ or WFQ queues. To avoid packet drop, be certain to allocate to the priority queue the most optimum amount of bandwidth, taking into consideration the type of codec used and interface characteristics. IP RTP Priority will not allow traffic beyond the allocated amount.

It is always safest to allocate to the priority queue slightly more than the known required amount of bandwidth. For example, suppose you allocated 24 kbps bandwidth, the standard amount required for voice transmission, to the priority queue. This allocation seems safe because transmission of voice packets occurs at a constant bit rate. However, because the network and the router or switch can use some of the bandwidth and introduce jitter and delay, allocating slightly more than the required amount of bandwidth (such as 25 kbps) ensures constancy and availability.

The IP RTP Priority admission control policy takes RTP header compression into account. Therefore, while configuring the *bandwidth* parameter of the **ip rtp priority** command you only need to configure for the bandwidth of the compressed call. For example, if a G.729 voice call requires 24 kbps uncompressed bandwidth (not including Layer 2 payload) but only 12 kbps compressed bandwidth, you only need to configure a bandwidth of 12 kbps. You need to allocate enough bandwidth for all calls if there will be more than one call.

The sum of all bandwidth allocation for voice and data flows on the interface cannot exceed 75 percent of the total available bandwidth. Bandwidth allocation for voice packets takes into account the payload plus the IP, RTP, and UDP headers, but again, not the Layer 2 header. Allowing 25 percent bandwidth for other overhead is conservative and safe. On a PPP link, for instance, overhead for Layer 2 headers assumes 4 kbps. The amount of configurable bandwidth for IP RTP Priority can be changed using the **max-reserved-bandwidth** command on the interface.

If you know how much bandwidth is required for additional overhead on a link, under aggressive circumstances in which you want to give voice traffic as much bandwidth as possible, you can override the 75 percent maximum allocation for the bandwidth sum allocated to all classes or flows by using the **max-reserved-bandwidth** command. If you want to override the fixed amount of bandwidth, exercise caution and ensure that you allow enough remaining bandwidth to support best-effort and control traffic, and Layer 2 overhead.

As another alternative, if the importance of voice traffic far exceeds that of data, you can allocate most of the 75 percent bandwidth used for flows and classes to the voice priority queue. Unused bandwidth at any given point will be made available to the other flows or classes.

Restrictions

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

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

Frame Relay IP RTP Priority

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

This feature extends the functionality offered by the **ip rtp priority** command by supporting Frame Relay PVCs. This feature allows you to specify a range of UDP ports whose voice traffic is guaranteed strict priority service over any other queues or classes using the same output interface. Strict priority means that if packets exist in the priority queue, they are dequeued and sent before packets in other queues are dequeued. This process is performed on a per-PVC basis, rather than at the interface level.

For information on how to configure Frame Relay IP RTP Priority, see the "Configuring Weighted Fair Queueing" module.

Frame Relay PVC Interface Priority Queueing

The Frame Relay PVC Interface Priority Queueing (PIPQ) feature provides an interface-level priority queueing scheme in which prioritization is based on destination PVC rather than packet contents. For example, Frame Relay (FR) PIPQ allows you to configure a PVC transporting voice traffic to have absolute priority over a PVC transporting signalling traffic, and a PVC transporting signalling traffic to have absolute priority over a PVC transporting data.

For information on how to configure Frame Relay PIPQ, see the "Configuring Weighted Fair Queueing" module. For information about Frame Relay, see the "Configuring Frame Relay" module.

Frame Relay PIPQ provides four levels of priority: high, medium, normal, and low. The Frame Relay packet is examined at the interface for the data-link connection identifier (DLCI) value. The packet is then sent to the correct priority queue based on the priority level configured for that DLCI.

Note When using Frame Relay PIPQ, configure the network so that different types of traffic are transported on separate PVCs. Frame Relay PIPQ is not meant to be used when an individual PVC carries different traffic types that have different QoS requirements.

You assign priority to a PVC within a Frame Relay map class. All PVCs using or inheriting that map class will be classed according to the configured priority. If a PVC does not have a map class associated with it, or if the map class associated with it does not have priority explicitly configured, then the packets on that PVC will be queued on the default "normal" priority queue.

If you do not enable Frame Relay PIPQ on the interface using the **frame-relay interface-queue priority** command in interface configuration mode, configuring PVC priority within a map class will not be effective. At this time you have the option to also set the size (in maximum number of packets) of the four priority queues.

Frame Relay PIPQ works with or without Frame Relay Traffic Shaping (FRTS) and FRF.12 (or higher). The interface-level priority queueing takes the place of the FIFO queueing or dual FIFO queueing normally used by FRTS and FRF.12 (or higher). PVC priority assigned within FR PIPQ takes precedence over FRF.12 priority, which means that all packets destined for the same PVC will be queued on the same interface queue whether they were fragmented or not.

Note Although high priority PVCs most likely will transport only small packets of voice traffic, you may want to configure FRF.12 (or higher) on these PVCs anyway to guard against any unexpectedly large packets.

Restrictions

The following restrictions apply to Frame Relay PIPQ:

- It is not supported on loopback or tunnel interfaces, or interfaces that explicitly disallow priority queueing.
- It is not supported with hardware compression.
- It cannot be enabled on an interface that is already configured with queueing other than FIFO queueing. FR PIPQ can be enabled if WFQ is configured, as long as WFQ is the default interface queueing method.

Prerequisites

The following prerequisites apply to Frame Relay PIPQ:

- PVCs should be configured to carry a single type of traffic.
- The network should be configured with adequate call admission control to prevent starvation of any of the priority queues.

Low Latency Queueing

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

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

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

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

One of the ways in which the strict PQ used within CBWFQ differs from its use outside CBWFQ is in the parameters it takes. Outside CBWFQ, you can use the **ip rtp priority** command to specify the range of UDP ports whose voice traffic flows are to be given priority service. Using the **priority** command, you are no longer limited to a UDP port number to stipulate priority flows because you can configure

the priority status for a class within CBWFQ. Instead, all of the valid match criteria used to specify traffic for a class now apply to priority traffic. These methods of specifying traffic for a class include matching on access lists, protocols, and input interfaces. Moreover, within an access list you can specify that traffic matches are allowed based on the IP differentiated services code point (DSCP) value that is set using the first six bits of the ToS byte in the IP header.

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

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

LLQ Bandwidth Allocation

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

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

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

Priority traffic metering has the following qualities:

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

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

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

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

Consider this case that uses ATM. Suppose a voice stream of 60 bytes emitting 50 packets per second is encoded using G.729. Prior to converting the voice stream to cells, the meter for the priority queue used for the voice stream assesses the length of the packet after the Layer 2 Logical Link Control (LLC) headers have been added.

Given the 8-byte Layer 2 LLC header, the meter will take into account a 68-byte packet. Because ATM cells are a standard 53 bytes long, before the 68-byte packet is emitted on the line, it is divided into two 53-byte ATM cells. Thus, the bandwidth consumed by this flow is 106 bytes per packet.

For this case, then, you must configure the bandwidth to be at least 27.2 kbps (68 * 50 * 8 = 27.2 kbps). However, recall that you must also allow for the ATM cell tax overhead, which is not accounted for by the configured bandwidth. In other words, the sum of the bandwidths for all classes must be less than the interface bandwidth by at least 15.2 kbps ([106 – 68] * 50 * 8 = 15.2 kbps). You should also remember to allow bandwidth for router-introduced jitter.

Note The sum of all bandwidth allocation on an interface cannot exceed 75 percent of the total available interface bandwidth. However, under aggressive circumstances in which you want to configure more than 75 percent of the interface bandwidth to classes, you can override the 75 percent maximum sum allocated to all classes or flows using the **max-reserved-bandwidth** command. The **max-reserved-bandwidth** command is intended for use on main interfaces only; it has no effect on virtual circuits (VCs) or ATM permanent virtual circuits (PVCs).

LLQ with IP RTP Priority

LLQ and IP RTP Priority can be configured at the same time, but IP RTP Priority takes precedence. To demonstrate how they work together, consider the following configuration:

```
policy-map llqpolicy
 class voice
 priority 50

ip rtp priority 16384 20000 40
service-policy output llqpolicy
```

In this example, packets that match the 16384 to 20000 port range will be given priority with 40 kbps bandwidth; packets that match the voice class will be given priority with 50 kbps bandwidth. In the event of congestion, packets that match the 16384 to 20000 port range will receive no more than 40 kbps of bandwidth, and packets that match the voice class will receive no more than 50 kbps of bandwidth.

If packets match both criteria (ports 16384 to 20000 and class voice), IP RTP Priority takes precedence. In this example, the packets will be considered to match the 16384 to 20000 port range and will be accounted for in the 40 kbps bandwidth.

LLQ and Committed Burst Size

The functionality of LLQ has been extended to allow you to specify the Committed Burst (Bc) size in LLQ. This functionality is provided with the Configuring Burst Size in Low Latency Queueing feature. With this new functionality, the network can now accommodate temporary bursts of traffic and handle network traffic more efficiently.

Note The default Bc size used by LLQ is intended to handle voice-like non-bursty traffic. If you want to configure LLQ to handle the traffic of non-voice applications, you may need to increase the burst size accordingly, based on the application in use on your network.

LLQ and per-VC Hold Queue Support for ATM Adapters

By default, the queueing mechanism in use determines the size of the hold queue, and, therefore, the number of packets contained in the queue. The Configurable per-VC Hold Queue Support for ATM Adapters feature allows you to expand the default hold queue size and change (or vary) the number of packets the queue can contain. With this new feature, the hold queue can contain a maximum of 1024 packets.

This feature allows you to specify the number of packets contained in the hold queue, per VC, on ATM adapters that support per-VC queueing.

Note This feature is supported only on the Cisco 7200 series routers, and on Cisco 2600 and 3600 series adapters that support per-VC queueing.

For related information about per-VC and ATM configurations, see the "IP to ATM Class of Service Overview" module and the "Configuring IP to ATM Class of Service" module.

Why Use LLQ?

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

- LLQ provides strict priority service on ATM VCs and serial interfaces. (The IP RTP Priority feature allows priority queueing only on interfaces.)
- LLQ is not limited to UDP port numbers. Because you can configure the priority status for a class within CBWFQ, you are no longer limited to UDP port numbers to stipulate priority flows. Instead, all of the valid match criteria used to specify traffic for a class now apply to priority traffic.
- By configuring the maximum amount of bandwidth allocated for packets belonging to a class, you can avoid starving nonpriority traffic.

Restrictions

The following restrictions apply to LLQ:

- If you use access lists to configure matching port numbers, this feature provides priority matching for all port numbers, both odd and even. Because voice typically exists on even port numbers, and control packets are generated on odd port numbers, control packets are also given priority when using this feature. On very slow links, giving priority to both voice and control packets may produce degraded voice quality. Therefore, if you are only assigning priority based on port numbers, you should use the **ip rtp priority** command instead of the **priority** command. (The **ip rtp priority** command provides priority only for even port numbers.)
- The **random-detect** command, **queue-limit** command, and **bandwidth** policy-map class configuration command cannot be used while the **priority** command is configured.
- The **priority** command can be configured in multiple classes, but it should only be used for voice-like, constant bit rate (CBR) traffic.

Distributed Low Latency Queueing

The Distributed LLQ feature provides the ability to specify low latency behavior for a traffic class on a VIP-based Cisco 7500 series router except one with a PA-A3-8T1IMA module. LLQ allows delay-sensitive data such as voice to be dequeued and sent before packets in other queues are dequeued.

The Distributed LLQ feature also introduces the ability to limit the depth of a device transmission ring. Before the introduction of Distributed LLQ, the maximum transmission ring depth was not a user-configurable parameter. Therefore, particles could accumulate on a transmission ring without limitation, which could result in unavoidable high latencies. The Distributed LLQ feature allows users to limit the number of particles that may exist on a transmission ring, effectively lowering the latency incurred by packets sitting on that transmission ring.

The **priority** command is used to allow delay-sensitive data to be dequeued and sent first. LLQ enables use of a single priority queue within which individual classes of traffic can be placed. To enqueue class traffic to the priority queue, you configure the **priority** command for the class after you specify the named class within a policy map. The amount of bandwidth available for the priority queue can be specified either as a set amount of bandwidth in kbps or as a percentage of all available bandwidth (beginning in Cisco IOS Release 12.1(5)T).

Within a policy map, you can give one or more classes priority status. When multiple classes within a single policy map are configured as priority classes, all traffic from these classes is enqueued to the same, single, priority queue.

The **tx-ring-limit** command allows the user to specify the number of allowable particles on a transmission ring, effectively lowering the latency for that transmission ring. One packet can contain multiple particles, and a typical particle is 512 bytes in size (the size depends on the interface types. For some interface types, a typical particle size is 256 bytes.) These particles can no longer accumulate on a transmission ring and cause unavoidable high latencies.

Distributed LLQ is supported on the Cisco 7500 RSP series router with a VIP except when a PA-A3-8T 1IMA module is configured.

This feature also supports the *Class-Based Quality of Service* MIB.

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

Guaranteeing Bandwidth with the priority Command

One method of using the **priority** command for a traffic class is to specify a *bandwidth* argument that gives the maximum bandwidth in kpbs. The other method of using the **priority** command for a traffic class, which was introduced in Cisco IOS Release 12.1(5)T, is to specify a percentage of available bandwidth to be reserved for the priority queue. The *bandwidth* value or percentage guarantees the configured bandwidth to the priority class under worst-case congestion scenarios. If excess bandwidth is available, the priority class will be allowed to utilize the bandwidth. If no excess bandwidth is available, the priority traffic will be constrained to the configured rate via packet drops. Each individual class that is configured to a bandwidth value will have its traffic constrained to its individual rate. When a class is constrained to its individual rate, the traffic is permitted a certain amount of burstiness because of the token bucket mechanism policing the stream. This amount of burstiness is controlled by the optional *burst* parameter in the **priority** command (this burstiness cannot be specified when specifying a priority queue based on a percentage of available bandwidth). The *burst* parameter specifies, in bytes, the amount of traffic allowed to pass through the token bucket as a one-time burst in excess of the token bucket drop parameters. The default burst value is 200 milliseconds of traffic at the configured token bucket drop parameters.

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

The bandwidth allocated for a priority queue always includes the Layer 2 encapsulation header. However, it does not include other headers, such as ATM cell tax overheads. When you calculate the amount of bandwidth to allocate for a given priority class, you must account for the fact that the Layer

2 headers are included. When ATM is used, you must account for the fact that ATM cell tax overhead is not included. You must also allow bandwidth for the possibility of jitter introduced by routers in the voice path.

Consider this case that uses ATM: Suppose a voice stream of 60 bytes emitting 50 packets per second is encoded using G.729. Prior to converting the voice stream to cells, the meter for the priority queue used for the voice stream assesses the length of the packet after the Layer logical link control (LLC) headers have been added.

Given the 8-byte Layer 2 LLC header, the meter will take into account a 68-byte packet. Because ATM cells are a standard 53 bytes long, before the 68-kbps packet is emitted on the line, it is divided into two 53-byte ATM cells. Thus, the bandwidth consumed by this flow is 106 bytes per packet.

For this case, then, you must configure the bandwidth to be at least 27.2 kbps (68 * 50 * 8 = 27.2 kbps). However, recall that you must also allow for the ATM cell tax overhead, which is not accounted for by the configured bandwidth. In other words, the sum of the bandwidths for all classes must be less than the interface bandwidth by at least 15.2 kbps ([106 – 68] * 50 * 8 = 15.2 kbps). You should also remember to allow bandwidth for router-introduced jitter.

Benefits

Provides Priority Service on ATM VCs and Serial Interface

The PQ scheme allows delay-sensitive data such as voice to be dequeued and sent before packets in other queues are dequeued. This feature provides PQ on ATM VCs.

Admission Control

By configuring the maximum amount of bandwidth allocated for packets belonging to a class, you can avoid starving nonpriority traffic.

Limiting Particles on a Transmission Ring

The Distributed LLQ feature also introduces particle limiting for transmission rings. Before the introduction of Distributed LLQ, the transmission ring depth was not user-configurable. Therefore, a user could experience unavoidable high latencies on a transmission ring.

The Distributed LLQ feature allows users to limit the number of particles on a transmission ring to a predefined limit, effectively lowering the latency on transmission rings.

Restrictions

The following restrictions apply to the Distributed LLQ feature:

- If you use access lists to configure matching port numbers, this feature provides priority matching for all port numbers. Because voice typically exists on even port numbers, and control packets are generated on odd port numbers, control packets are also given priority when using this feature. On very slow links, giving priority to both voice and control packets may produce degraded voice quality.
- The **priority** command can be used in conjunction with the **set** command. The **priority** command cannot be used in conjunction with any other command, including the **random-detect**, **queue-limit**, and **bandwidth** commands.
- The **priority** command can be configured in multiple traffic classes. If the traffic is not CBR traffic, you must configure a large enough *bandwidth-kbps* parameter to absorb the data bursts.

- Because 1 percent of the available bandwidth is reserved for the default traffic class, the sum of the percentage for the **bandwidth percent** and **priority percent** command reservations cannot exceed 99 percent.
- Priority queues can be reserved by either size or percentage values, but not both, in the same policy map. Therefore, if the **priority** command is used without the **percent** option in a policy map, the **bandwidth** command, if used, must also be used without the **percent** option, and vice versa. Similarly, if the **priority percent** command is used in a policy map, the **bandwidth percent** command must be used to specify bandwidth allocation for the class, and vice versa. The **priority** and **priority percent** commands also cannot be used in the same policy map.
- The **bandwidth** and **priority** commands cannot be used in the same class map. These commands can be used together in the same policy map, however.

  The following commands cannot be used in the same class or policy map with the **priority** command:

  - **priority percent**
  - **bandwidth percent**

  The following commands cannot be used in the same class or policy map with the **priority percentage** command:

  - **priority** (without the **percent** option)
  - **bandwidth** (without the **percent** option)
- The **tx-ring-limit** command can only affect a VBR VC on a PA-A3 port adapter. The **tx-ring-limit** command does not affect UBR VCs.
- DLLQ is not supported on Cisco 7500 series routers with PA-A3-8T1IMA modules.

Prerequisites

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

- ACLs
- ATM PVCs
- Bandwidth management
- CBWFQ
- LFI
- Virtual templates and virtual access interfaces

Low Latency Queueing for Frame Relay

LLQ for Frame Relay provides a strict priority queue for voice traffic and weighted fair queues for other classes of traffic. With this feature, LLQ is available at the Frame Relay VC level when FRTS is configured.

LLQ, also called PQ/CBWFQ, is a superset of and more flexible than previous Frame Relay QoS offerings, in particular RTP prioritization and PQ/WFQ.

With RTP prioritization and PQ/WFQ, traffic that matches a specified UDP/RTP port range is considered high priority and allocated to the priority queue (PQ). With LLQ for Frame Relay, you set up classes of traffic according to protocol, interface, or access lists, and then define policy maps to establish how the classes are handled in the priority queue and weighted fair queues.

Queues are set up on a per-PVC basis: each PVC has a PQ and an assigned number of fair queues. The fair queues are assigned weights proportional to the bandwidth requirements of each class; a class requiring twice the bandwidth of another will have half the weight. Oversubscription of the bandwidth is not permitted. The CLI will reject a change of configuration that would cause the total bandwidth to be exceeded. This functionality differs from that of WFQ, in which flows are assigned a weight based on IP precedence. WFQ allows higher precedence traffic to obtain proportionately more of the bandwidth, but the more flows there are, the less bandwidth is available to each flow.

The PQ is policed to ensure that the fair queues are not starved of bandwidth. When you configure the PQ, you specify in kbps the maximum amount of bandwidth available to that queue. Packets that exceed that maximum are dropped. There is no policing of the fair queues.

LLQ for Frame Relay is configured using a combination of **class-map**, **policy-map**, and Frame Relay map class commands. The **class-map** command defines traffic classes according to protocol, interface, or access list. The **policy-map** command defines how each class is treated in the queueing system according to bandwidth, priority, queue limit, or WRED. The **service-policy output** map class command attaches a policy map to a Frame Relay VC.

Policies not directly related to LLQ—for example, traffic shaping, setting IP precedence, and policing—are not supported by the **class-map** and **policy-map** commands for Frame Relay VCs. You must use other configuration mechanisms, such as map class commands, to configure these policies.

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

Restrictions

Only the following class map and policy map commands are supported:

- The **match** class-map configuration command
- The **priority**, **bandwidth**, **queue-limit**, **random-detect**, and **fair-queue** policy-map configuration commands

Prerequisites

The following tasks must be completed before LLQ for Frame Relay can be enabled:

- FRTS must be enabled on the interface.
- An output service policy must be configured in the map class associated with the interface, subinterface, or DLCI.
- Any queue other than a FIFO queue that is configured in the map class must be removed. LLQ for Frame Relay cannot be configured if there is already a non-FIFO queue configured, except for the default queue that is created when fragmentation is enabled.

How It Works

LLQ for Frame Relay is used in conjunction with the features described in the following sections:

- RTP Prioritization
- Voice over Frame Relay
- Frame Relay Fragmentation
- IP Cisco Express Forwarding Switching

RTP Prioritization

RTP prioritization provides a strict PQ scheme for voice traffic. Voice traffic is identified by its RTP port numbers and classified into a priority queue configured by the **frame-relay ip rtp priority** map-class configuration command. You classify traffic as voice by specifying an RTP port number range. If traffic matches the specified range, it is classified as voice and queued in the LLQ PQ, and the interface priority queue. If traffic does not fall within the specified RTP port range, it is classified by the service policy of the LLQ scheme.

The **ip rtp priority** command is available in both interface configuration mode and map-class configuration mode. Only the **frame relay ip rtp priority** map-class configuration command is supported in this feature.

Voice over Frame Relay

Voice over Frame Relay (VoFR) uses the LLQ priority queue (PQ) rather than its own PQ mechanism. The **frame-relay voice bandwidth** map-class configuration command configures the total bandwidth available for VoFR traffic. The visible bandwidth made available to the other queues will be the minimum committed information rate (CIR) minus the voice bandwidth.

The **frame-relay voice bandwidth** map-class configuration command also configures a call admission control function, which ensures that sufficient VoFR bandwidth remains before allowing a call. There is no policing of the voice traffic once the call has been established.

For VoFR with no data, all voice and call control packets are queued in the LLQ priority queueing (PQ). For VoFR with data, a VoFR PVC may carry both voice and data packets in different subchannels. VoFR data packets are fragmented and interleaved with voice packets to ensure good latency bounds for voice packets and scalability for voice and data traffic.

Note that when VoFR is enabled, there is no need to configure a priority class map for voice. The only VoFR commands to be used with LLQ for Frame Relay are the **frame-relay voice bandwidth** map-class configuration command and the **vofr data** Frame Relay DLCI configuration command.

Note It is possible—though not recommended—to configure other traffic for the PQ at the same time as VoFR. Doing so could cause delays because interleaving non-VoFR packets in the PQ would not be possible, causing the PQ (and any VoFR packets on it) to be held up during fragmentation until the entire fragmented packet has been sent.

Frame Relay Fragmentation

The purpose of Frame Relay fragmentation (FRF.12) is to support voice and data packets on lower-speed links without causing excessive delay to the voice packets. Large data packets are fragmented and interleaved with the voice packets.

When FRF.12 is configured with LLQ, small packets classified for the PQ pass through unfragmented onto both the LLQ PQ and the high priority interface queue. Large packets destined for PQ are shaped and fragmented when dequeued.

Use the **frame-relay fragment** and **service-policy** map-class configuration commands to enable LLQ with FRF.12.

IP Cisco Express Forwarding Switching

IP CEF switching is not affected by LLQ functionality.

Custom Queueing

CQ allows you to specify a certain number of bytes to forward from a queue each time the queue is serviced, thereby allowing you to share the network resources among applications with specific minimum bandwidth or latency requirements. You can also specify a maximum number of packets in each queue.

For information on how to configure CQ, see the "Configuring Custom Queueing" module.

How It Works

CQ handles traffic by specifying the number of packets or bytes to be serviced for each class of traffic. It services the queues by cycling through them in round-robin fashion, sending the portion of allocated bandwidth for each queue before moving to the next queue. If one queue is empty, the router will send packets from the next queue that has packets ready to send.

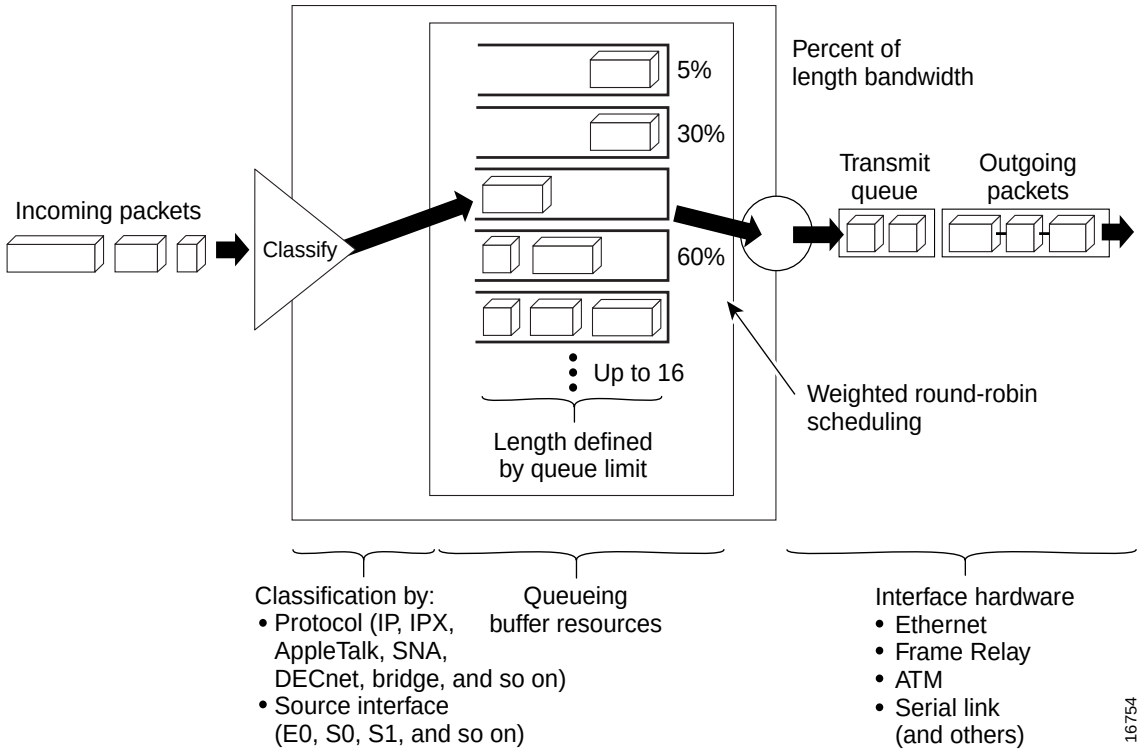
When CQ is enabled on an interface, the system maintains 17 output queues for that interface. You can specify queues 1 through 16. Associated with each output queue is a configurable byte count, which specifies how many bytes of data the system should deliver from the current queue before it moves on to the next queue.

Queue number 0 is a system queue; it is emptied before any of the queues numbered 1 through 16 are processed. The system queues high priority packets, such as keepalive packets and signalling packets, to this queue. Other traffic cannot be configured to use this queue.

For queue numbers 1 through 16, the system cycles through the queues sequentially (in a round-robin fashion), dequeueing the configured byte count from each queue in each cycle, delivering packets in the current queue before moving on to the next one. When a particular queue is beingprocessed, packets are sent until the number of bytes sent exceeds the queue byte count or the queue is empty. Bandwidth used by a particular queue can be indirectly specified only in terms of byte count and queue length.

Figure 2 shows how CQ behaves.

Figure 2 Custom Queueing



CQ ensures that no application or specified group of applications achieves more than a predetermined proportion of overall capacity when the line is under stress. Like PQ, CQ is statically configured and does not automatically adapt to changing network conditions.

On most platforms, all protocols are classified in the fast-switching path.

Determining Byte Count Values for Queues

In order to allocate bandwidth to different queues, you must specify the byte count for each queue.

How the Byte Count Is Used

The router sends packets from a particular queue until the byte count is exceeded. Once the byte count value is exceeded, the packet that is currently being sent will be completely sent. Therefore, if you set the byte count to 100 bytes and the packet size of your protocol is 1024 bytes, then every time this queue is serviced, 1024 bytes will be sent, not 100 bytes.

For example, suppose one protocol has 500-byte packets, another has 300-byte packets, and a third has 100-byte packets. If you want to split the bandwidth evenly across all three protocols, you might choose to specify byte counts of 200, 200, and 200 for each queue. However, this configuration does not result in a 33/33/33 ratio. When the router services the first queue, it sends a single 500-byte packet; when it services the second queue, it sends a 300-byte packet; and when it services the third queue, it sends two 100-byte packets. The effective ratio is 50/30/20.

Thus, setting the byte count too low can result in an unintended bandwidth allocation.

However, very large byte counts will produce a "jerky" distribution. That is, if you assign 10 KB, 10 KB, and 10 KB to three queues in the example given, each protocol is serviced promptly when its queue is the one being serviced, but it may be a long time before the queue is serviced again. A better solution is to specify 500-byte, 600-byte, and 500-byte counts for the queue. This configuration results in a ratio of 31/38/31, which may be acceptable.

In order to service queues in a timely manner and ensure that the configured bandwidth allocation is as close as possible to the required bandwidth allocation, you must determine the byte count based on the packet size of each protocol, otherwise your percentages may not match what you configure.

Note CQ was modified in Cisco IOS Release 12.1. When the queue is depleted early, or the last packet from the queue does not exactly match the configured byte count, the amount of deficit is remembered and accounted for the next time the queue is serviced. Beginning with Cisco IOS Release 12.1, you need not be as accurate in specifying byte counts as you did when using earlier Cisco IOS releases that did not take deficit into account.

Note Some protocols, such as Internetwork Packet Exchange (IPX), will negotiate the frame size at session startup time.

Determining the Byte Count

To determine the correct byte counts, perform the following steps:

Step 1 For each queue, divide the percentage of bandwidth you want to allocate to the queue by the packet size, in bytes. For example, assume the packet size for protocol A is 1086 bytes, protocol B is 291 bytes, and protocol C is 831 bytes. We want to allocate 20 percent for A, 60 percent for B, and 20 percent for C. The ratios would be:

20/1086, 60/291, 20/831 or

0.01842, 0.20619, 0.02407

Step 2 Normalize the numbers by dividing by the lowest number:

1, 11.2, 1.3

The result is the ratio of the number of packets that must be sent so that the percentage of bandwidth that each protocol uses is approximately 20, 60, and 20 percent.

Step 3 A fraction in any of the ratio values means that an additional packet will be sent. Round up the numbers to the next whole number to obtain the actual packet count.

In this example, the actual ratio will be 1 packet, 12 packets, and 2 packets.

Step 4 Convert the packet number ratio into byte counts by multiplying each packet count by the corresponding packet size.

In this example, the number of packets sent is one 1086-byte packet, twelve 291-byte packets, and two 831-byte packets, or 1086, 3492, and 1662 bytes, respectively, from each queue. These are the byte counts you would specify in your CQ configuration.

Step 5 To determine the bandwidth distribution this ratio represents, first determine the total number of bytes sent after all three queues are serviced:

(1 * 1086) + (12 * 291) + (2 * 831) = 1086 + 3492 + 1662 = 6240

Step 6 Then determine the percentage of the total number of bytes sent from each queue:

1086/6240, 3492/6240, 1662/6240 = 17.4, 56, and 26.6 percent

This result is close to the desired ratio of 20/60/20.

Step 7 If the actual bandwidth is not close enough to the desired bandwidth, multiply the original ratio of 1:11.2:1.3 by the best value, trying to get as close to three integer values as possible. Note that the multiplier you use need not be an integer. For example, if we multiply the ratio by two, we get 2:22.4:2.6. We would now send two 1086-byte packets, twenty-three 291-byte packets, and three 831-byte packets, or 2172/6693/2493, for a total of 11,358 bytes. The resulting ratio is 19/59/22 percent, which is much closer to the desired ratio that we achieved.

The bandwidth that a custom queue will receive is given by the following formula:

```
(queue byte count / total byte count of all queues) * bandwidth capacity of the interface
```

where bandwidth capacity is equal to the interface bandwidth minus the bandwidth for priority queues.

Window Size

Window size also affects the bandwidth distribution. If the window size of a particular protocol is set to one, then that protocol will not place another packet into the queue until it receives an acknowledgment. The CQ algorithm moves to the next queue if the byte count is exceeded or no packets are in that queue.

Therefore, with a window size of one, only one frame will be sent each time. If your frame count is set to 2 kilobytes, and your frame size is 256 bytes, then only 256 bytes will be sent each time this queue is serviced.

Why Use CQ?

You can use the Cisco IOS QoS CQ feature to provide specific traffic guaranteed bandwidth at a potential congestion point, assuring the traffic a fixed portion of available bandwidth and leaving the remaining bandwidth to other traffic. For example, you could reserve half of the bandwidth for SNA data, allowing the remaining half to be used by other protocols.

If a particular type of traffic is not using the bandwidth reserved for it, then unused bandwidth can be dynamically allocated to other traffic types.

Restrictions

CQ is statically configured and does not adapt to changing network conditions. With CQ enabled, the system takes longer to switch packets than FIFO because the packets are classified by the processor card.

Priority Queueing

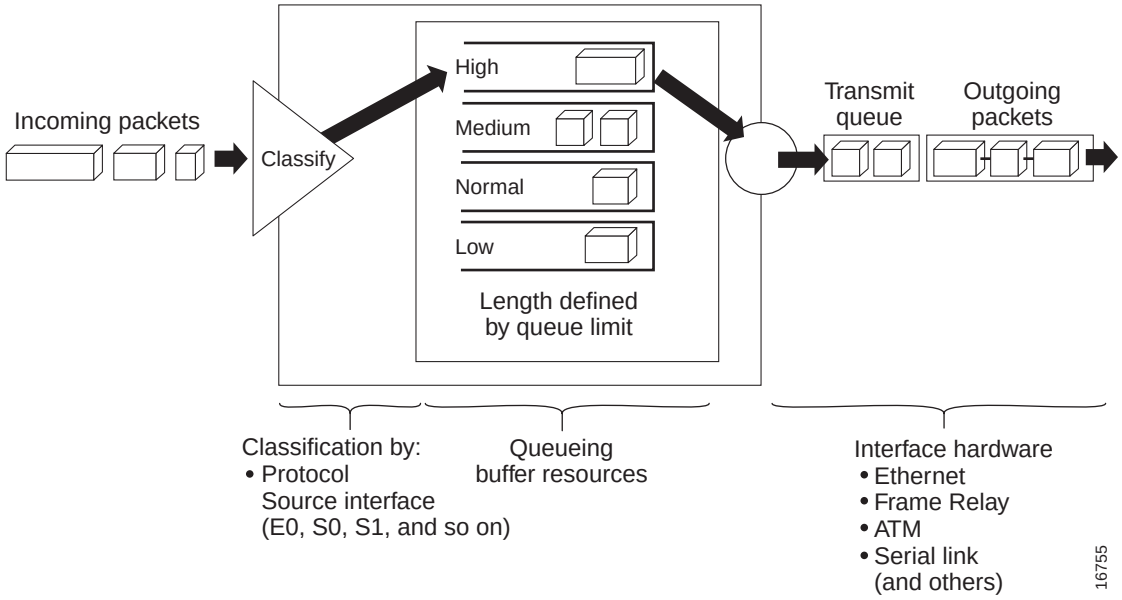
PQ allows you to define how traffic is prioritized in the network. You configure four traffic priorities. You can define a series of filters based on packet characteristics to cause the router to place traffic into these four queues; the queue with the highest priority is serviced first until it is empty,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

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

How It Works

During transmission, PQ gives priority queues absolute preferential treatment over low priority queues; important traffic, given the highest priority, always takes precedence over less important traffic. Packets are classified based on user-specified criteria and placed into one of the four output queues—high, medium, normal, and low—based on the assigned priority. Packets that are not classified by priority fall into the normal queue. Figure 3 illustrates this process.

Figure 3 Priority Queueing



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

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

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

How Packets Are Classified for Priority Queueing

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

Packets can be classified by the following criteria:

- Protocol or subprotocol type
- Incoming interface
- Packet size

- Fragments
- Access list

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

Why Use Priority Queueing?

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

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

Restrictions

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

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

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

PQ is not supported on any tunnels.

Bandwidth Management

RSVP, CBWFQ, LLQ, IP RTP Priority, Frame Relay IP RTP Priority, and Frame Relay PIPQ can all reserve and consume bandwidth, up to a maximum of the reserved bandwidth on an interface.

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

- For RSVP, use the **ip rsvp bandwidth** command.
- For CBWFQ, use the **bandwidth** policy-map class configuration command. For more information on CBWFQ bandwidth allocation, see the section "Class-Based Weighted Fair Queueing" in this chapter. For LLQ, you can allocate bandwidth using the **priority** command. For more information on LLQ bandwidth allocation, see the section "Frame Relay PVC Interface Priority Queueing" in this chapter.
- For IP RTP Priority, use the **ip rtp priority** command. For more information on IP RTP Priority bandwidth allocation, see the section "IP RTP Priority" in this chapter.
- For Frame Relay IP RTP Priority, use the **frame-relay ip rtp priority** command. For more information on Frame Relay IP RTP Priority, see the section "Frame Relay IP RTP Priority" in this chapter.

- For Frame Relay PVC Interface Priority Queueing, use the **frame-relay interface-queue priority** command. For more information on Frame Relay PIPQ, see the section "Frame Relay PVC Interface Priority Queueing" in this chapter.

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

If you find that it is necessary to change the maximum reserved bandwidth, you can change the maximum bandwidth by using the **max-reserved-bandwidth** command. The **max-reserved-bandwidth** command can be used only on interfaces; it cannot be used on VCs. On ATM VCs, ATM cell tax overhead is not included in the 75 percent maximum reserved bandwidth.

Configuring Weighted Fair Queueing

Configuring Weighted Fair Queueing

Feature History

Release	Modification
Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.

This chapter describes the tasks for configuring flow-based weighted fair queueing (WFQ), distributed WFQ (DWFQ), and class-based WFQ (CBWFQ), and distributed class-based WFQ (DCBWFQ) and the related features described in the following section, which provide strict priority queueing (PQ) within WFQ or CBWFQ:

- IP RTP Priority Queueing
- Frame Relay IP RTP Priority Queueing
- Frame Relay PVC Interface Priority Queueing
- Low Latency Queueing
- Distributed Low Latency Queueing
- Low Latency Queueing (LLQ) for Frame Relay
- Burst Size in Low Latency Queueing
- Per-VC Hold Queue Support for ATM Adapters

For complete conceptual information, see the "Congestion Management Overview" module.

For a complete description of the QoS commands in this chapter, see the *Cisco IOS Quality of Service Solutions Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

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

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Flow-Based Weighted Fair Queueing Configuration Task List

WFQ provides traffic priority management that automatically sorts among individual traffic streams without requiring that you first define access lists. WFQ can also manage duplex data streams such as those between pairs of applications, and simplex data streams such as voice or video. There are two categories of WFQ sessions: high bandwidth and low bandwidth. Low-bandwidth traffic has effective priority over high-bandwidth traffic, and high-bandwidth traffic shares the transmission service proportionally according to assigned weights.

When WFQ is enabled for an interface, new messages for high-bandwidth traffic streams are discarded after the configured or default congestive messages threshold has been met. However, low-bandwidth conversations, which include control message conversations, continue to enqueue data. As a result, the fair queue may occasionally contain more messages than its configured threshold number specifies.

With standard WFQ, packets are classified by flow. Packets with the same source IP address, destination IP address, source TCP or User Datagram Protocol (UDP) port, or destination TCP or UDP port belong to the same flow. WFQ allocates an equal share of the bandwidth to each flow. Flow-based WFQ is also called fair queueing because all flows are equally weighted.

The Cisco IOS software provides two forms of flow-based WFQ:

- Standard WFQ, which is enabled by default on all serial interfaces that run at 2 Mbps or below, and can run on all Cisco serial interfaces.
- Distributed WFQ, which runs only on Cisco 7000 series routers with a Route Switch Processor (RSP)-based RSP7000 interface processor or Cisco 7500 series routers with a Versatile Interface Processor (VIP)-based VIP2-40 or greater interface processor. (A VIP2-50 interface processor is strongly recommended when the aggregate line rate of the port adapters on the VIP is greater than DS3. A VIP2-50 interface processor is required for OC-3 rates.) For configuration information on DWFQ, see the section "Distributed Weighted Fair Queueing Configuration Task List" later in this chapter.

To configure flow-based WFQ, perform the tasks described in the following sections. The task in the first section is required; the task in the remaining section is optional.

- Configuring WFQ (Required)
- Monitoring Fair Queueing (Optional)

Flow-based WFQ is supported on unavailable bit rate (UBR), variable bit rate (VBR), and available bit rate (ABR) ATM connections.

See the end of this chapter for the section "Flow-Based WFQ Configuration Examples."

Configuring WFQ

To configure flow-based WFQ on an interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# fair-queue [congestive-discard-threshold [dynamic-queues [reservable-queues]]]`	Configures an interface to use WFQ.

Flow-based WFQ uses a traffic data stream discrimination registry service to determine to which traffic stream a message belongs. Refer to the table accompanying the description of the **fair-queue** (WFQ) command in the *Cisco IOS Quality of Service Solutions Command Reference* for the attributes of a message that are used to classify traffic into data streams.

Defaults are provided for the congestion threshold after which messages for high-bandwidth conversations are dropped, and for the number of dynamic and reservable queues; however, you can fine-tune your network operation by changing these defaults. Refer to the tables accompanying the description of the **fair-queue** (WFQ) command in the *Cisco IOS Quality of Service Solutions Command Reference* for the default number of dynamic queues that WFQ and CBWFQ use when they are enabled on an interface or ATM VC. These values do not apply for DWFQ.

Note WFQ is the default queueing mode on interfaces that run at E1 speeds (2.048 Mbps) or below. It is enabled by default for physical interfaces that do not use Link Access Procedure, Balanced (LAPB), X.25, or Synchronous Data Link Control (SDLC) encapsulations. WFQ is not an option for these protocols. WFQ is also enabled by default on interfaces configured for Multilink PPP (MLP). However, if custom queueing (CQ) or priority queueing (PQ0 is enabled for a qualifying link, it overrides fair queueing, effectively disabling it. Additionally, WFQ is automatically disabled if you enable autonomous or silicon switching.

Monitoring Fair Queueing

To monitor flow-based fair queueing services in your network, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router# show interfaces [interface]`	Displays statistical information specific to an interface.
`Router# show queue interface-type interface-number`	Displays the contents of packets inside a queue for a particular interface or virtual circuit (VC).
`Router# show queueing fair`	Displays status of the fair queueing configuration.

Distributed Weighted Fair Queueing Configuration Task List

To configure DWFQ, perform one of the mutually exclusive tasks described in the following sections:

- Configuring Flow-Based DWFQ, page 4
- Configuring QoS-Group-Based DWFQ, page 4
- Configuring Type of Service-Based DWFQ, page 5
- Monitoring DWFQ, page 5 (Optional)

If you enable flow-based DWFQ and then enable class-based DWFQ (either QoS-group based or ToS-based), class-based DWFQ will replace flow-based DWFQ.

If you enable class-based DWFQ and then want to switch to flow-based DWFQ, you must disable class-based DWFQ using the **no fair-queue class-based** command before enabling flow-based DWFQ.

If you enable one type of class-based DWFQ and then enable the other type, the second type will replace the first.

DWFQ runs only on Cisco 7000 series routers with an RSP-based RSP7000 interface processor or Cisco 7500 series routers with a VIP-based VIP2-40 or greater interface processor. (A VIP2-50 interface processor is strongly recommended when the aggregate line rate of the port adapters on the VIP is greater than DS3. A VIP2-50 interface processor is required for OC-3 rates.)

DWFQ can be configured on interfaces but not subinterfaces. It is not supported on Fast EtherChannel, tunnel, or other logical or virtual interfaces such as MLP.

See the end of this chapter for the section "DWFQ Configuration Examples."

Configuring Flow-Based DWFQ

To configure flow-based DWFQ, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# fair-queue`	Enables flow-based DWFQ.
Step 2	`Router(config-if)# fair-queue aggregate-limit` *aggregate-packet*	(Optional) Sets the total number of buffered packets before some packets may be dropped. Below this limit, packets will not be dropped.
Step 3	`Router(config-if)# fair-queue individual-limit` *individual-packet*	(Optional) Sets the maximum queue size for individual per-flow queues during periods of congestion.

For flow-based DWFQ, packets are classified by flow. Packets with the same source IP address, destination IP address, source TCP or UDP port, destination TCP or UDP port, and protocol belong to the same flow.

In general, you should not change the aggregate or individual limit value from the default. Use the **fair-queue aggregate-limit** and **fair-queue individual-limit** commands only if you have determined that you would benefit from using different values, based on your particular situation.

Configuring QoS-Group-Based DWFQ

To configure QoS-group-based DWFQ, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# fair-queue qos-group`	Enables QoS-group-based DWFQ.
Step 2	`Router(config-if)# fair-queue qos-group` *number* `weight` *weight*	For each QoS group, specifies the percentage of the bandwidth to be allocated to each class.
Step 3	`Router(config-if)# fair-queue aggregate-limit` *aggregate-packet*	(Optional) Sets the total number of buffered packets before some packets may be dropped. Below this limit, packets will not be dropped.
Step 4	`Router(config-if)# fair-queue individual-limit` *individual-packet*	(Optional) Sets the maximum queue size for every per-flow queue during periods of congestion.
Step 5	`Router(config-if)# fair-queue qos-group` *number* `limit` *class-packet*	(Optional) Sets the maximum queue size for a specific QoS group queue during periods of congestion.

In general, you should not change the aggregate, individual, or class limit value from the default. Use the **fair-queue aggregate-limit**, **fair-queue individual-limit**, and **fair-queue limit** commands only if you have determined that you would benefit from using different values, based on your particular situation.

Configuring Type of Service-Based DWFQ

To configure type of service (ToS)-based DWFQ, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# fair-queue tos`	Enables ToS-based DWFQ
Step 2	`Router(config-if)# fair-queue tos` *number* `weight` *weight*	(Optional) For each ToS class, specifies the percentage of the bandwidth to be allocated to each class.
Step 3	`Router(config-if)# fair-queue aggregate-limit` *aggregate-packet*	(Optional) Sets the total number of buffered packets before some packets may be dropped. Below this limit, packets will not be dropped.
Step 4	`Router(config-if)# fair-queue individual-limit` *individual-packet*	(Optional) Sets the maximum queue size for every per-flow queue during periods of congestion.
Step 5	`Router(config-if)# fair-queue tos` *number* `limit` *class-packet*	(Optional) Sets the maximum queue size for a specific ToS queue during periods of congestion.

In general, you should not change the aggregate, individual, or class limit value from the default. Use the **fair-queue aggregate-limit**, **fair-queue individual-limit**, and **fair-queue limit** commands only if you have determined that you would benefit from using different values, based on your particular situation.

Monitoring DWFQ

To monitor DWFQ, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router# show interfaces` [*interface*]	Displays the statistical information specific to an interface.
`Router# show queueing fair-queue`	Displays status of the fair queueing configuration.

Class-Based Weighted Fair Queueing Configuration Task List

To configure CBWFQ, perform the tasks described in the following sections. The tasks in the first three sections are required; the tasks in the remaining sections are optional.

- Defining Class Maps, page 6 (Required)
- Configuring Class Policy in the Policy Map, page 7 (Required)
- Attaching the Service Policy and Enabling CBWFQ, page 10 (Required)

- Modifying the Bandwidth for an Existing Policy Map Class, page 11 (Optional)
- Modifying the Queue Limit for an Existing Policy Map Class, page 11 (Optional)
- Configuring the Bandwidth Limiting Factor, page 11 (Optional)
- Deleting Classes, page 12 (Optional)
- Deleting Policy Maps, page 12 (Optional)
- Verifying Configuration of Policy Maps and Their Classes, page 12 (Optional)

CBWFQ is supported on VBR and ABR ATM connections. It is not supported on UBR connections.

See the end of this chapter for the section "CBWFQ Configuration Examples."

For information on how to configure per-VC WFQ and CBWFQ, see the "Configuring IP to ATM Class of Service" module.

Defining Class Maps

To create a class map containing match criteria against which a packet is checked to determine if it belongs to a class—and to effectively create the class whose policy can be specified in one or more policy maps—use the first command in global configuration mode to specify the class map name, then use one of the following commands in class-map configuration mode, as needed:

	Command	Purpose
Step 1	`Router(config)# class-map class-map-name`	Specifies the name of the class map to be created.
Step 2	`Router(config-cmap)# match access-group {access-group | name access-group-name}`	Specifies the name of the access control list (ACL) against whose contents packets are checked to determine if they belong to the class. CBWFQ supports numbered and named ACLs.
	or	
	`Router(config-cmap)# match input-interface interface-name`	Specifies the name of the input interface used as a match criterion against which packets are checked to determine if they belong to the class.
	or	
	`Router(config-cmap)# match protocol protocol`	Specifies the name of the protocol used as a match criterion against which packets are checked to determine if they belong to the class.
	or	
	`Router(config-cmap)# match mpls experimental number`	Specifies the value of the EXP field to be used as a match criterion against which packets are checked to determine if they belong to the class.

Other match criteria can be used when defining class maps. For additional match criteria, see "Applying QoS Features Using the MQC" module.

Configuring Class Policy in the Policy Map

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

- **class**
- **bandwidth** (policy-map class)
- **fair-queue** (for class-default class only)
- **queue-limit** or **random-detect**

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

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

You can configure class policies for as many classes as are defined on the router, up to the maximum of 64. However, the total amount of bandwidth allocated for all classes included in a policy map must not exceed 75 percent of the available bandwidth on the interface. The other 25 percent is used for control and routing traffic. (To override the 75 percent limitation, use the **max-reserved bandwidth** command.) If not all of the bandwidth is allocated, the remaining bandwidth is proportionally allocated among the classes, based on their configured bandwidth.

To configure class policies in a policy map, perform the optional tasks described in the following sections. If you do not perform the steps in these sections, the default actions are used.

- Configuring Class Policy Using Tail Drop, page 7 (Optional)
- Configuring Class Policy Using WRED Packet Drop, page 8 (Optional)
- Configuring the Class-Default Class Policy, page 9 (Optional)

Configuring Class Policy Using Tail Drop

To configure a policy map and create class policies that make up the service policy, use the first command in global configuration mode to specify the policy map name, then use the following commands in policy-map class configuration mode, as needed, to configure policy for a standard class. To configure policy for the default class, see the section "Configuring the Class-Default Class Policy" in this chapter.

	Command	Purpose
Step 1	`Router(config)# policy-map policy-map`	Specifies the name of the policy map to be created or modified.
Step 2	`Router(config-pmap)# class class-name`	Specifies the name of a class to be created and included in the service policy.

	Command	Purpose
Step 3	`Router(config-pmap-c)# bandwidth {bandwidth-kbps | percent percent}`	Specifies the amount of bandwidth, in kbps, or percentage of available bandwidth, to be assigned to the class. The amount of bandwidth configured should be large enough to also accommodate Layer 2 overhead.
Step 4	`Router(config-pmap-c)# queue-limit number-of-packets`	Specifies the maximum number of packets that can be queued for the class.

To configure policy for more than one class in the same policy map, repeat Step 2 through Step 4. Note that because this set of commands uses the **queue-limit** command, the policy map uses tail drop, not Weighted Random Early Detection (WRED) packet drop.

Configuring Class Policy Using WRED Packet Drop

To configure a policy map and create class policies comprising the service policy, use the first command in global configuration mode, as needed, to specify the policy map name, then use the following commands in policy-map class configuration mode, as needed, to configure policy for a standard class. To configure policy for the default class, see the section "Configuring the Class-Default Class Policy" in this chapter.

	Command	Purpose
Step 1	`Router(config)# policy-map policy-map`	Specifies the name of the policy map to be created or modified.
Step 2	`Router(config-pmap)# class class-name`	Specifies the name of a class to be created and included in the service policy.
Step 3	`Router(config-pmap-c)# bandwidth {bandwidth-kbps | percent percent}`	Specifies the amount of bandwidth, in kbps, or percentage of available bandwidth to be assigned to the class. The amount of bandwidth configured should be large enough to also accommodate Layer 2 overhead.
Step 4	`Router(config-pmap-c)# random-detect`	Enables WRED. The class policy will drop packets using WRED instead of tail drop.
Step 5	`Router(config-pmap-c)# random-detect exponential-weighting-constant exponent` or `Router(config-pmap-c)# random-detect precedence precedence min-threshold max-threshold mark-prob-denominator`	Configures the exponential weight factor used in calculating the average queue length. Configures WRED parameters for packets with a specific IP precedence. Repeat this command for each precedence.

To configure policy for more than one class in the same policy map, repeat Step 2 through Step 5. Note that this set of commands uses WRED packet drop, not tail drop.

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

Configuring the Class-Default Class Policy

The class-default class is used to classify traffic that does not fall into one of the defined classes. Once a packet is classified, all of the standard mechanisms that can be used to differentiate service among the classes apply. The class-default class was predefined when you created the policy map, but you must configure it. If no default class is configured, then by default the traffic that does not match any of the configured classes is flow classified and given best-effort treatment.

By default, the class-default class is defined as flow-based WFQ. However, configuring the default class with the **bandwidth** policy-map class configuration command disqualifies the default class as flow-based WFQ.

To configure a policy map and configure the class-default class to use tail drop, use the first command in global configuration mode to specify the policy map name, then to configure policy for the default class use the following commands in policy-map class configuration mode:

	Command	Purpose
Step 1	`Router(config)# policy-map policy-map`	Specifies the name of the policy map to be created or modified.
Step 2	`Router(config-pmap)# class class-default default-class-name`	Specifies the default class so that you can configure or modify its policy.
Step 3	`Router(config-pmap-c)# bandwidth {bandwidth-kbps | percent percent}` or `Router(config-pmap-c)# fair-queue [number-of-dynamic-queues]`	Specifies the amount of bandwidth, in kbps, or percentage of available bandwidth to be assigned to the class. The amount of bandwidth configured should be large enough to also accommodate Layer 2 overhead. Specifies the number of dynamic queues to be reserved for use by flow-based WFQ running on the default class. The number of dynamic queues is derived from the bandwidth of the interface. Refer to the tables accompanying the description of the **fair-queue** (WFQ) command in the *Cisco IOS Quality of Service Solutions Command Reference* for the default number of dynamic queues that WFQ and CBWFQ use when they are enabled on an interface or ATM VC.
Step 4	`Router(config-pmap-c)# queue-limit number-of-packets`	Specifies the maximum number of packets that the queue for the default class can accumulate.

To configure a policy map and configure the class-default class to use WRED packet drop, use the first command in global configuration mode to specify the policy map name, then to configure policy for the default class use the following commands in policy-map class configuration mode:

	Command	Purpose
Step 1	`Router(config)# policy-map policy-map`	Specifies the name of the policy map to be created or modified.
Step 2	`Router(config-pmap)# class class-default default-class-name`	Specifies the default class so that you can configure or modify its policy.

	Command	Purpose
Step 3	`Router(config-pmap-c)# bandwidth {bandwidth-kbps | percent percent}` or `Router(config-pmap-c)# fair-queue [number-of-dynamic-queues]`	Specifies the amount of bandwidth, in kbps, or percentage of available bandwidth to be assigned to the class. The amount of bandwidth configured should be large enough to also accommodate Layer 2 overhead. Specifies the number of dynamic queues to be reserved for use by flow-based WFQ running on the default class The number of dynamic queues is derived from the bandwidth of the interface. Refer to the tables accompanying the description of the **fair-queue** (WFQ) command in the *Cisco IOS Quality of Service Solutions Command Reference* for the default number of dynamic queues that WFQ and CBWFQ use when they are enabled on an interface or ATM VC.
Step 4	`Router(config-pmap-c)# random-detect`	Enables WRED. The class policy will drop packets using WRED instead of tail drop.
Step 5	`Router(config-pmap-c)# random-detect exponential-weighting-constant exponent` or `Router(config-pmap-c)# random-detect precedence precedence min-threshold max-threshold mark-prob-denominator`	Configures the exponential weight factor used in calculating the average queue length. Configures WRED parameters for packets with a specific IP precedence. Repeat this command for each precedence.

Attaching the Service Policy and Enabling CBWFQ

To attach a service policy to the output interface and enable CBWFQ on the interface, use the following command in interface configuration mode. When CBWFQ is enabled, all classes configured as part of the service policy map are installed in the fair queueing system.

Command	Purpose
`Router(config-if)# service-policy output policy-map`	Enables CBWFQ and attaches the specified service policy map to the output interface.

Configuring CBWFQ on a physical interface is only possible if the interface is in the default queueing mode. Serial interfaces at E1 (2.048 Mbps) and below use WFQ by default—other interfaces use FIFO by default. Enabling CBWFQ on a physical interface overrides the default interface queueing method. Enabling CBWFQ on an ATM permanent virtual circuit (PVC) does not override the default queueing method.

Modifying the Bandwidth for an Existing Policy Map Class

To change the amount of bandwidth allocated for an existing class, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# policy-map policy-map`	Specifies the name of the policy map containing the class to be modified.
Step 2	`Router(config-pmap)# class class-name`	Specifies the name of a class whose bandwidth you want to modify.
Step 3	`Router(config-pmap-c)# bandwidth {bandwidth-kbps | percent percent}`	Specifies the new amount of bandwidth, in kbps, or percentage of available bandwidth to be used to reconfigure the class. The amount of bandwidth configured should be large enough to also accommodate Layer 2 overhead.

Modifying the Queue Limit for an Existing Policy Map Class

To change the maximum number of packets that can accrue in a queue reserved for an existing class, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# policy-map policy-map`	Specifies the name of the policy map containing the class to be modified.
Step 2	`Router(config-pmap)# class class-name`	Specifies the name of a class whose queue limit you want to modify.
Step 3	`Router(config-pmap-c)# queue-limit number-of-packets`	Specifies the new maximum number of packets that can be queued for the class to be reconfigured. The default and maximum number of packets is 64.

Configuring the Bandwidth Limiting Factor

To change the maximum reserved bandwidth allocated for Resource Reservation Protocol (RSVP), CBWFQ, LLQ, IP RTP Priority, Frame Relay IP RTP Priority, and Frame Relay PVC Interface Priority Queueing (PIPQ), use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# max-reserved-bandwidth percent`	Changes the maximum configurable bandwidth for RSVP, CBWFQ, LLQ, IP RTP Priority, Frame Relay IP RTP Priority, and Frame Relay PVC Interface Priority Queueing. The default is 75 percent.

Deleting Classes

To delete one or more class maps from a service policy map, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# policy-map policy-map`	Specifies the name of the policy map containing the classes to be deleted.
Step 2	`Router(config-pmap)# no class class-name`	Specifies the name of the classes to be deleted.
Step 3	`Router(config-pmap-c)# no class class-default`	Deletes the default class.

Deleting Policy Maps

To delete a policy map, use the following command in global configuration mode:

Command	Purpose
`Router(config)# no policy-map policy-map`	Specifies the name of the policy map to be deleted.

Verifying Configuration of Policy Maps and Their Classes

To display the contents of a specific policy map, a specific class from a specific policy map, or all policy maps configured on an interface, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router# show policy-map policy-map`	Displays the configuration of all classes that make up the specified policy map.
`Router# show policy-map policy-map class class-name`	Displays the configuration of the specified class of the specified policy map.
`Router# show policy-map interface interface-name`	Displays the configuration of all classes configured for all policy maps on the specified interface.
`Router# show queue interface-type interface-number`	Displays queueing configuration and statistics for a particular interface.

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

Distributed Class-Based Weighted Fair Queueing Configuration Task List

To configure DCBWFQ, perform the tasks described in the following sections. Although all the tasks are listed as optional, you must complete the task in either the first or second section.

- Modifying the Bandwidth for an Existing Traffic Class, page 13 (Optional)
- Modifying the Queue Limit for an Existing Traffic Class, page 13 (Optional)
- Monitoring and Maintaining DCBWFQ, page 14 (Optional)

DCBWFQ is configured using user-defined traffic classes and service policies. Traffic classes and service policies are configured using the Modular Quality of Service Command-Line Interface (CLI) feature. For information on how to configure QoS with the Modular QoS CLI, see the "Applying QoS Features Using the MQC" module.

See the end of this chapter for the section "Verifying Configuration of Policy Maps and Their Classes."

Modifying the Bandwidth for an Existing Traffic Class

To change the amount of bandwidth allocated for an existing traffic class in congested environments, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# policy-map policy-map`	Specifies the name of the traffic policy to be created or modified.
Step 2	`Router(config-pmap)# class class-name`	Specifies the name of a traffic class whose bandwidth you want to modify.
Step 3	`Router(config-pmap-c)# bandwidth bandwidth-kbps`	Specifies the amount of allocated bandwidth, in kbps, to be reserved for the traffic class in congested network environments.

After configuring the traffic policy with the **policy-map** command, you must still attach the traffic policy to an interface before it is successfully enabled. For information on attaching a traffic policy to an interface, see the "Applying QoS Features Using the MQC" module.

Modifying the Queue Limit for an Existing Traffic Class

To change the maximum number of packets that can accrue in a queue reserved for an existing traffic class, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# policy-map policy-map`	Specifies the name of the traffic policy to be created or modified.

	Command	Purpose
Step 2	`Router(config-pmap)# class class-name`	Specifies the name of a traffic class whose queue limit you want to modify.
Step 3	`Router(config-pmap-c)# queue-limit number-of-packets`	Specifies the new maximum number of packets that can be queued for the traffic class to be reconfigured. The default and maximum number of packets is 64.

After configuring the service policy with the **policy-map** command, you must still attach the traffic policy to an interface before it is successfully enabled. For information on attaching a traffic policy to an interface, see the "Applying QoS Features Using the MQC" module.

Monitoring and Maintaining DCBWFQ

To display the configuration of a traffic policy and its associated traffic classes, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router# show policy-map`	Displays all configured traffic policies.
`Router# show policy-map policy-map-name`	Displays the user-specified traffic policy.
`Router# show policy-map interface`	Displays statistics and configurations of all input and output policies attached to an interface.
`Router# show policy-map interface interface-spec`	Displays configuration and statistics of the input and output policies attached to a particular interface.
`Router# show policy-map interface interface-spec input`	Displays configuration and statistics of the input policy attached to an interface.
`Router# show policy-map interface interface-spec output`	Displays configuration statistics of the output policy attached to an interface.
`Router# show policy-map [interface [interface-spec [input | output] [class class-name]]]`	Displays the configuration and statistics for the class name configured in the policy.

IP RTP Priority Configuration Task List

To configure IP RTP Priority, perform the tasks described in the following sections. The task in the first section is required; the tasks in the remaining sections are optional.

- Configuring IP RTP Priority, page 15 (Required)
- Configuring the Bandwidth Limiting Factor, page 15 (Optional)
- Verifying IP RTP Priority, page 15 (Optional)
- Monitoring and Maintaining IP RTP Priority, page 16 (Optional)

See the end of this chapter for the section "IP RTP Priority Configuration Examples."

Frame Relay Traffic Shaping (FRTS) and Frame Relay Fragmentation (FRF.12 or higher) must be configured before the Frame Relay IP RTP Priority feature is used. For information about configuring FRTS and FRF.12, see the "MQC-Based Frame Relay Traffic Shaping" module and the "FRF.20 Support" modules, respectively.

Configuring IP RTP Priority

To reserve a strict priority queue for a set of RTP packet flows belonging to a range of UDP destination ports, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ip rtp priority starting-rtp-port-number port-number-range bandwidth`	Reserves a strict priority queue for a set of RTP packet flows belonging to a range of UDP destination ports.

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

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

The **frame-relay ip rtp priority** command provides strict PQ for Frame Relay PVCs. For more information about this command, see the *Cisco IOS Quality of Service Solutions Command Reference*.

Configuring the Bandwidth Limiting Factor

To change the maximum reserved bandwidth allocated for CBWFQ, LLQ, and the IP RTP Priority feature, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# max-reserved-bandwidth percent`	Changes the maximum configurable bandwidth for CBWFQ, LLQ, and IP RTP Priority. The default is 75 percent.

Verifying IP RTP Priority

To display the contents of the priority queue (such as queue depth and the first packet queued), use the following command in EXEC mode:

Command	Purpose
`Router# show queue interface-type interface-number`	Displays queueing configuration and statistics for a particular interface.

Monitoring and Maintaining IP RTP Priority

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

Command	Purpose
`Router# debug priority`	Displays priority queueing output if packets are dropped from the priority queue.
`Router# show queue interface-type interface-number`	Displays queueing configuration and statistics for a particular interface.

Frame Relay IP RTP Priority Configuration Task List

To configure Frame Relay IP RTP Priority, perform the tasks described in the following sections. The task in the first section is required; the tasks in the remaining sections are optional.

- Configuring Frame Relay IP RTP Priority, page 16 (Required)
- Verifying Frame Relay IP RTP Priority, page 17 (Optional)
- Monitoring and Maintaining Frame Relay IP RTP Priority, page 17 (Optional)

See the end of this chapter for the section "Frame Relay IP RTP Priority Configuration Examples."

Configuring Frame Relay IP RTP Priority

To reserve a strict priority queue on a Frame Relay PVC for a set of RTP packet flows belonging to a range of UDP destination ports, use the following command in map-class configuration mode:

Command	Purpose
`Router(config-map-class)# frame-relay ip rtp priority starting-rtp-port-number port-number-range bandwidth`	Reserves a strict priority queue for a set ofRTP packet flows belonging to a range of UDP destination ports.

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

Verifying Frame Relay IP RTP Priority

To verify the Frame Relay IP RTP Priority feature, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router# show frame relay pvc`	Displays statistics about PVCs for Frame Relay interfaces.
`Router# show queue interface-type interface-number`	Displays fair queueing configuration and statistics for a particular interface.
`Router# show traffic-shape queue`	Displays information about the elements queued at a particular time at the VC data-link connection identifier (DLCI) level.

Monitoring and Maintaining Frame Relay IP RTP Priority

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

Command	Purpose
`Router# debug priority`	Displays priority queueing output if packets are dropped from the priority queue.

Frame Relay PVC Interface Priority Configuration Task List

To configure the Frame Relay PVC Interface Priority feature, perform the tasks described in the following sections. The tasks in the first three sections are required; the tasks in the remaining sections are optional.

- Configuring PVC Priority in a Map Class, page 18 (Required)
- Enabling Frame Relay PIPQ and Setting Queue Limits, page 18 (Required)
- Assigning a Map Class to a PVC, page 18 (Required)
- Verifying Frame Relay PIPQ, page 19 (Optional)
- Monitoring and Maintaining Frame Relay PIPQ, page 19 (Optional)

See the end of this chapter for the section "Frame Relay PVC Interface PQ Configuration Examples."

Configuring PVC Priority in a Map Class

To configure PVC priority within a map class, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# map-class frame-relay map-class-name`	Specifies a Frame Relay map class.
Step 2	`Router(config-map-class)# frame-relay interface-queue priority {high | medium | normal | low}`	Assigns a PVC priority level to a Frame Relay map class.

Enabling Frame Relay PIPQ and Setting Queue Limits

To enable Frame Relay (FR) PIPQ and set the priority queue sizes, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type number [name-tag]`	Configures an interface type and enters interface configuration mode.
Step 2	`Router(config-if)# encapsulation frame-relay [cisco | ietf]`	Enables Frame Relay encapsulation.
Step 3	`Router(config-if)# frame-relay interface-queue priority [high-limit medium-limit normal-limit low-limit]`	Enables Frame Relay PIPQ and sets the priority queue limits.

Assigning a Map Class to a PVC

To assign a map class to a specific PVC, use the following commands beginning in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# frame-relay interface-dlci dlci`	Specifies a single PVC on a Frame Relay interface.
Step 2	`Router(config-fr-dlci)# class map-class-name`	Associates a map class with a specified PVC.

Verifying Frame Relay PIPQ

To verify the configuration of Frame Relay (FR) PIPQ, use the following commands in privileged EXEC mode, as needed:

Command	Purpose
Router# **show frame-relay pvc** [**interface** *interface*][*dlci*]	Displays statistics about PVCs for Frame Relay interfaces.
Router# **show interfaces** [*type number*][*first*][*last*]	Displays the statistical information specific to a serial interface.
Router# **show queueing** [**custom** | **fair** | **priority** | **random-detect** [**interface** *atm_subinterface* [**vc** [[**vpi/**] **vci**]]]]	Lists all or selected configured queueing strategies.

Monitoring and Maintaining Frame Relay PIPQ

To monitor and maintain Frame Relay (FR) PIPQ, use the following commands in privileged EXEC mode, as needed:

Command	Purpose
Router# **debug priority**	Displays priority queueing output if packets are dropped from the priority queue.
Router# **show frame-relay pvc** [**interface** *interface*][*dlci*]	Displays statistics about PVCs for Frame Relay interfaces.
Router# **show interfaces** [*type number*][*first*][*last*]	Displays the statistical information specific to a serial interface.
Router# **show queue** *interface-name interface-number* [**vc** [*vpi/*] *vci*][*queue-number*]	Displays the contents of packets inside a queue for a particular interface or VC.
Router# **show queueing** [**custom** | **fair** | **priority** | **random-detect** [**interface** ***atm_subinterface*** [**vc** [[*vpi/*] *vci*]]]]	Lists all or selected configured queueing strategies.

Low Latency Queueing Configuration Task List

To configure LLQ, perform the tasks described in the following sections. The task in the first section is required; the tasks in the remaining sections are optional.

- Configuring LLQ, page 20 (Required)
- Configuring the Bandwidth Limiting Factor, page 20 (Optional)
- Verifying LLQ, page 20 (Optional)
- Monitoring and Maintaining LLQ, page 21 (Optional)

See the end of this chapter for the section "LLQ Configuration Examples."

Configuring LLQ

To give priority to a class within a policy map, use the following command in policy-map class configuration mode:

Command	Purpose
`Router(config-pmap-c)# priority bandwidth`	Reserves a strict priority queue for this class of traffic.

Configuring the Bandwidth Limiting Factor

To change the maximum reserved bandwidth allocated for CBWFQ, LLQ, and IP RTP Priority, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# max-reserved-bandwidth percent`	Changes the maximum configurable bandwidth for CBWFQ, LLQ, and IP RTP Priority. The default is 75 percent.

Verifying LLQ

To display the contents of the priority queue, such as queue depth and the first packet queued, use the following command in EXEC mode:

Command	Purpose
`Router# show queue interface-type interface-number`	Displays queueing configuration and statistics for a particular interface.

The priority queue is the queue whose conversation ID is equal to the number of dynamic queues plus 8. The packets in the priority queue have a weight of 0.

Monitoring and Maintaining LLQ

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

Command	Purpose
`Router# debug priority`	Displays priority queueing output if packets are dropped from the priority queue.
`Router# show queue interface-type interface-number`	Displays queueing configuration and statistics for a particular interface.
`Router# show policy-map interface interface-name`	Displays the configuration of all classes configured for all traffic policies on the specified interface. Displays if packets and bytes were discarded or dropped for the priority class in the traffic policy attached to the interface.

Distributed LLQ Configuration Task List

To configure Distributed LLQ, perform the tasks described in the following sections. The tasks in the first two sections are required; the tasks in the remaining sections are optional.

- Configuring a Priority Queue for an Amount of Available Bandwidth, page 21 (Required)
- Configuring a Priority Queue for a Percentage of Available Bandwidth, page 22 (Required)
- Configuring a Transmission Ring Limit, page 22 (Optional)
- Verifying Distributed LLQ, page 23 (Optional)
- Verifying a Transmission Ring Limit, page 23 (Optional)
- Monitoring and Maintaining Distributed LLQ, page 23 (Optional)

See the end of this chapter for the section "Distributed LLQ Configuration Examples."

Configuring a Priority Queue for an Amount of Available Bandwidth

To give priority to a traffic class based on the amount of available bandwidth within a traffic policy, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# policy-map policy-name`	Specifies the name of the policy map to configure. Enters policy-map configuration mode.
Step 2	`Router(config-pmap)# class class-name`	Specifies the name of a predefined class included in the service policy. Enters policy-map class configuration mode.
Step 3	`Router(config-pmap-c)# priority kpbs [bytes]`	Reserves a priority queue with a specified amount of available bandwidth for CBWFQ traffic.

The traffic policy configured in this section is not yet attached to an interface. For information on attaching a traffic policy to an interface, see the "Applying QoS Features Using the MQC" module.

Configuring a Priority Queue for a Percentage of Available Bandwidth

To give priority to a class based on a percentage of available bandwidth, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# policy-map policy-name`	Specifies the name of the traffic policy to configure. Enters policy-map configuration mode.
Step 2	`Router(config-pmap)# class class-name`	Specifies the name of a predefined class included in the service policy. Enters policy-map class configuration mode.
Step 3	`Router(config-pmap-c)# priority percent percent`	Reserves a priority queue with a specified percentage of available bandwidth for CBWFQ traffic.

The traffic policy configured in this section is not yet attached to an interface. For information on attaching a traffic policy to an interface, see the "Applying QoS Features Using the MQC" module.

Configuring a Transmission Ring Limit

To limit the number of allowable particles on a transmission ring on an ATM PVC, use the following commands beginning in global interface configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface atm interface-name`	Specifies the name of the ATM interface to configure.
Step 2	`Router(config-if)# atm pvc vcd-number vpi-number vci-number Encapsulation-type tx-ring-limit ring-limit`	Specifies the ATM PVC to configure, the encapsulation type, and the transmission ring limit value.

To limit the number of allowable particles on a transmission ring on an ATM subinterface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface atm subinterface name`	Specifies the name of the subinterface to configure.
Step 2	`Router(config-subif)# pvc pvc-name`	Specifies the name of the PVC to configure.
Step 3	`Router(config-if-atm-vc)# tx-ring-limit ring-limit`	Specifies the transmission ring limit value.

Verifying Distributed LLQ

To view the contents of the priority queue, such as queue depth and the first packet queued, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router# show interfaces [interface-type interface-number] fair-queue`	Displays information and statistics about WFQ for a VIP-based interface.
`Router# show policy-map policy-map-name`	Displays the contents of a policy map, including the priority setting in a specific policy map.

The priority queue is the queue in which the conversation ID is equal to the number of dynamic queues plus 8. The packets in the priority queue have a weight of 0.

Verifying a Transmission Ring Limit

To display the contents of the interface or the PVC, use the following command in EXEC mode:

Command	Purpose
`Router# show atm vc vc-name`	Displays the contents of a VC. The **show atm vc** command output will indicate the transmission ring limit value if the **tx-ring-limit** command is successfully enabled.

Monitoring and Maintaining Distributed LLQ

To tune your Real-Time Transport Protocol (RTP) bandwidth or to decrease RTP traffic if the priority queue is experiencing drops, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router# show interfaces [interface-type interface-number] fair-queue`	Displays information and statistics about WFQ for a VIP-based interface.
`Router# show policy-map policy-map-name`	Displays the contents of a traffic policy, including the priority setting in a specific policy map.
`Router# show policy interface interface-name`	Displays the configuration of all classes configured for all service policies on the specified interface. Displays if packets and bytes were discarded or dropped for the priority class in the service policy attached to the interface.
`Router# show atm vc vc-name`	Displays the contents of a VC. The **show atm vc** command output will indicate the transmission ring limit value if the **tx-ring-limit** command is successfully enabled.

Low Latency Queueing for Frame Relay Configuration Task List

To configure LLQ for Frame Relay, perform the tasks described in the following sections. The tasks in the first three sections are required; the tasks in the remaining section are optional.

- Defining Class Maps, page 6 (Required)
- Configuring Class Policy in the Policy Map, page 24 (Required)
- Attaching the Service Policy and Enabling LLQ for Frame Relay, page 26 (Required)
- Verifying Configuration of Policy Maps and Their Classes, page 27 (Optional)
- Monitoring and Maintaining LLQ for Frame Relay, page 27 (Optional)

See the end of this chapter for the section "LLQ for Frame Relay Configuration Examples."

Defining Class Maps

To create a class map containing match criteria against which a packet is checked to determine if it belongs to a class, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# class-map class-map-name`	Specifies the name of the class map to be created.
Step 2	`Router(config-cmap)# match access-group {access-group | name access-group-name}`	Specifies the name of the ACL against whose contents packets are checked to determine if they belong to the class.
	or	
	`Router(config-cmap)# match input-interface interface-name`	Specifies the name of the input interface used as a match criterion against which packets are checked to determine if they belong to the class.
	or	
	`Router(config-cmap)# match protocol protocol`	Specifies the name of the protocol used as a match criterion against which packets are checked to determine if they belong to the class.

Configuring Class Policy in the Policy Map

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

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

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

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

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

To configure class policies in a policy map, perform the tasks described in the following sections. The task in the first section is required; the tasks in the remaining sections are optional.

- Configuring Class Policy for a LLQ Priority Queue, page 25 (Required)
- Configuring Class Policy Using a Specified Bandwidth and WRED Packet Drop, page 25 (Optional)
- Configuring the Class-Default Class Policy, page 26 (Optional)

Configuring Class Policy for a LLQ Priority Queue

To configure a policy map and give priority to a class within the policy map, use the following commands beginning in global configuration mode:

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

Configuring Class Policy Using a Specified Bandwidth and WRED Packet Drop

To configure a policy map and create class policies that make up the service policy, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# policy-map policy-map`	Specifies the name of the policy map to be created or modified.
Step 2	`Router(config-pmap)# class class-name`	Specifies the name of a class to be created and included in the service policy.

	Command	Purpose
Step 3	`Router(config-pmap-c)# bandwidth bandwidth-kbps`	Specifies the amount of bandwidth to be assigned to the class, in kbps, or as a percentage of the available bandwidth. Bandwidth must be specified in kbps or as a percentage consistently across classes. (Bandwidth of the priority queue must be specified in kbps.)
Step 4	`Router(config-pmap-c)# random-detect`	Enables WRED.

To configure policy for more than one class in the same policy map, repeat Step 2 through Step 4.

Configuring the Class-Default Class Policy

The class-default class is used to classify traffic that does not fall into one of the defined classes. Even though the class-default class is predefined when you create the policy map, you still have to configure it. If a default class is not configured, then traffic that does not match any of the configured classes is given best-effort treatment, which means that the network will deliver the traffic if it can, without any assurance of reliability, delay prevention, or throughput.

To configure a policy map and the class-default class, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# policy-map policy-map`	Specifies the name of the policy map to be created or modified.
Step 2	`Router(config-pmap)# class class-default default-class-name`	Specifies the default class so that you can configure or modify its policy.
Step 3	`Router(config-pmap-c)# bandwidth bandwidth-kbps` or `Router(config-pmap-c)# fair-queue [number-of-dynamic-queues]`	Specifies the amount of bandwidth, in kbps, to be assigned to the class. Specifies the number of dynamic queues to be reserved for use by flow-based WFQ running on the default class. The number of dynamic queues is derived from the bandwidth of the interface.
Step 4	`Router(config-pmap-c)# queue-limit number-of-packets`	Specifies the maximum number of packets that the queue for the default class can accumulate.

Attaching the Service Policy and Enabling LLQ for Frame Relay

To attach a service policy to the output interface and enable LLQ for Frame Relay, use the following command in map-class configuration mode. When LLQ is enabled, all classes configured as part of the service policy map are installed in the fair queueing system.

Command	Purpose
`Router(config-map-class)# service-policy output policy-map`	Attaches the specified service policy map to the output interface and enables LLQ for Frame Relay.

Verifying Configuration of Policy Maps and Their Classes

To display the contents of a specific policy map or all policy maps configured on an interface, use the following commands in EXEC mod, as needed:

Command	Purpose
`Router# show frame-relay pvc dlci`	Displays statistics about the PVC and the configuration of classes for the policy map on the specified DLCI.
`Router# show policy-map interface interface-name`	When FRTS is configured, displays the configuration of classes for all Frame Relay VC-level policy maps. When FRTS is not configured, displays the configuration of classes for the interface-level policy.
`Router# show policy-map interface interface-name dlci dlci`	When FRTS is configured, displays the configuration of classes for the policy map on the specified DLCI.

Monitoring and Maintaining LLQ for Frame Relay

For a list of commands that can be used to monitor LLQ for Frame Relay, see the previous section "Verifying Configuration of Policy Maps and Their Classes."

Configuring Burst Size in LLQ Configuration Task List

To configure the burst size in LLQ, perform the tasks described in the following sections. The tasks in the first two sections are required; the task in the remaining section is optional.

- Configuring the LLQ Bandwidth, page 28 (Required)
- Configuring the LLQ Burst Size, page 28 (Required)
- Verifying the LLQ Burst Size, page 28 (Optional)

See the end of this chapter for "Burst Size in LLQ Configuration Examples."

Configuring the LLQ Bandwidth

To configure the LLQ bandwidth, use the following command in policy-map class configuration mode:

Command	Purpose
`Router(config)# priority bandwidth`	Specifies the maximum amount of bandwidth, in kpbs, for the priority traffic.

Configuring the LLQ Burst Size

To configure the LLQ burst size, use the following command in policy-map class configuration mode:

Command	Purpose
`Router(config)# priority bandwidth burst`	Specifies the burst size in bytes. The range is from 32 to 2 million.

Verifying the LLQ Burst Size

To verify the LLQ burst size, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router# show policy-map`	Displays the configuration of all classes comprising the specified service policy map or all classes for all existing policy maps.
`Router# show policy-map interface`	Displays the configuration of classes configured for service polices on the specified interface or PVC.

Per-VC Hold Queue Support for ATM Adapters Configuration Task List

To configure the per-VC hold queue support for ATM adapters, perform the tasks described in the following sections. The task in the first section is required; the task in the remaining section is optional.

- Configuring the per-VC Hold Queue on an ATM Adapter, page 29 (Required)
- Verifying the Configuration of the per-VC Hold Queue on an ATM Adapter, page 29 (Optional)

See the end of this chapter for "Per-VC Hold Queue Support for ATM Adapters Examples."

For related information about per-VC and ATM configurations, see the "IP to ATM Class of Service Overview" and "Configuring IP to ATM Class of Service" modules.

Configuring the per-VC Hold Queue on an ATM Adapter

To configure the per-VC hold queue on an ATM adapter, use the following command in global configuration mode:

Command	Purpose
`Router(config)# vc-hold-queue number-of-packets`	Specifies the number of packets contained in the per-VC hold queue. This can be a number from 5 to 1024.

Verifying the Configuration of the per-VC Hold Queue on an ATM Adapter

To verify the configuration of the per-VC hold queue on an ATM adapter, use the following command in EXEC mode:

Command	Purpose
`Router# show queueing interface`	Displays the queueing statistics of an interface or VC.

Flow-Based WFQ Configuration Examples

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

```
Router(config)# interface Serial 3/0
Router(config-if)# ip unnumbered Ethernet 0/0
Router(config-if)# fair-queue 64 512 18
```

For information on how to configure WFQ, see the section "Flow-Based Weighted Fair Queueing Configuration Task List" in this chapter.

DWFQ Configuration Examples

The following sections provide DWFQ configuration examples:

For information on how to configure DWFQ, see the section "Distributed Weighted Fair Queueing Configuration Task List" in this chapter.

Flow-Based DWFQ : Example

The following example enables DWFQ on the HSSI interface 0/0/0:

```
Router(config)# interface Hssi0/0/0
Router(config-if)# description 45Mbps to R2
```

```
Router(config-if)# ip address 200.200.14.250 255.255.255.252
Router(config-if)# fair-queue
```

The following is sample output from the **show interfaces fair-queue** command for this configuration:

```
Router# show interfaces hssi 0/0/0 fair-queue

Hssi0/0/0 queue size 0
     packets output 35, drops 0
WFQ: global queue limit 401, local queue limit 200
```

QoS-Group-Based DWF: Example

The following example configures QoS-group-based DWFQ. Committed access rate (CAR) policies are used to assign packets with an IP Precedence value of 2 to QoS group 2, and packets with an IP Precedence value of 6 are assigned to QoS group 6.

```
Router(config)# interface Hssi0/0/0
Router(config-if)# ip address 188.1.3.70 255.255.255.0
Router(config-if)# rate-limit output access-group rate-limit 6 155000000 2000000 8000000
conform-action set-qos-transmit 6 exceed-action drop
Router(config-if)# rate-limit output access-group rate-limit 2 155000000 2000000 8000000
conform-action set-qos-transmit 2 exceed-action drop
Router(config-if)# fair-queue qos-group
Router(config-if)# fair-queue qos-group 2 weight 10
Router(config-if)# fair-queue qos-group 2 limit 27
Router(config-if)# fair-queue qos-group 6 weight 30
Router(config-if)# fair-queue qos-group 6 limit 27
!
Router(config)# access-list rate-limit 2 2
Router(config)# access-list rate-limit 6 6
```

The following sample output shows how to view WFQ statistics using the **show interfaces fair-queue** command:

```
Router# show interfaces fair-queue

 Hssi0/0/0 queue size 0
        packets output 806232, drops 1
 WFQ: aggregate queue limit 54, individual queue limit 27
    max available buffers 54

     Class 0: weight 60 limit 27 qsize 0 packets output 654 drops 0
     Class 2: weight 10 limit 27 qsize 0 packets output 402789 drops 0
     Class 6: weight 30 limit 27 qsize 0 packets output 402789 drops 1
```

ToS-Based DWFQ: Example

The following example configures type of service (ToS)-based DWFQ using the default parameters:

```
Router# configure terminal
Router(config)# interface Hssi0/0/0
Router(config-if)# fair-queue tos
Router(config-if)# end
```

The following is output of the **show running-config** command for the HSSI interface 0/0/0. Notice that the router automatically adds the default weights and limits for the ToS classes to the configuration.

```
interface Hssi0/0/0
```

```
 ip address 188.1.3.70 255.255.255.0
 fair-queue tos
 fair-queue tos 1 weight 20
 fair-queue tos 1 limit 27
 fair-queue tos 2 weight 30
 fair-queue tos 2 limit 27
 fair-queue tos 3 weight 40
 fair-queue tos 3 limit 27
```

The following sample output shows how to view DWFQ statistics using the **show interfaces fair-queue** command:

```
Router# show interfaces fair-queue

 Hssi0/0/0 queue size 0
        packets output 1417079, drops 2
 WFQ: aggregate queue limit 54, individual queue limit 27
    max available buffers 54

     Class 0: weight 10 limit 27 qsize 0 packets output 1150 drops 0
     Class 1: weight 20 limit 27 qsize 0 packets output 0 drops 0
     Class 2: weight 30 limit 27 qsize 0 packets output 775482 drops 1
     Class 3: weight 40 limit 27 qsize 0 packets output 0 drops 0
```

CBWFQ Configuration Examples

The following sections provide CBWFQ configuration examples:

- Class Map Configuration: Example, page 31
- Policy Creation: Example, page 32
- Policy Attachment to Interfaces: Example, page 32
- CBWFQ Using WRED Packet Drop: Example, page 32
- Display Service Policy Map Content Examples, page 33

For information on how to configure CBWFQ, see the section "Class-Based Weighted Fair Queueing Configuration Task List" in this chapter.

Class Map Configuration: Example

In the following example, ACLs 101 and 102 are created. Next, two class maps are created and their match criteria are defined. For the first map class, called class1, the numbered ACL 101 is used as the match criterion. For the second map class, called class2, the numbered ACL 102 is used as the match criterion. Packets are checked against the contents of these ACLs to determine if they belong to the class.

```
Router(config)# access-list 101 permit udp host 10.10.10.10 host 10.10.10.20 range 16384
20000
Router(config# access-list 102 permit udp host 10.10.10.10 host 10.10.10.20 range 53000
56000

Router(config)# class-map class1
Router(config-cmap)# match access-group 101
Router(config-cmap)# exit

Router(config-cmap)# class-map class2
Router(config-cmap)# match access-group 102
Router(config-cmap)# exit
```

Policy Creation: Example

In the following example, a policy map called policy1 is defined to contain policy specification for the two classes, class1 and class2. The match criteria for these classes were defined in the previous "Class Map Configuration: Example" section.

For class1, the policy specifies the bandwidth allocation request and the maximum number of packets that the queue for this class can accumulate. For class2, the policy specifies only the bandwidth allocation request, so the default queue limit of 64 packets is assumed.

```
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
```

Policy Attachment to Interfaces: Example

The following example shows how to attach an existing policy map. After you define a policy map, you can attach it to one or more interfaces to specify the service policy for those interfaces. Although you can assign the same policy map to multiple interfaces, each interface can have only one policy map attached at the input and one policy map attached at the output.

The policy map in this example was defined in the previous section, "Policy Creation: Example."

```
Router(config)# interface e1/1
Router(config-if)# service output policy1
Router(config-if)# exit

Router(config)# interface fa1/0/0
Router(config-if)# service output policy1
Router(config-if)# exit
```

CBWFQ Using WRED Packet Drop: Example

In the following example, the class map called class1 is created and defined to use the input FastEthernet interface 0/1 as a match criterion to determine if packets belong to the class. Next, the policy map policy1 is defined to contain policy specification for class1, which is configured for WRED packet drop.

```
Router(config)# class-map class1
Router(config-cmap)# match input-interface FastEthernet0/1
!
Router(config)# policy-map policy1
Router(config-pmap)# class class1
Router(config-pmap-c)# bandwidth 1000
Router(config-pmap-c)# random-detect
!
Router(config)# interface serial0/0
Router(config-if)# service-policy output policy1
!
```

Display Service Policy Map Content Examples

The following examples show how to display the contents of service policy maps. Four methods can be used to display the contents.

- Display all classes that make up a specified service policy map
- Display all classes configured for all service policy maps
- Display a specified class of a service policy map
- Display all classes configured for all service policy maps on a specified interface

All Classes for a Specified Service Policy Map

The following example displays the contents of the service policy map called pol1:

```
Router# show policy-map po1

Policy Map po1
 Weighted Fair Queueing
    Class class1
       Bandwidth 937 (kbps) Max thresh 64 (packets)
    Class class2
        Bandwidth 937 (kbps)  Max thresh 64 (packets)
    Class class3
        Bandwidth 937 (kbps)  Max thresh 64 (packets)
    Class class4
        Bandwidth 937 (kbps)  Max thresh 64 (packets)
    Class class5
        Bandwidth 937 (kbps)  Max thresh 64 (packets)
    Class class6
        Bandwidth 937 (kbps)  Max thresh 64 (packets)
    Class class7
        Bandwidth 937 (kbps)  Max thresh 64 (packets)
    Class class8
         Bandwidth 937 (kbps)  Max thresh 64 (packets)
```

All Classes for All Service Policy Maps

The following example displays the contents of all policy maps on the router:

```
Router# show policy-map

Policy Map poH1
 Weighted Fair Queueing
    Class class1
       Bandwidth 937 (kbps) Max thresh 64 (packets)
    Class class2
        Bandwidth 937 (kbps)  Max thresh 64 (packets)
    Class class3
        Bandwidth 937 (kbps)  Max thresh 64 (packets)
    Class class4
        Bandwidth 937 (kbps)  Max thresh 64 (packets)
    Class class5
        Bandwidth 937 (kbps)  Max thresh 64 (packets)
    Class class6
        Bandwidth 937 (kbps)  Max thresh 64 (packets)
    Class class7
        Bandwidth 937 (kbps)  Max thresh 64 (packets)
    Class class8
         Bandwidth 937 (kbps)  Max thresh 64 (packets)
```

```
Policy Map policy2
 Weighted Fair Queueing
    Class class1
       Bandwidth 300 (kbps) Max thresh 64 (packets)
    Class class2
        Bandwidth 300  (kbps)  Max thresh 64 (packets)
    Class class3
        Bandwidth 300 (kbps)  Max thresh 64 (packets)
    Class class4
        Bandwidth 300 (kbps)  Max thresh 64 (packets)
    Class class5
        Bandwidth 300 (kbps)  Max thresh 64 (packets)
    Class class6
        Bandwidth 300 (kbps)  Max thresh 64 (packets)
```

Specified Class for a Service Policy Map

The following example displays configurations for the class called class7 that belongs to the policy map called po1:

```
Router# show policy-map po1 class class7

Class class7
 Bandwidth 937 (kbps) Max Thresh 64 (packets)
```

All Classes for All Service Policy Maps on a Specified Interface

The following example displays configurations for classes on the output Ethernet interface 2/0. The numbers shown in parentheses are for use with the Management Information Base (MIB).

```
Router# show policy-map interface e2/0

Ethernet2/0

  Service-policy output:p1 (1057)

    Class-map:c1 (match-all) (1059/2)
      19 packets, 1140 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match:ip precedence 0  (1063)
      Weighted Fair Queueing
        Output Queue:Conversation 265
        Bandwidth 10 (%) Max Threshold 64 (packets)
        (pkts matched/bytes matched) 0/0
        (depth/total drops/no-buffer drops) 0/0/0

    Class-map:c2 (match-all) (1067/3)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match:ip precedence 1  (1071)
      Weighted Fair Queueing
        Output Queue:Conversation 266
        Bandwidth 10 (%) 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) (1075/0)
      8 packets, 2620 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match:any  (1079)
```

Distributed CBWFQ Configuration Examples

The following sections provide DCBWFQ configuration examples:

For information on how to configure DCBWFQ, see the section "Distributed Class-Based Weighted Fair Queueing Configuration Task List" in this chapter.

Traffic Class Configuration: Example

In the following example, two traffic classes are created and their match criteria are defined. For the first traffic class, called class1, the numbered ACL 101 is used as the match criterion. For the second traffic class, called class2, the numbered ACL 102 is used as the match criterion. Packets are checked against the contents of these ACLs to determine if they belong to the traffic class.

```
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
```

For additional information on traffic classes, see the "Applying QoS Features Using the MQC" module.

Traffic Policy Creation: Example

In the following example, a traffic policy called policy1 is defined to associate QoS features with the two traffic classes, class1 and class2. The match criteria for these traffic classes were defined in the previous "Class Map Configuration: Example" section.

For class1, the QoS policies include bandwidth allocation request and maximum packet count limit for the queue reserved for the traffic class. For class2, the policy specifies only a bandwidth allocation request, so the default queue limit of 64 packets is assumed.

```
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)# exit

Router(config-pmap)# class class2
Router(config-pmap-c)# bandwidth 2000
Router(config-pmap)# exit
```

For additional information on traffic policy configurations, see the "Applying QoS Features Using the MQC" module.

Traffic Policy Attachment to an Interface: Example

The following example shows how to attach an existing traffic policy to an interface. After you define a traffic policy, you can attach it to one or more interfaces to specify a traffic policy for those interfaces. Although you can assign the same traffic policy to multiple interfaces, each interface can have only one traffic policy attached at the input and one policy map attached at the output at one time.

```
Router(config)# interface fe1/0/0
Router(config-if)# service output policy1
Router(config-if)# exit
```

For additional information on attaching traffic policy configurations to interfaces, see the "Applying QoS Features Using the MQC" module.

IP RTP Priority Configuration Examples

The following sections provide IP RTP Priority configuration examples:

For information on how to configure IP RTP Priority, see the section "IP RTP Priority Configuration Task List" in this chapter.

CBWFQ Configuration: Example

The following example first defines a CBWFQ configuration and then reserves a strict priority queue:

```
! The following commands define a class map:
Router(config)# class-map class1
Router(config-cmap)# match access-group 101
Router(config-cmap)# exit

! The following commands create and attach a policy map:
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)# random-detect
Router(config-pmap-c)# random-detect precedence 0 32 256 100
Router(config-pmap-c)# exit
Router(config)# interface Serial1
Router(config-if)# service-policy output policy1

! The following command reserves a strict priority queue:
Router(config-if)# ip rtp priority 16384 16383 40
```

The **queue-limit** and **random-detect** commands are optional commands for CBWFQ configurations. The **queue-limit** command is used forconfiguring tail drop limits for a class queue. The **random-detect** command is used for configuring RED drop limits for a class queue, similar to the **random-detect** command available on an interface.

Virtual Template Configuration: Example

The following example configures a strict priority queue in a virtual template configuration with CBWFQ. The **max-reserved-bandwidth** command changes the maximum reserved bandwidth allocated for CBWFQ and IP RTP Priority from the default (75 percent) to 80 percent.

```
Router(config)# multilink virtual-template 1
Router(config)# interface virtual-template 1
Router(config-if)# ip address 172.16.1.1 255.255.255.0
Router(config-if)# no ip directed-broadcast
Router(config-if)# ip rtp priority 16384 16383 25
Router(config-if)# service-policy output policy1
Router(config-if)# ppp multilink
Router(config-if)# ppp multilink fragment-delay 20
Router(config-if)# ppp multilink interleave
Router(config-if)# max-reserved-bandwidth 80
Router(config-if)# end

Router(config)# interface Serial0/1
Router(config-if)# bandwidth 64
Router(config-if)# ip address 1.1.1.2 255.255.255.0
Router(config-if)# no ip directed-broadcast
Router(config-if)# encapsulation ppp
Router(config-if)# ppp multilink
Router(config-if)# end
```

Note To make the virtual access interface function properly, the **bandwidth** policy-map class configuration command should not be configured on the virtual template. It needs to be configured on the actual interface, as shown in the example.

Multilink Bundle Configuration: Example

The following example configures a strict priority queue in a multilink bundle configuration with WFQ. The advantage to using multilink bundles is that you can specify different ip rtp priority parameters on different interfaces.

The following commands create multilink bundle 1, which is configured for a maximum ip rtp priority bandwidth of 200 kbps. The **max-reserved-bandwidth** command changes the maximum reserved bandwidth allocated for WFQ and IP RTP Priority.

```
Router(config)# interface multilink 1
Router(config-if)# ip address 172.17.254.161 255.255.255.248
Router(config-if)# no ip directed-broadcast
Router(config-if)# ip rtp priority 16384 16383 200
Router(config-if)# no ip mroute-cache
Router(config-if)# fair-queue 64 256 0
Router(config-if)# ppp multilink
Router(config-if)# ppp multilink fragment-delay 20
Router(config-if)# ppp multilink interleave
Router(config-if)# max-reserved-bandwidth 80
```

The following commands create multilink bundle 2, which is configured for a maximum ip rtp priority bandwidth of 100 kbps:

```
Router(config)# interface multilink 2
Router(config-if)# ip address 172.17.254.162 255.255.255.248
Router(config-if)# no ip directed-broadcast
Router(config-if)# ip rtp priority 16384 16383 100
```

```
Router(config-if)# no ip mroute-cache
Router(config-if)# fair-queue 64 256 0
Router(config-if)# ppp multilink
Router(config-if)# ppp multilink fragment-delay 20
Router(config-if)# ppp multilink interleave
```

In the next part of the example, the **multilink-group** command configures serial interface 2/0 to be part of multilink bundle 1:

```
Router(config)# interface serial 2/0
Router(config-if)# bandwidth 256
Router(config-if)# no ip address
Router(config-if)# no ip directed-broadcast
Router(config-if)# encapsulation ppp
Router(config-if)# no ip mroute-cache
Router(config-if)# no fair-queue
Router(config-if)# clockrate 256000
Router(config-if)# ppp multilink
Router(config-if)# multilink-group 1
```

Next, serial interface 2/1 is configured to be part of multilink bundle 2.

```
Router(config)# interface serial 2/1
Router(config-if)# bandwidth 128
Router(config-if)# no ip address
Router(config-if)# no ip directed-broadcast
Router(config-if)# encapsulation ppp
Router(config-if)# no ip mroute-cache
Router(config-if)# no fair-queue
Router(config-if)# clockrate 128000
Router(config-if)# ppp multilink
Router(config-if)# multilink-group 2
```

Debug: Example

The following example shows sample output from the **debug priority** command. In this example, 64 indicates the actual priority queue depth at the time the packet was dropped.

```
Router# debug priority

*Feb 28 16:46:05.659:WFQ:dropping a packet from the priority queue 64
*Feb 28 16:46:05.671:WFQ:dropping a packet from the priority queue 64
*Feb 28 16:46:05.679:WFQ:dropping a packet from the priority queue 64
*Feb 28 16:46:05.691:WFQ:dropping a packet from the priority queue 64
*Feb 28 16:46:05.699:WFQ:dropping a packet from the priority queue 64
*Feb 28 16:46:05.711:WFQ:dropping a packet from the priority queue 64
*Feb 28 16:46:05.719:WFQ:dropping a packet from the priority queue 64
```

Frame Relay IP RTP Priority Configuration Examples

This "Strict Priority Service to Matching RTP Packets: Example" section provides a configuration example.

For information on how to configure Frame Relay IP RTP Priority queueing, see the section "Frame Relay IP RTP Priority Configuration Task List" in this chapter.

Strict Priority Service to Matching RTP Packets: Example

The following example first configures the Frame Relay map class called voip and then applies the map class to PVC 100 to provide strict priority service to matching RTP packets. In this example, RTP packets on PVC 100 with UDP ports in the range 16384 to 32764 will be matched and given strict priority service.

```
map-class frame-relay voip
 frame-relay cir 256000
 frame-relay bc 2560
 frame-relay be 600
 frame-relay mincir 256000
 no frame-relay adaptive-shaping
 frame-relay fair-queue
 frame-relay fragment 250
 frame-relay ip rtp priority 16384 16380 210

interface Serial5/0
 ip address 10.10.10.10 255.0.0.0
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 load-interval 30
 clockrate 1007616
 frame-relay traffic-shaping
 frame-relay interface-dlci 100
  class voip
 frame-relay ip rtp header-compression
 frame-relay intf-type dce
```

Frame Relay PVC Interface PQ Configuration Examples

This section provides configuration examples for Frame Relay PIPQ.

For information on how to configure Frame Relay PIPQ, see the section "Frame Relay PVC Interface Priority Configuration Task List" in this chapter.

This example shows the configuration of four PVCs on serial interface 0. DLCI 100 is assigned high priority, DLCI 200 is assigned medium priority, DLCI 300 is assigned normal priority, and DLCI 400 is assigned low priority.

The following commands configure Frame Relay map classes with PVC priority levels:

```
Router(config)# map-class frame-relay HI
Router(config-map-class)# frame-relay interface-queue priority high
Router(config-map-class)# exit
Router(config)# map-class frame-relay MED
Router(config-map-class)# frame-relay interface-queue priority medium
Router(config-map-class)# exit
Router(config)# map-class frame-relay NORM
Router(config-map-class)# frame-relay interface-queue priority normal
Router(config-map-class)# exit
Router(config)# map-class frame-relay LOW
Router(config-map-class)# frame-relay interface-queue priority low
Router(config-map-class)# exit
```

The following commands enable Frame Relay encapsulation and Frame Relay PIPQ on serial interface 0. The sizes of the priority queues are set at a maximum of 20 packets for the high priority queue, 40 for the medium priority queue, 60 for the normal priority queue, and 80 for the low priority queue.

```
Router(config)# interface Serial0
Router(config-if)# encapsulation frame-relay
Router(config-if)# frame-relay interface-queue priority 20 40 60 80
```

The following commands assign priority to four PVCs by associating the DLCIs with the configured map classes:

```
Router(config-if)# frame-relay interface-dlci 100
Router(config-fr-dlci)# class HI
Router(config-fr-dlci)# exit
Router(config-if)# frame-relay interface-dlci 200
Router(config-fr-dlci)# class MED
Router(config-fr-dlci)# exit
Router(config-if)# frame-relay interface-dlci 300
Router(config-fr-dlci)# class NORM
Router(config-fr-dlci)# exit
Router(config-if)# frame-relay interface-dlci 400
Router(config-fr-dlci)# class LOW
Router(config-fr-dlci)# exit
```

LLQ Configuration Examples

The following sections provide LLQ configuration examples:

- ATM PVC Configuration: Example, page 40
- Virtual Template Configuration: Example, page 41
- Multilink Bundle Configuration: Example, page 37

For information on how to configure LLQ, see the section "Low Latency Queueing Configuration Task List" in this chapter.

ATM PVC Configuration: Example

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

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

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

Next, the class map voice is defined, and the policy map called policy1 is created; a strict priority queue for the class voice is reserved, a bandwidth of 20 kbps is configured for the class bar, and the default class is configured for WFQ. The **service-policy** command then attaches the policy map to the PVC interface 0/102 on the subinterface atm1/0.2.

```
Router(config)# class-map voice
Router(config-cmap)# match access-group 102

Router(config)# policy-map policy1
Router(config-pmap)# class voice
Router(config-pmap-c)# priority 50
Router(config-pmap)# class bar
Router(config-pmap-c)# bandwidth 20
Router(config-pmap)# class class-default
```

```
Router(config-pmap-c)# fair-queue

Router(config)# interface atm1/0.2
Router(config-subif)# pvc 0/102
Router(config-subif-vc)# service-policy output policy1
```

Virtual Template Configuration: Example

The following example configures a strict priority queue in a virtual template configuration with CBWFQ. Traffic on virtual template 1 that is matched by access list 102 will be directed to the strict priority queue.

First, the class map voice is defined, and the policy map called policy1 is created. A strict priority queue (with a guaranteed allowed bandwidth of 50 kbps) is reserved for the class called voice.

```
Router(config)# class-map voice
Router(config-cmap)# match access-group 102
Router(config)# policy-map policy1
Router(config-pmap)# class voice
Router(config-pmap-c)# priority 50
```

Next, the **service-policy** command attaches the policy map called policy1 to virtual template 1.

```
Router(config)# multilink virtual-template 1
Router(config)# interface virtual-template 1
Router(config-if)# ip address 172.16.1.1 255.255.255.0
Router(config-if)# no ip directed-broadcast
Router(config-if)# service-policy output policy1
Router(config-if)# ppp multilink
Router(config-if)# ppp multilink fragment-delay 20
Router(config-if)# ppp multilink interleave
Router(config-if)# end

Router(config)# interface serial 2/0
Router(config-if)# bandwidth 256
Router(config-if)# no ip address
Router(config-if)# no ip directed-broadcast
Router(config-if)# encapsulation ppp
Router(config-if)# no fair-queue
Router(config-if)# clockrate 256000
Router(config-if)# ppp multilink
```

Multilink Bundle Configuration: Example

The following example configures a strict priority queue in a multilink bundle configuration with CBWFQ. Traffic on serial interface 2/0 that is matched by access list 102 will be directed to the strict priority queue. The advantage to using multilink bundles is that you can specify different **priority** parameters on different interfaces. To specify different **priority** parameters, you would configure two multilink bundles with different parameters.

First, the class map voice is defined, and the policy map called policy1 is created. A strict priority queue (with a guaranteed allowed bandwidth of 50 kbps) is reserved for the class called voice.

```
Router(config)# class-map voice
Router(config-cmap)# match access-group 102
Router(config)# policy-map policy1
Router(config-pmap)# class voice
Router(config-pmap-c)# priority 50
```

The following commands create multilink bundle 1. The policy map called policy1 is attached to the bundle by the **service-policy** command.

```
Router(config)# interface multilink 1
Router(config-if)# ip address 172.17.254.161 255.255.255.248
Router(config-if)# no ip directed-broadcast
Router(config-if)# no ip mroute-cache
Router(config-if)# service-policy output policy1
Router(config-if)# ppp multilink
Router(config-if)# ppp multilink fragment-delay 20
Router(config-if)# ppp multilink interleave
```

In the next part of the example, the **multilink-group** command configures serial interface 2/0 to be part of multilink bundle 1, which effectively directs traffic on serial interface 2/0 that is matched by access list 102 to the strict priority queue:

```
Router(config)# interface serial 2/0
Router(config-if)# bandwidth 256
Router(config-if)# no ip address
Router(config-if)# no ip directed-broadcast
Router(config-if)# encapsulation ppp
Router(config-if)# no fair-queue
Router(config-if)# clockrate 256000
Router(config-if)# ppp multilink
Router(config-if)# multilink-group 1
```

Distributed LLQ Configuration Examples

The following sections provide distributed LLQ configuration examples:

- Enabling PQ for an Amount of Available Bandwidth on an ATM Subinterface: Example, page 42
- Enabling PQ for a Percentage of Available Bandwidth on an ATM Subinterface: Example, page 43
- Limiting the Transmission Ring Limit on an ATM Interface: Example, page 44
- Limiting the Transmission Ring Limit on an ATM PVC Subinterface: Example, page 44

For information on how to configure distributed LLQ, see the section "Distributed LLQ Configuration Task List" in this chapter.

Enabling PQ for an Amount of Available Bandwidth on an ATM Subinterface: Example

The **priority** command can be enabled on an ATM subinterface, and that subinterface must have only one enabled ATM PVC. This configuration provides a sufficient amount of ATM PVC support.

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

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

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

Next, the traffic class called voice is defined, and the policy map called policy1 is created; a priority queue for the class voice is reserved with a guaranteed allowed bandwidth of 50 kpbs and an allowable burst size of 60 bytes, a bandwidth of 20 kbps is configured for the class called bar, and the default class is configured for flow-based fair queuing. The **service-policy** command then attaches the policy map to the PVC interface 0/102 on the subinterface atm1/0.

```
Router(config)# class-map voice
Router(config-cmap)# match access-group 102

Router(config)# policy-map policy1
Router(config-pmap)# class voice
Router(config-pmap-c)# priority 50 60
Router(config-pmap)# class bar
Router(config-pmap-c)# bandwidth 20
Router(config-pmap)# class class-default
Router(config-pmap-c)# fair-queue

Router(config)# interface atm1/0
Router(config-subif)# pvc 0/102

Router(config-subif)# service-policy output policy1
```

Enabling PQ for a Percentage of Available Bandwidth on an ATM Subinterface: Example

The **priority percent** command can be enabled on an ATM subinterface, and that subinterface must have only one enabled ATM PVC. This configuration provides a sufficient amount of ATM PVC support.

In the following example, a priority queue with a guaranteed allowed bandwidth percentage of 15 percent is reserved for traffic that is sent from the source address 10.10.10.10 to the destination address 10.10.10.20, in the range of ports 16384 through 20000 and 53000 through 56000.

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

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

Next, the traffic class called voice is defined, and the policy map called policy1 is created; a priority queue for the class voice is reserved with a guaranteed allowed bandwidth percentage of 15 percent, a bandwidth percentage of 20 percent is configured for the class called bar, and the default class is configured for flow-based fair queueing. The **service-policy** command then attaches the policy map to the ATM subinterface 1/0.2.

```
Router(config)# class-map voice
Router(config-cmap)# match access-group 102

Router(config)# policy-map policy1
Router(config-pmap)# class voice
Router(config-pmap-c)# priority percent 15
Router(config-pmap)# class bar
Router(config-pmap-c)# bandwidth percent 20
Router(config-pmap)# class class-default
Router(config-pmap-c)# fair-queue

Router(config)# interface atm1/0.2
Router(config-subif)# service-policy output policy1
```

Limiting the Transmission Ring Limit on an ATM Interface: Example

In the following example, the number of particles on the transmission ring of an ATM interface is limited to seven particles:

```
Router(config)# interface atm 1/0/0
Router(config-if)# atm pvc 32 0 32 tx-ring-limit 7
```

Limiting the Transmission Ring Limit on an ATM PVC Subinterface: Example

In the following example, the number of particles on the transmission ring of an ATM PVC subinterface is limited to ten particles:

```
Router(config)# interface ATM1/0/0.1 point-to-point
Router(config-subif)# pvc 2/200
Router(config-if-atm-vc)# tx-ring-limit 10
```

The **tx-ring-limit** command can be applied to several ATM PVC subinterfaces on a single interface. Every individual PVC can configure a transmission ring limit.

LLQ for Frame Relay Configuration Examples

The following section provides a LLQ for Frame Relay configuration examples.

For information on how to configure LLQ for Frame Relay, see the section "Low Latency Queueing for Frame Relay Configuration Task List" in this chapter.

The following example shows how to configure a PVC shaped to a 64K CIR with fragmentation. The shaping queue is configured with a class for voice, two data classes for IP precedence traffic, and a default class for best-effort traffic. WRED is used as the drop policy on one of the data classes.

The following commands define class maps and the match criteria for the class maps:

```
!
class-map voice
 match access-group 101
!
class-map immediate-data
 match access-group 102
!
class-map priority-data
 match access-group 103

!
access-list 101 permit udp any any range 16384 32767
access-list 102 permit ip any any precedence immediate
access-list 103 permit ip any any precedence priority
```

The following commands create and define a policy map called mypolicy:

```
!
policy-map mypolicy
 class voice
  priority 16
 class immediate-data
  bandwidth 32
  random-detect
 class priority-data
  bandwidth 16
```

```
 class class-default
  fair-queue 64
  queue-limit 20
```

The following commands enable Frame Relay fragmentation and attach the policy map to DLCI 100:

```
!
interface Serial1/0.1 point-to-point
 frame-relay interface-dlci 100
   class fragment
!
map-class frame-relay fragment
 frame-relay cir 64000
 frame-relay mincir 64000
 frame-relay bc 640
 frame-relay fragment 50
 service-policy output mypolicy
```

Burst Size in LLQ Configuration Examples

For information on how to configure the burst size in LLQ, see the section "Configuring Burst Size in LLQ Configuration Task List" in this chapter.

The following example configures the burst parameter to 1250 bytes for the class called Voice, which has an assigned bandwidth of 1000 kbps:

```
policy policy1
  class Voice
   priority 1000 1250
```

Per-VC Hold Queue Support for ATM Adapters Examples

For information on how to configure per-VC hold queue support for ATM Adapters, see the section "Per-VC Hold Queue Support for ATM Adapters Configuration Task List" in this chapter.

The following example sets the per-VC hold queue to 55:

```
interface atm2/0.1
 pvc 1/101
  vc-hold-queue 55
```

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

Low Latency Queueing (LLQ) for IPSec Encryption Engines

Feature History

Release	Modification
12.2(13)T	This feature was introduced.
12.2(14)S	This feature was integrated into Cisco IOS Release 12.2(14)S.

This feature module describes the Low Latency Queueing (LLQ) for IPSec encryption engines feature in Cisco IOS Release 12.2(13)T and 12.2(14)S. It includes the following sections:

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Feature Overview

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

Benefits

The Low Latency Queueing (LLQ) for IPSec encryption engines feature guarantees a certain level of crypto engine processing time for priority designated traffic.

Note On the Cisco 2600 platform, with the exception of the Cisco 2691 router, the CPU utilization maximizes out before the crypto engine becomes congested, so latency is not improved.

Better Voice Performance

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

Improved Latency and Jitters

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

Restrictions

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

Related Features and Technologies

- CBWFQ
- Priority Queueing
- Weighted Fair Queueing

Related Documents

- *Quality of Service Solutions Command Reference*
- "Configuring Weighted Fair Queueing" module

Supported Platforms

12.2(14)S and higher

The LLQ for IPSec encryption engines feature is supported on the following platform:

- Cisco 7200 series

12.2(13)T

The LLQ for IPSec encryption engines feature is supported on all platforms using Cisco IOS Release 12.2(13)T or later, including:

- Cisco 2600 series
- Cisco 3600 series
- Cisco 7100 series
- Cisco 7200 series

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that are supported on specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side-by-side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

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

Availability of Cisco IOS Software Images

Platform support for particular Cisco IOS software releases is dependent on the availability of the software images for those platforms. Software images for some platforms may be deferred, delayed, or changed without prior notice. For updated information about platform support and availability of software images for each Cisco IOS software release, see the online release notes or, if supported, Cisco Feature Navigator.

Supported Standards, MIBs, and RFCs

Standards

- No new or modified standards are supported by this feature.

MIBs

- No new or modified standards are supported by this feature.

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

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

- No new or modified RFCs are supported by this feature.

Prerequisites

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

- Access control lists
- Bandwidth management
- CBWFQ

Configuration Tasks

To configure LLQ for IPSec encryption engines, perform the tasks described in the following section.

Note See the"Applying QoS Features Using the MQC" module to learn more about configuring policy maps on interfaces.

- Defining Class Maps (required)
- Configuring Class Policy in the Policy Map (required)
- Configuring Class Policy for a Priority Queue (required)
- Configuring Class Policy Using a Specified Bandwidth (optional)
- Configuring the Class-Default Class Policy (optional)
- Attaching the Service Policy (required)
- Verifying Configuration of Policy Maps and Their Classes (optional)

Defining Class Maps

To create a class map containing match criteria against which a packet is checked to determine if it belongs to a class, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# class-map class-map-name`	Specifies the name of the class map to be created.
Step 2	`Router(config-cmap)# match access-group {access-group | name access-group-name}` or `Router(config-cmap)# match input-interface interface-name` or `Router(config-cmap)# match protocol protocol`	Specifies the name of the access control list (ACL) against whose contents packets are checked to determine if they belong to the class. Specifies the name of the input interface used as a match criterion against which packets are checked to determine if they belong to the class. Specifies the name of the protocol used as a match criterion against which packets are checked to determine if they belong to the class.

Configuring Class Policy in the Policy Map

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

- **priority**
- **bandwidth**

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

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

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

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

To configure class policies in a policy map, perform the tasks described in the following sections. The task in the first section is required; the tasks in the remaining sections are optional.

Configuring Class Policy for a Priority Queue

To configure a policy map and give priority to a class within the policy map, use the following commands beginning in global configuration mode:

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

Configuring Class Policy Using a Specified Bandwidth

To configure a policy map and create class policies that make up the service policy, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# policy-map policy-map`	Specifies the name of the policy map to be created or modified.
Step 2	`Router(config-cmap)# class class-name`	Specifies the name of a class to be created and included in the service policy.
Step 3	`Router(config-pmap-c)# bandwidth bandwidth-kbps`	Specifies the amount of bandwidth to be assigned to the class, in kbps, or as a percentage of the available bandwidth. Bandwidth must be specified in kbps or as a percentage consistently across classes. (Bandwidth of the priority queue must be specified in kbps.)

To configure more than one class in the same policy map, repeat Step 2 and Step 3.

Configuring the Class-Default Class Policy

The class-default class is used to classify traffic that does not fall into one of the defined classes. Even though the class-default class is predefined when you create the policy map, you still have to configure it. If a default class is not configured, then traffic that does not match any of the configured classes is given best-effort treatment, which means that the network will deliver the traffic if it can, without any assurance of reliability, delay prevention, or throughput.

To configure a policy map and the class-default class, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# policy-map policy-map`	Specifies the name of the policy map to be created or modified.
Step 2	`Router(config-cmap)# class class-default default-class-name`	Specifies the default class so that you can configure or modify its policy.
Step 3	`Router(config-pmap-c)# bandwidth bandwidth-kbps` `or` `Router(config-pmap-c)# fair-queue [number-of-dynamic-queues]`	Specifies the amount of bandwidth, in kbps, to be assigned to the class. Specifies the number of dynamic queues to be reserved for use by flow-based WFQ running on the default class. The number of dynamic queues is derived from the bandwidth of the interface.

Attaching the Service Policy

To attach a service policy to the output interface and enable LLQ for IPSec encryption engines, use the following command in map-class configuration mode:

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

Verifying Configuration of Policy Maps and Their Classes

To display the contents of a specific policy map or all policy maps configured on an interface, use the following commands in EXEC mode, as needed:

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

Monitoring and Maintaining LLQ for IPSec Encryption Engines

To monitor and maintain LLQ for IPSec encryption engines, use the following command in EXEC mode:

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

For a more detailed list of commands that can be used to monitor LLQ for IPSec encryption engines, see the section "Verifying Configuration of Policy Maps and Their Classes"

Configuration Examples

This section provides the following configuration example:

- LLQ for IPSec Encryption Engines Example

LLQ for IPSec Encryption Engines Example

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

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

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

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

```
Router(config)# class-map voice
Router(config-cmap)# match access-group 102
Router(config)# policy-map policy1
Router(config-pmap)# class voice
Router(config-pmap-c)# priority 50
```

```
Router(config-pmap)# class bar
Router(config-pmap-c)# bandwidth 20
Router(config-pmap)# class class-default
Router(config-pmap-c)# fair-queue
Router(config)# interface fastethernet0/0
Router(config-if)# service-policy output policy1
```

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Quality of Service Solutions Command Reference* at http://www.cisco.com/en/US/docs/ios/qos/command/reference/qos_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

- **show crypto eng qos**

Glossary

IKE—Internet Key Exchange. IKE establishes a shared security policy and authenticates keys for services (such as IPSec). Before any IPSec traffic can be passed, each router/firewall/host must verify the identity of its peer. This can be done by manually entering preshared keys into both hosts or by a CA service.

IPSec—IP Security. A framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPSec provides these security services at the IP layer. IPSec uses IKE to handle the negotiation of protocols and algorithms based on local policy and to generate the encryption and authentication keys to be used by IPSec. IPSec can protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

Configuring Custom Queueing

This chapter describes the tasks for configuring QoS custom queueing (CQ) on a router.

For complete conceptual information, see the "Congestion Management Overview" module.

For a complete description of the CQ commands in this chapter, see the *Cisco IOS Quality of Service Solutions Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

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

Note CQ is not supported on any tunnels.

Custom Queueing Configuration Task List

You must follow certain required, basic steps to enable CQ for your network. In addition, you can choose to assign packets to custom queues based on protocol type, interface where the packets enter the router, or other criteria you specify.

To configure CQ, perform the tasks described in the following sections. The tasks in first and third sections are required; the tasks in the remaining sections are optional.

- Defining the Custom Queue List (Required)
- Specifying the Maximum Size of the Custom Queues (Optional)
- Assigning Packets to Custom Queues (Required)
- Monitoring Custom Queue Lists (Optional)

See the end of this chapter for the section "Custom Queueing Configuration Examples."

Defining the Custom Queue List

To assign a custom queue list to an interface, use the following commands beginning in global configuration mode:

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

	Command	Purpose
Step 1	`Router(config)# interface interface-type interface-number`	Specifies the interface, and then enters interface configuration mode.
Step 2	`Router(config-if)# custom-queue-list list`	Assigns a custom queue list to the interface. The list argument is any number from 1 to 16. There is no default assignment.

Note Use the **custom-queue-list** command in place of the **priority-list** command. Only one queue list can be assigned per interface.

CQ allows a fairness not provided with priority queueing (PQ). With CQ, you can control the available bandwidth on an interface when it is unable to accommodate the aggregate traffic enqueued. Associated with each output queue is a configurable byte count, which specifies how many bytes of data should be delivered from the current queue by the system before the system moves on to the next queue. When a particular queue is being processed, packets are sent until the number of bytes sent exceeds the queue byte count defined by the **queue-list queue byte-count** command (see the following section "Specifying the Maximum Size of the Custom Queues"), or until the queue is empty.

Specifying the Maximum Size of the Custom Queues

You can specify the maximum number of packets allowed in each of the custom queues. The default is 20 entries.

You can also specify the approximate number of bytes to be forwarded from each queue during its turn in the cycle. The number is used as an average number, because whole packets must be forwarded.

To specify the approximate number of bytes to be forwarded from each queue during its turn in the cycle, use the following commands in global configuration mode, as needed:

Command	Purpose
`Router(config)# queue-list list-number queue queue-number limit limit-number`	Specifies the maximum number of packets allowed in each of the custom queues. The *limit-number* argument specifies the number of packets that can be queued at any one time. The range is from 0 to 32767.
`Router(config)# queue-list list-number queue queue-number byte-count byte-count-number`	Designates the average number of bytes forwarded per queue. The *byte-count-number* argument specifies the average number of bytes the system allows to be delivered from a given queue during a particular cycle.

Assigning Packets to Custom Queues

You can assign packets to custom queues based on the protocol type or interface where the packets enter the router. Additionally, you can set the default queue for packets that do not match other assignment rules. You can also specify multiple rules.

To define the CQ lists, use the following commands in global configuration mode, as needed:

Command	Purpose
`Router(config)# queue-list list-number protocol protocol-name queue-number queue-keyword keyword-value`	Establishes queueing priorities based on the protocol type.
`Router(config)# queue-list list-number interface interface-type interface-number queue-number`	Establishes CQ based on packets entering from a given interface.
`Router(config)# queue-list list-number default queue-number`	Assigns a queue number for those packets that do not match any other rule in the custom queue list.

All protocols supported by Cisco are allowed. The *queue-keyword* variable provides additional options, including byte count, TCP service and port number assignments, and AppleTalk, IP, IPX, VINES, or XNS access list assignments. Refer to the **queue-list protocol** command syntax description in the *Cisco IOS Quality of Service Solutions Command Reference.*

When you use multiple rules, remember that the system reads the **queue-list** commands in order of appearance. When classifying a packet, the system searches the list of rules specified by **queue-list** commands for a matching protocol or interface type. When a match is found, the packet is assigned to the appropriate queue. The list is searched in the order it is specified, and the first matching rule terminates the search.

Monitoring Custom Queue Lists

To display information about the input and output queues when CQ is enabled on an interface, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router# show queue interface-type interface-number`	Displays the contents of packets inside a queue for a particular interface or virtual circuit (VC).
`Router# show queueing custom`	Displays the status of the CQ lists.
`Router# show interfaces interface-type interface-number`	Displays the current status of the custom output queues when CQ is enabled.

Custom Queueing Configuration Examples

The following sections provide custom queueing examples:

- Custom Queue List Defined Example
- Maximum Specified Size of the Custom Queues Examples
- Packets Assigned to Custom Queues Examples

For information on how to configure CQ, see the section "Custom Queueing Configuration Task List" in this chapter.

Custom Queue List Defined Example

The following example illustrates how to assign custom queue list number 3 to serial interface 0:

```
interface serial 0
custom-queue-list 3
```

Maximum Specified Size of the Custom Queues Examples

The following example specifies the maximum number of packets allowed in each custom queue. The queue length of queue 10 is increased from the default 20 packets to 40 packets.

```
queue-list 3 queue 10 limit 40
```

The queue length limit is the maximum number of packets that can be enqueued at any time, with the range being from 0 to 32767 queue entries.

The following example decreases queue list 9 from the default byte count of 1500 to 1400 for queue number 10:

```
queue-list 9 queue 10 byte-count 1400
```

The byte count establishes the lowest number of bytes the system allows to be delivered from a given queue during a particular cycle.

Packets Assigned to Custom Queues Examples

The following examples assign packets to custom queues by either protocol type or interface type, and the default assignment for unmatched packets.

Protocol Type

The following example assigns traffic that matches IP access list 10 to queue number 1:

```
queue-list 1 protocol ip 1 list 10
```

The following example assigns Telnet packets to queue number 2:

```
queue-list 4 protocol ip 2 tcp 23
```

The following example assigns User Datagram Protocol (UDP) Domain Name Service (DNS) packets to queue number 3:

```
queue-list 4 protocol ip 3 udp 53
```

Interface Type

In this example, queue list 4 establishes queueing priorities for packets entering on serial interface 0. The queue number assigned is 10.

```
queue-list 4 interface serial 0 10
```

Default Queue

You can specify a default queue for packets that do not match other assignment rules. In this example, the default queue for list 10 is set to queue number 2:

```
queue-list 10 default 2
```

Configuring Priority Queueing

Feature History

Release	Modification
Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.

This chapter describes the tasks for configuring priority queueing (PQ) on a router.

For complete conceptual information, see the "Congestion Management Overview" module.

For a complete description of the PQ commands in this chapter, see the *Cisco IOS Quality of Service Solutions Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

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

Priority Queueing Configuration Task List

To configure PQ, perform the tasks described in the following sections. The tasks inthe first two sections are required; the task in remaining section is optional.

- Defining the Priority List (Required)
- Assigning the Priority List to an Interface (Required)
- Monitoring Priority Queueing Lists (Optional)

See the end of this chapter for the section "Priority Queueing Configuration Examples."

Defining the Priority List

A priority list contains the definitions for a setof priority queues. Thepriority list specifies which queue a packet will be placed in and, optionally, the maximum length of the different queues.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

In order to perform queueing using a priority list, you must assign the list to an interface. The same priority list can be applied to multiple interfaces. Alternatively, you can create many different priority policies to apply to different interfaces.

To define a priority list, perform the tasks described in the following sections. The task in the first section is required; the task in the remaining section is optional.

- Assigning Packets to Priority Queues, page 2 (Required)
- Specifying the Maximum Size of the Priority Queues, page 2 (Optional)

Assigning Packets to Priority Queues

Assign packets to priority queues based on the following qualities:

- Protocol type
- Interface where the packets enter the router

You can specify multiple assignment rules. The **priority-list** commands are read in order of appearance until a matching protocol or interface type is found. When a match is found, the packet is assigned to the appropriate queue and the search ends. Packets that do not match other assignment rules are assigned to the default queue.

To specify which queue to place a packet in, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# priority-list list-number protocol protocol-name {high | medium | normal | low} queue-keyword keyword-value`	Establishes queueing priorities based on the protocol type.
Step 2	`Router(config)# priority-list list-number interface interface-type interface-number {high | medium | normal | low}`	Establishes queueing priorities for packets entering from a given interface.
Step 3	`Router(config)# priority-list list-number default {high | medium | normal | low}`	Assigns a priority queue for those packets that do not match any other rule in the priority list.

All protocols supported by Cisco are allowed. The *queue-keyword* argument provides additional options including byte count, TCP service and port number assignments, and AppleTalk, IP, IPX, VINES, or XNS access list assignments. Refer to the **priority-list protocol** command syntax description in the *Cisco IOS Quality of Service Solutions Command Reference*.

Specifying the Maximum Size of the Priority Queues

To specify the maximum number of packets allowed in each of the priority queues, use the following command in global configuration mode:

Command	Purpose
`Router(config)# priority-list list-number queue-limit [high-limit [medium-limit [normal-limit [low-limit]]]]`	Specifies the maximum number of packets allowed in each of the priority queues.

Use the **priority-list queue-limit** command for each priority list. The default queue limit arguments are listed in Table 1.

Table 1 Default Priority Queue Packet Limits

Priority Queue Argument	Packet Limits
high-limit	20
medium-limit 40	
normal-limit	60
low-limit 80	

Assigning the Priority List to an Interface

You can assign a priority list number to an interface. Only one list can be assigned per interface. To assign a priority group to an interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface interface-type interface-number`	Specifies the interface, and then enters interface configuration mode.
Step 2	`Router(config-if)# priority-group list-number`	Assigns a priority list number to the interface.

Monitoring Priority Queueing Lists

To display information about the input and output queues, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router# show queue interface-type interface-number`	Displays the contents of packets inside a queue for a particular interface or VC.
`Router# show queueing priority`	Displays the status of the priority queueing lists.

Priority Queueing Configuration Examples

The following sections provide PQ configuration examples:

- Priority Queueing Based on Protocol Type: Example, page 4
- Priority Queueing Based on Interface: Example, page 4
- Maximum Specified Size of the Priority Queue: Example, page 4
- Priority List Assigned to an Interface: Example, page 4
- Priority Queueing Using Multiple Rules: Example, page 4

For information on how to configure PQ, see the section "Priority Queueing Configuration Task List" in this module.

Priority Queueing Based on Protocol Type: Example

The following example establishes queueing based on protocol type. The example assigns 1 as the arbitrary priority list number, specifies IP as the protocol type, and assigns a high priority level to traffic that matches IP access list 10.

```
access-list 10 permit 239.1.1.0 0.0.0.255
priority-list 1 protocol ip high list 10
```

Priority Queueing Based on Interface: Example

The following example establishes queueing based on interface. The example sets any packet type entering on Ethernet interface 0 to a medium priority.

```
priority-list 3 interface ethernet 0 medium
```

Maximum Specified Size of the Priority Queue: Example

The following example changes the maximum number of packets in the high priority queue to 10. The medium-limit, normal, and low-limit queue sizes remain at their default 40-, 60-, and 80-packet limits.

```
priority-list 4 queue-limit 10 40 60 80
```

Priority List Assigned to an Interface: Example

The following example assigns priority group list 4 to serial interface 0:

```
interface serial 0
  priority-group 4
```

Note The **priority-group** *list-number* command is not available on ATM interfaces that do not support fancy queueing.

Priority Queueing Using Multiple Rules: Example

When classifying a packet, the system searches the list of rules specified by **priority-list** commands for a matching protocol type. The following example specifies four rules:

- DECnet packets with a byte count less than 200 are assigned a medium priority queue level.
- IP packets originating or destined to TCP port 23 are assigned a medium priority queue level.
- IP packets originating or destined to User Datagram Protocol (UDP) port 53 are assigned a medium priority queue level.
- All IP packets are assigned a high priority queue level.

Remember that when using multiple rules for a single protocol, the system reads the priority settings in the order of appearance.

```
priority-list 4 protocol decnet medium lt 200
priority-list 4 protocol ip medium tcp 23
priority-list 4 protocol ip medium udp 53
priority-list 4 protocol ip high
```

Congestion Avoidance

Congestion Avoidance Overview

Congestion avoidance techniques monitor network traffic loads in an effort to anticipate and avoid congestion at common network bottlenecks. Congestion avoidance is achieved through packet dropping. Among the more commonly used congestion avoidance mechanisms is Random Early Detection (RED), which is optimum for high-speed transit networks. Cisco IOS QoS includes an implementation of RED that, when configured, controls when the router drops packets. If you do not configure Weighted Random Early Detection (WRED), the router uses the cruder default packet drop mechanism called tail drop.

For an explanation of network congestion, see the chapter "Quality of Service Overview."

This chapter gives a brief description of the kinds of congestion avoidance mechanisms provided by the Cisco IOS QoS features. It discusses the following features:

- Tail drop. This is the default congestion avoidance behavior when WRED is not configured.
- WRED. WRED and distributed WRED (DWRED)—both of which are the Cisco implementations of RED—combine the capabilities of the RED algorithm with the IP Precedence feature. Within the section on WRED, the following related features are discussed:
  - Flow-based WRED. Flow-based WRED extends WRED to provide greater fairness to all flows on an interface in regard to how packets are dropped.
  - DiffServ Compliant WRED. DiffServ Compliant WRED extends WRED to support Differentiated Services (DiffServ) and Assured Forwarding (AF) Per HopBehavior (PHB). This feature enables customers toimplement AF PHB by coloring packets according to differentiated services code point (DSCP) values and then assigning preferential drop probabilities to those packets.

For information on how to configure WRED, DWRED, flow-based WRED, and DiffServ Compliant WRED, see the chapter "Configuring Weighted Random Early Detection" in this book.

Tail Drop

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

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Weighted Random Early Detection

This section gives a brief introduction to RED concepts and addresses WRED, the Cisco implementation of RED for standard Cisco IOS platforms.

WRED avoids the globalization problems that occur when tail drop is used as the congestion avoidance mechanism on the router. Global synchronization occurs as waves of congestion crest only to be followed by troughs during which the transmission link is not fully utilized. Global synchronization of TCP hosts, for example, can occur because packets are dropped all at once. Global synchronization manifests when multiple TCP hosts reduce their transmission rates in response to packet dropping, then increase their transmission rates once again when the congestion is reduced.

About Random Early Detection

The RED mechanism was proposed by Sally Floyd and Van Jacobson in the early 1990s to address network congestion in a responsive rather than reactive manner. Underlying the RED mechanism is the premise that most traffic runs on data transport implementations that are sensitive to loss and will temporarily slow down when some of their traffic is dropped. TCP, which responds appropriately—even robustly—to traffic drop by slowing down its traffic transmission, effectively allows the traffic-drop behavior of RED to work as a congestion-avoidance signalling mechanism.

TCP constitutes the most heavily used network transport. Given the ubiquitous presence of TCP, RED offers a widespread, effective congestion-avoidance mechanism.

In considering the usefulness of RED when robust transports such as TCP are pervasive, it is important to consider also the seriously negative implications of employing RED when a significant percentage of the traffic is not robust in response to packet loss. Neither Novell NetWare nor AppleTalk is appropriately robust in response to packet loss, therefore you should not use RED for them.

How It Works

RED aims to control the average queue size by indicating to the end hosts when they should temporarily slow down transmission of packets.

RED takes advantage of the congestion control mechanism of TCP. By randomly dropping packets prior to periods of high congestion, RED tells the packet source to decrease its transmission rate. Assuming the packet source is using TCP, it will decrease its transmission rate until all the packets reach their destination, indicating that the congestion is cleared. You can use RED as a way to cause TCP to slow down transmission of packets. TCP not only pauses, but it also restarts quickly and adapts its transmission rate to the rate that the network can support.

RED distributes losses in time and maintains normally low queue depth while absorbing spikes. When enabled on an interface, RED begins dropping packets when congestion occurs at a rate you select during configuration.

For an explanation of how the Cisco WRED implementation determines parameters to use in the WRED queue size calculations and how to determine optimum values to use for the weight factor, see the section "Average Queue Size" later in this chapter.

Packet Drop Probability

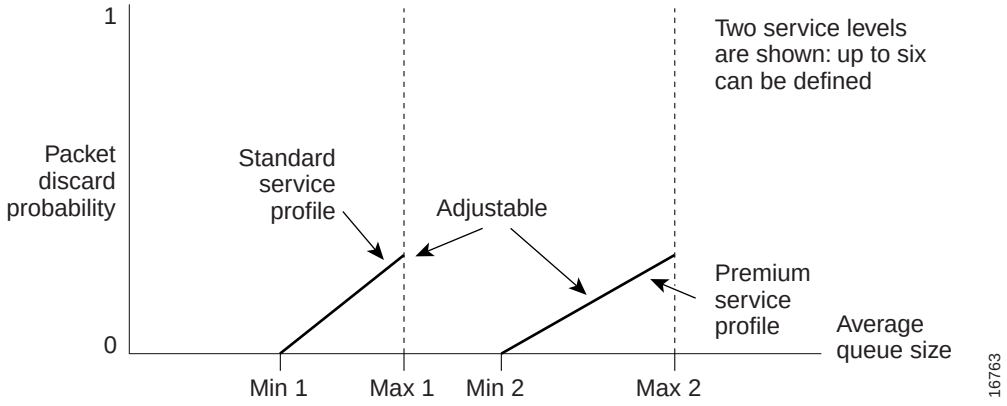
The packet drop probability is based on the minimum threshold, maximum threshold, and mark probability denominator.

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

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

When the average queue size is above the maximum threshold, all packets are dropped. Figure 1 summarizes the packet drop probability.

Figure 1 RED Packet Drop Probability



The minimum threshold value should be set high enough to maximize the 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 of TCP hosts (global synchronization of TCP hosts can occur as multiple TCP hosts reduce their transmission rates). If the difference between the maximum and minimum thresholds is too small, many packets may be dropped at once, resulting in global synchronization.

How TCP Handles Traffic Loss

Note The sections "How TCP Handles Traffic Loss" and "How the Router Interacts with TCP" contain detailed information that you need not read in order to use WRED or to have a general sense of the capabilities of RED. If you want to understand why problems of global synchronization occur in response to congestion when tail drop is used by default and how RED addresses them, read these sections.

When the recipient of TCP traffic—called the receiver—receives a data segment, it checks the four octet (32-bit) sequence number of that segment against the number the receiver expected, which would indicate that the data segment was received in order. If the numbers match, the receiver delivers all of the data that it holds to the target application, then it updates the sequence number to reflect the next number in order, and finally it either immediately sends an acknowledgment (ACK) packet to the sender or it schedules an ACK to be sent to the sender after a short delay. The ACK notifies the sender that the receiver received all data segments up to but not including the one marked with the new sequence number.

Receivers usually try to send an ACK in response to alternating data segments they receive; they send the ACK because for many applications, if the receiver waits out a small delay, it can efficiently include its reply acknowledgment on a normal response to the sender. However, when the receiver receives a data segment out of order, it immediately responds with an ACK to direct the sender to resend the lost data segment.

When the sender receives an ACK, it makes this determination: It determines if any data is outstanding. If no data is outstanding, the sender determines that the ACK is a keepalive, meant to keep the line active, and it does nothing. If data is outstanding, the sender determines whether the ACK indicates that the receiver has received some or none of the data. If the ACK indicates receipt of some data sent, the sender determines if new credit has been granted to allow it to send more data. When the ACK indicates receipt of none of the data sent and there is outstanding data, the sender interprets the ACK to be a repeatedly sent ACK. This condition indicates that some data was received out of order, forcing the receiver to remit the first ACK, and that a second data segment was received out of order, forcing the receiver to remit the second ACK. In most cases, the receiver would receive two segments out of order because one of the data segments had been dropped.

When a TCP sender detects a dropped data segment, it resends the segment. Then it adjusts its transmission rate to half of what is was before the drop was detected. This is the TCP back-off or slow-down behavior. Although this behavior is appropriately responsive to congestion, problems can arise when multiple TCP sessions are carried on concurrently with the same router and all TCP senders slow down transmission of packets at the same time.

How the Router Interacts with TCP

Note The sections "How TCP Handles Traffic Loss" and "How the Router Interacts with TCP" contain detailed information that you need not read in order to use WRED or to have a general sense of the capabilities of RED. If you want to understand why problems of global synchronization occur in response to congestion when tail drop is used by default and how RED addresses them, read these sections.

To see how the router interacts with TCP, we will look at an example. In this example, on average, the router receives traffic from one particular TCP stream every other, every 10th, and every 100th or 200th message in the interface in MAE-EAST or FIX-WEST. A router can handle multiple concurrent TCP sessions. Because network flows are additive, there is a high probability that when traffic exceeds the Transmit Queue Limit (TQL) at all, it will vastly exceed the limit. However, there is also a high probability that the excessive traffic depth is temporary and that traffic will not stay excessively deep except at points where traffic flows merge or at edge routers.

If the router drops all traffic that exceeds the TQL, as is done when tail drop is used by default, many TCP sessions will simultaneously go into slow start. Consequently, traffic temporarily slows down to the extreme and then all flows slow-start again; this activity creates a condition of global synchronization.

However, if the router drops no traffic, as is the case when queueing features such as fair queueing or custom queueing (CQ) are used, then the data is likely to be stored in main memory, drastically degrading router performance.

By directing one TCP session at a time to slow down, RED solves the problems described, allowing for full utilization of the bandwidth rather than utilization manifesting as crests and troughs of traffic.

About WRED

WRED combines the capabilities of the RED algorithm with the IP Precedence feature to provide for preferential traffic handling of higher priority packets. WRED can selectively discard lower priority traffic when the interface begins to get congested and provide differentiated performance characteristics for different classes of service.

You can configure WRED to ignore IP precedence when making drop decisions so that nonweighted RED behavior is achieved.

For interfaces configured to use the Resource Reservation Protocol (RSVP) feature, WRED chooses packets from other flows to drop rather than the RSVP flows. Also, IP Precedence governs which packets are dropped—traffic that is at a lower precedence has a higher drop rate and therefore is more likely to be throttled back.

WRED differs from other congestion avoidance techniques such as queueing strategies because it attempts to anticipate and avoid congestion rather than control congestion once it occurs.

Why Use WRED?

WRED makes early detection of congestion possible and provides for multiple classes of traffic. It also protects against global synchronization. For these reasons, WRED is useful on any output interface where you expect congestion to occur.

However, WRED is usually used in the core routers of a network, rather than at the edge of the network. Edge routers assign IP precedences to packets as they enter the network. WRED uses these precedences to determine how to treat different types of traffic.

WRED provides separate thresholds and weights for different IP precedences, allowing you to provide different qualities of service in regard to packet dropping for different traffic types. Standard traffic may be dropped more frequently than premium traffic during periods of congestion.

WRED is also RSVP-aware, and it can provide the controlled-load QoS service of integrated service.

How It Works

By randomly dropping packets prior to periods of high congestion, WRED tells the packet source to decrease its transmission rate. If the packet source is using TCP, it will decrease its transmission rate until all the packets reach their destination, which indicates that the congestion is cleared.

WRED generally drops packets selectively based on IP precedence. Packets with a higher IP precedence are less likely to be dropped than packets with a lower precedence. Thus, the higher the priority of a packet, the higher the probability that the packet will be delivered.

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. Thus, WRED allows the transmission line to be used fully at all times.

In addition, WRED statistically drops more packets from large users than small. Therefore, traffic sources that generate the most traffic are more likely to be slowed down than traffic sources that generate little traffic.

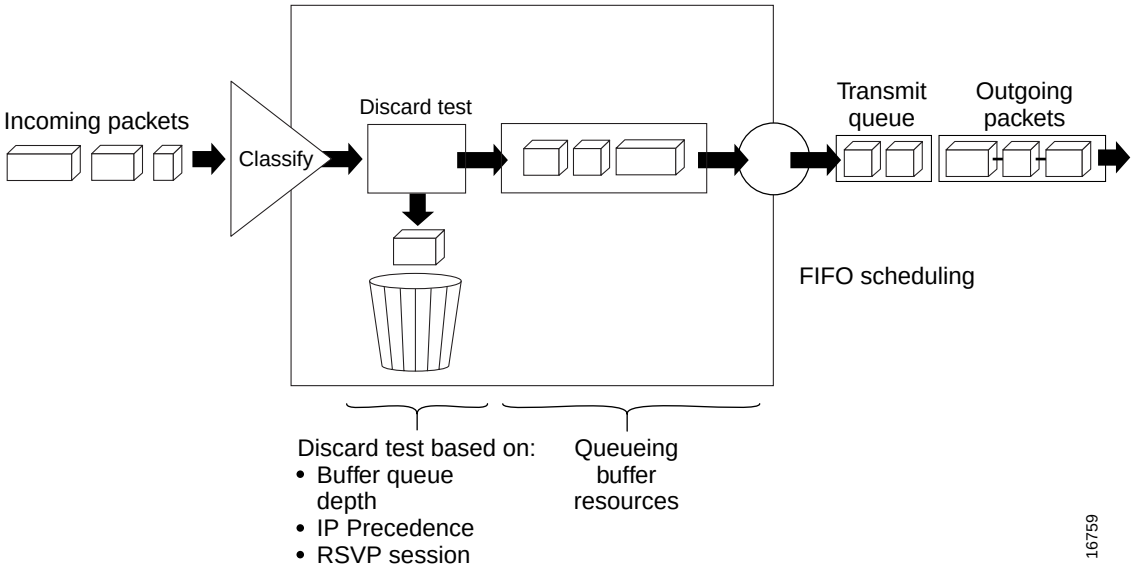
WRED avoids the globalization problems that occur when tail drop is used as the congestion avoidance mechanism. Global synchronization manifests when multiple TCP hosts reduce their transmission rates in response to packet dropping, then increase their transmission rates once again when the congestion is reduced.

WRED is only useful when the bulk of the traffic is TCP/IP traffic. With TCP, dropped packets indicate congestion, so the packet source will reduce its transmission rate. With other protocols, packet sources may not respond or may resend dropped packets at the same rate. Thus, dropping packets does not decrease congestion.

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.

Figure 2 illustrates how WRED works.

Figure 2 Weighted Random Early Detection



Average Queue Size

The router automatically determines parameters to use in the WRED calculations. The average queue size is based on the previous average and the current size of the queue. The formula is:

```
average = (old_average * (1-2 -n)) + (current_queue_size * 2 -n)
```

where *n* is the exponential weight factor, a user-configurable value. The default value of the exponential weight factor is 9. It is recommended to use only the default value for the exponential weight factor. Change this value from the default value only if you have determined that your scenario would benefit from using a different value.

For high values of *n*, the previous average becomes more important. A large factor smooths out the peaks and lows in queue length. The average queue size is unlikely to change very quickly, avoiding drastic swings in size. The WRED process will be slow to start dropping packets, but it may continue dropping packets for a time after the actual queue size has fallen below the minimum threshold. The slow-moving average will accommodate temporary bursts in traffic.

Note If the value of *n* gets too high, WRED will not react to congestion. Packets will be sent or dropped as if WRED were not in effect.

For low values of *n*, the average queue size closely tracks the current queue size. The resulting average may fluctuate with changes in the traffic levels. In this case, the WRED process responds quickly to long queues. Once the queue falls below the minimum threshold, the process will stop dropping packets.

If the value of *n* gets too low, WRED will overreact to temporary traffic bursts and drop traffic unnecessarily.

Restrictions

You cannot configure WRED on the same interface as Route Switch Processor (RSP)-based CQ, priority queueing (PQ), or weighted fair queueing (WFQ).

Distributed Weighted Random Early Detection

Distributed WRED (DWRED) is an implementation of WRED for the Versatile Interface Processor (VIP). DWRED provides the complete set of functions for the VIP that WRED provides on standard Cisco IOS platforms.

The DWRED feature is only supported on Cisco 7000 series routers with an RSP-based RSP7000 interface processor and Cisco 7500 series routers with a VIP-based VIP2-40 or greater interface processor. A VIP2-50 interface processor is strongly recommended when the aggregate line rate of the port adapters on the VIP is greater than DS3. A VIP2-50 interface processor is required for OC-3 rates.

DWRED is configured the same way as WRED. If you enable WRED on a suitable VIP interface, such as a VIP2-40 or greater with at least 2 MB of SRAM, DWRED will be enabled instead.

In order to use DWRED, distributed Cisco Express Forwarding (dCEF) switching must be enabled on the interface. For information about dCEF, see the "Cisco Express Forwarding Features Roadmap" module.

You can configure both DWRED and distributed weighted fair queueing (DWFQ) on the same interface, but you cannot configure distributed WRED on an interface for which RSP-based CQ, PQ, or WFQ is configured.

You can enable DWRED using the Modular Quality of Service Command-Line Interface (Modular QoS CLI) feature. For complete conceptual and configuration information on the Modular QoS CLI feature, see the "Applying QoS Features Using the MQC" module.

How It Works

When a packet arrives and DWRED is enabled, the following events occur:

- The average queue size is calculated. See the "Average Queue Size" section for details.
- If the average is less than the minimum queue threshold, the arriving packet is queued.
- If the average is between the minimum queue threshold and the maximum queue threshold, the packet is either dropped or queued, depending on the packet drop probability. See the "Packet-Drop Probability" section for details.
- If the average queue size is greater than the maximum queue threshold, the packet is automatically dropped.

Average Queue Size

The average queue size is based on the previous average and the current size of the queue. The formula is:

```
average = (old_average * (1-1/2^n)) + (current_queue_size * 1/2^n)
```

where *n* is the exponential weight factor, a user-configurable value.

For high values of *n*, the previous average queue size becomes more important. A large factor smooths out the peaks and lows in queue length. The average queue size is unlikely to change very quickly, avoiding drastic swings in size. The WRED process will be slow to start dropping packets, but it may continue dropping packets for a time after the actual queue size has fallen below the minimum threshold. The slow-moving average will accommodate temporary bursts in traffic.

Note If the value of *n* gets too high, WRED will not react to congestion. Packets will be sent or dropped as if WRED were not in effect.

For low values of *n*, the average queue size closely tracks the current queue size. The resulting average may fluctuate with changes in the traffic levels. In this case, the WRED process responds quickly to long queues. Once the queue falls below the minimum threshold, the process stops dropping packets.

If the value of *n* gets too low, WRED will overreact to temporary traffic bursts and drop traffic unnecessarily.

Packet-Drop Probability

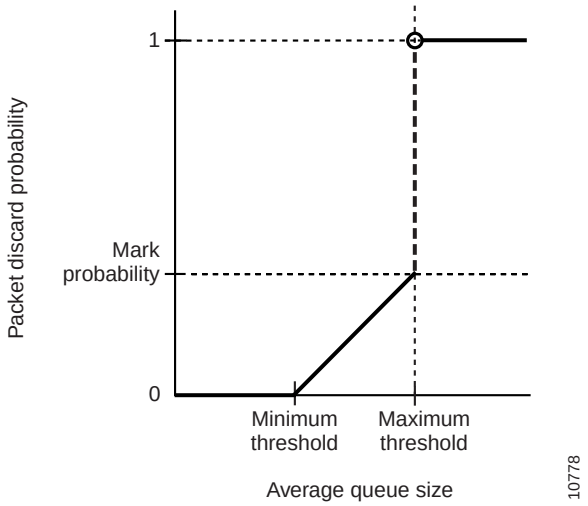
The probability that a packet will be dropped is based on the minimum threshold, maximum threshold, and mark probability denominator.

When the average queue size 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.

The mark probability denominator is the fraction of packets dropped when the average queue size 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.

When the average queue size is above the maximum threshold, all packets are dropped. Figure 3 summarizes the packet drop probability.

Figure 3 Packet Drop Probability



The minimum threshold value should be set high enough to maximize the 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 of TCP hosts (global synchronization of TCP hosts can occur as multiple TCP hosts reduce their transmission rates). If the difference between the maximum and minimum thresholds is too small, many packets may be dropped at once, resulting in global synchronization.

Why Use DWRED?

DWRED provides faster performance than does RSP-based WRED. You should run DWRED on the VIP if you want to achieve very high speed on the Cisco 7500 series platform—for example, you can achieve speed at the OC-3 rates by running WRED on a VIP2-50 interface processor.

Additionally, the same reasons you would use WRED on standard Cisco IOS platforms apply to using DWRED. (See the section "Why Use WRED?" earlier in this chapter.) For instance, when WRED or DWRED is not configured, tail drop is enacted during periods of congestion. Enabling DWRED obviates the global synchronization problems that result when tail drop is used to avoid congestion.

The DWRED feature provides the benefit of consistent traffic flows. When RED is not configured, output buffers fill during periods of congestion. When the buffers are full, tail drop occurs; all additional packets are dropped. Because the packets are dropped all at once, global synchronization of TCP hosts can occur as multiple TCP hosts reduce their transmission rates. The congestion clears, and the TCP hosts increase their transmission rates, resulting in waves of congestion followed by periods when the transmission link is not fully used.

RED 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 buffer is full, RED avoids dropping large numbers of packets at once and minimizes the chances of global synchronization. Thus, RED allows the transmission line to be used fully at all times.

In addition, RED statistically drops more packets from large users than small. Therefore, traffic sources that generate the most traffic are more likely to be slowed down than traffic sources that generate little traffic.

DWRED provides separate thresholds and weights for different IP precedences, allowing you to provide different qualities of service for different traffic. Standard traffic may be dropped more frequently than premium traffic during periods of congestion.

Restrictions

The following restrictions apply to the DWRED feature:

- Interface-based DWRED cannot be configured on a subinterface. (A subinterface is one of a number of virtual interfaces on a single physical interface.)
- DWRED is not supported on Fast EtherChannel and tunnel interfaces.
- RSVP is not supported on DWRED.
- DWRED is useful only when the bulk of the traffic is TCP/IP traffic. With TCP, dropped packets indicate congestion, so the packet source reduces its transmission rate. With other protocols, packet sources may not respond or may resend dropped packets at the same rate. Thus, dropping packets does not necessarily decrease congestion.
- DWRED treats non-IP traffic as precedence 0, the lowest precedence. Therefore, non-IP traffic is usually more likely to be dropped than IP traffic.
- DWRED cannot be configured on the same interface as RSP-based CQ, PQ, or WFQ. However, both DWRED and DWFQ can be configured on the same interface.

Note Do not use the **match protocol** command to create a traffic class with a non-IP protocol as a match criterion. The VIP does not support matching of non-IP protocols.

Prerequisites

This section provides the prerequisites that must be met before you configure the DWRED feature.

Weighted Fair Queueing

Attaching a service policy to an interface disables WFQ on that interface if WFQ is configured for the interface. For this reason, you should ensure that WFQ is not enabled on such an interface before configuring DWRED.

For information on WFQ, see the chapter "Configuring Weighted Fair Queueing" in this book.

WRED

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

Access Control Lists

You can specify a numbered access list as the match criterion for any traffic class that you create. For this reason, before configuring DWRED you should know how to configure access lists.

Cisco Express Forwarding

In order to use DWRED, dCEF switching must be enabled on the interface. For information on dCEF, see the "Cisco Express Forwarding Features Roadmap" module.

Flow-Based WRED

Flow-based WRED is a feature that forces WRED to afford greater fairness to all flows on an interface in regard to how packets are dropped.

Why Use Flow-Based WRED?

Before you consider the advantages that use of flow-based WRED offers, it helps to think about how WRED (without flow-based WRED configured) affects different kinds of packet flows. Even before flow-based WRED classifies packet flows, flows can be thought of as belonging to one of the following categories:

- Nonadaptive flows, which are flows that do not respond to congestion.
- Robust flows, which on average have a uniform data rate and slow down in response to congestion.
- Fragile flows, which, though congestion-aware, have fewer packets buffered at a gateway than do robust flows.

WRED tends toward bias against fragile flows because all flows, even those with relatively fewer packets in the output queue, are susceptible to packet drop during periods of congestion. Though fragile flows have fewer buffered packets, they are dropped at the same rate as packets of other flows.

To provide fairness to all flows, flow-based WRED has the following features:

- It ensures that flows that respond to WRED packet drops (by backing off packet transmission) are protected from flows that do not respond to WRED packet drops.
- It prohibits a single flow from monopolizing the buffer resources at an interface.

How It Works

Flow-based WRED relies on the following two main approaches to remedy the problem of unfair packet drop:

- It classifies incoming traffic into flows based on parameters such as destination and source addresses and ports.
- It maintains state about active flows, which are flows that have packets in the output queues.

Flow-based WRED uses this classification and state information to ensure that each flow does not consume more than its permitted share of the output buffer resources. Flow-based WRED determines which flows monopolize resources and it more heavily penalizes these flows.

To ensure fairness among flows, flow-based WRED maintains a count of the number of active flows that exist through an output interface. Given the number of active flows and the output queue size, flow-based WRED determines the number of buffers available per flow.

To allow for some burstiness, flow-based WRED scales the number of buffers available per flow by a configured factor and allows each active flow to have a certain number of packets in the output queue. This scaling factor is common to all flows. The outcome of the scaled number of buffers becomes the per-flow limit. When a flow exceeds the per-flow limit, the probability that a packet from that flow will be dropped increases.

DiffServ Compliant WRED

DiffServ Compliant WRED extends the functionality of WRED to enable support for DiffServ and AF Per Hop Behavior PHB. This feature enables customers to implement AF PHB by coloring packets according to DSCP values and then assigning preferential drop probabilities to those packets.

Note This feature can be used with IP packets only. It is not intended for use with Multiprotocol Label Switching (MPLS)-encapsulated packets.

The Class-Based Quality of Service MIB supports this feature. This MIB is actually the following two MIBs:

- CISCO-CLASS-BASED-QOS-MIB
- CISCO-CLASS-BASED-QOS-CAPABILITY-MIB

The DiffServ Compliant WRED feature supports the following RFCs:

- RFC 2474, *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*
- RFC 2475, *An Architecture for Differentiated Services Framework*
- RFC 2597, *Assured Forwarding PHB*
- RFC 2598, *An Expedited Forwarding PHB*

How It Works

The DiffServ Compliant WRED feature enables WRED to use the DSCP value when it calculates the drop probability for a packet. The DSCP value is the first six bits of the IP type of service (ToS) byte.

This feature adds two new commands, **random-detect dscp** and **dscp**. It also adds two new arguments, *dscp-based* and *prec-based*, to two existing WRED-related commands—the **random-detect** (interface) command and the **random-detect-group** command.

The *dscp-based* argument enables WRED to use the DSCP value of a packet when it calculates the drop probability for the packet. The *prec-based* argument enables WRED to use the IP Precedence value of a packet when it calculates the drop probability for the packet.

These arguments are optional (you need not use any of them to use the commands) but they are also mutually exclusive. That is, if youuse the *dscp-based* argument, you cannot use the *prec-based* argument with the same command.

After enabling WRED to use the DSCP value, you can then use the new **random-detect dscp** command to change the minimum and maximum packet thresholds for that DSCP value.

Three scenarios for using these arguments are provided.

Usage Scenarios

The new *dscp-based* and *prec-based* arguments can be used whether you are using WRED at the interface level, at the per-virtual circuit (VC) level, or at the class level (as part of class-based WFQ (CBWFQ) with policy maps).

WRED at the Interface Level

At the interface level, if you want to have WRED use the DSCP value when it calculates the drop probability, you can use the *dscp-based* argument with the **random-detect** (interface) command to specify the DSCP value. Then use the **random-detect dscp** command to specify the minimum and maximum thresholds for the DSCP value.

WRED at the per-VC Level

At the per-VC level, if you want to have WRED use the DSCP value when it calculates the drop probability, you can use the *dscp-based* argument with the **random-detect-group** command. Then use the **dscp** command to specify the minimum and maximum thresholds for the DSCP value or the mark-probability denominator.

This configuration can then be applied to each VC in the network.

WRED at the Class Level

If you are using WRED at the class level (with CBWFQ), the *dscp-based* and *prec-based* arguments can be used within the policy map.

First, specify the policy map, the class, and the bandwidth. Then, if you want WRED to use the DSCP value when it calculates the drop probability, use the *dscp-based* argument with the **random-detect** (interface) command to specify the DSCP value. Then use the **random-detect dscp** command to modify the default minimum and maximum thresholds for the DSCP value.

This configuration can then be applied wherever policy maps are attached (for example, at the interface level, the per-VC level, or the shaper level).

Usage Points to Note

Remember the following points when using the new commands and the new arguments included with this feature:

- If you use the *dscp-based* argument, WRED will use the DSCP value to calculate the drop probability.
- If you use the *prec-based* argument, WRED will use the IP Precedence value to calculate the drop probability.
- The *dscp-based* and *prec-based* arguments are mutually exclusive.
- If you do not specify either argument, WRED will use the IP Precedence value to calculate the drop probability (the default method).
- The **random-detect dscp** command must be used in conjunction with the **random-detect** (interface) command.
- The **random-detect dscp** command can only be used if you use the *dscp-based* argument with the **random-detect** (interface) command.
- The **dscp** command must be used in conjunction with the **random-detect-group** command.
- The **dscp** command can only be used if you use the *dscp-based* argument with the **random-detect-group** command.

For more information about using these commands, see the *Cisco IOS Quality of Service Solutions Command Reference*.

Configuring Weighted Random Early Detection

Configuring Weighted Random Early Detection

Feature History

Release	Modification
Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.

This chapter describes the tasks for configuring Weighted Random Early Detection (WRED), distributed WRED (DWRED), flow-based WRED, and DiffServ Compliant WRED on a router.

For complete conceptual information, see the "Congestion Avoidance Overview" module in this book.

For a complete description of the WRED and DWRED commands in this chapter, see the *Cisco IOS Quality of Service Solutions Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

The RSVP-ATM QoS Interworking and IP to ATM Class of Service features also use WRED. For information on how to configure these features with WRED, see the chapters "Configuring RSVP-ATM QoS Interworking" and "Configuring IP to ATM Class of Service" in this book.

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

Note WRED is useful with adaptive traffic such as TCP/IP. With TCP, dropped packets indicate congestion, so the packet source will reduce its transmission rate. With other protocols, packet sources may not respond or may resend dropped packets at the same rate. Thus, dropping packets does not decrease congestion.

WRED treats non-IP traffic as precedence 0, the lowest precedence. Therefore, non-IP traffic is more likely to be dropped than IP traffic.

You cannot configure WRED on the same interface as Route Switch Processor (RSP)-based custom queueing (CQ), priority queueing (PQ), or weighted fair queueing (WFQ). However, you can configure both DWRED and DWFQ on the same interface.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Weighted Random Early Detection Configuration Task List

Random Early Detection (RED) is a congestion avoidance mechanism that takes advantage of the congestion control mechanism of TCP. By randomly dropping packets prior to periods of high congestion, RED tells the packet source to decrease its transmission rate. WRED drops packets selectively based on IP precedence. Edge routers assign IP precedences to packets as they enter the network. (WRED is useful on any output interface where you expect to have congestion. However, WRED is usually used in the core routers of a network, rather than at the edge.) WRED uses these precedences to determine how it treats different types of traffic.

When a packet arrives, the following events occur:

1. The average queue size is calculated.
2. If the average is less than the minimum queue threshold, the arriving packet is queued.
3. If the average is between the minimum queue threshold for that type of traffic and the maximum threshold for the interface, the packet is either dropped or queued, depending on the packet drop probability for that type of traffic.
4. If the average queue size is greater than the maximum threshold, the packet is dropped.

See the "Congestion Avoidance Overview"module in this book for more details on the queue calculations and how WRED works.

To configure WRED on an interface, perform the tasks described in the following sections. The task in the first section is required; the tasks in the remaining sections are optional.

- Enabling WRED (Required)
- Changing WRED Parameters (Optional)
- Monitoring WRED (Optional)

See the end of this chapter for the section "WRED Configuration Examples."

Enabling WRED

To enable WRED, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# random-detect`	Enables WRED. If you configure this command on a Versatile Interface Processor (VIP) interface, DWRED is enabled.

You need not specify any other commands or parameters in order to configure WRED on the interface. WRED will use the default parameter values.

Changing WRED Parameters

To change WRED parameters, use the following commands in interface configuration mode, as needed:

Command	Purpose
`Router(config-if)# random-detect exponential-weighting-constant exponent`	Configures the weight factor used in calculating the average queue length.
`Router(config-if)# random-detect precedence precedence min-threshold max-threshold mark-prob-denominator`	Configures parameters for packets with a specific IP Precedence. The minimum threshold for IP Precedence 0 corresponds to half the maximum threshold for the interface. Repeat this command for each precedence. To configure RED, rather than WRED, use the same parameters for each precedence.

When you enable WRED with the **random-detect** interface configuration 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.

The default minimum threshold depends on the precedence. The minimum threshold for IP Precedence 0 corresponds to half of the maximum threshold. The values for the remaining precedences fall between half the maximum threshold and the maximum threshold at evenly spaced intervals.

Note The default WRED parameter values are based on the best available data. We recommend that you do not change the parameters from their default values unless you have determined that your applications will benefit from the changed values.

Monitoring WRED

To monitor WRED services in your network, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router# show queue interface-type interface-number`	Displays the header information of the packets inside a queue. This command does not support DWRED.
`Router# show queueing interface interface-number [vc [[vpi/] vci]]`	Displays the WRED statistics of a specific virtual circuit (VC) on an interface.
`Router# show queueing random-detect`	Displays the queueing configuration for WRED.
`Router# show interfaces [type slot | port-adapter | port]`	Displays WRED configuration on an interface.

DWRED Configuration Task List

To configure DWRED, perform the tasks described in the following sections. The tasks in the first two sections are required; the task in the remaining section is optional.

- Configuring DWRED in a Traffic Policy (Required)
- Configuring DWRED to Use IP Precedence Values in a Traffic Policy (Required)
- Monitoring and Maintaining DWRED (Optional)

See the end of this chapter for the section "DWRED Configuration Examples."

Configuring DWRED in a Traffic Policy

To configure DWRED in a traffic policy, use the **policy-map** command in global configuration mode to specify the traffic policy name. Then to configure the traffic policy, use the following commands in policy-map configuration mode:

	Command	Purpose
Step 1	`Router(config)# policy-map policy-map`	Specifies the name of the traffic policy to be created or modified.
Step 2	`Router(config-pmap)# class class-name`	Specifies the name of a traffic class to be created and included in the traffic policy
	Steps 3, 4, and 5 are optional. If you do not want to configure the exponential weight factor, specify the amount of bandwidth, or specify the number of queues to be reserved, you can skip these three steps and continue with step 6.	
Step 3	`Router(config-pmap-c)# random-detect exponential-weighting-constant exponent`	Configures the exponential weight factor used in calculating the average queue length.
Step 4	`Router(config-pmap-c)# bandwidth bandwidth-kbps`	Specifies the amount of bandwidth, in kbps, to be assigned to the traffic class.
Step 5	`Router(config-pmap-c)# fair-queue [queue-limit queue-values]`	Specifies the number of queues to be reserved for the traffic class.
Step 6	`Router(config-pmap-c)# queue-limit number-of-packets`	Specifies the maximum number of packets that can be queued for the specified traffic class.

The default traffic class for the traffic policy is the traffic class to which traffic is directed if that traffic does not satisfy the match criteria of other traffic classes whose policy is defined in the traffic policy. To configure a policy for more than one traffic class in the same policy map, repeat Step 2 through Step 4.

To attach a traffic policy to an interface and enable CBWFQ on the interface, you must create a traffic policy. You can configure traffic class policies for as many traffic classes as are defined on the router, up to the maximum of 64.

After configuring the traffic policy with the **policy-map** command, you must still attach the traffic policy to an interface before it is successfully enabled. For information on attaching a traffic policy to an interface, see the "Applying QoS Features Using the MQC" module.

Configuring DWRED to Use IP Precedence Values in a Traffic Policy

To configure DWRED to drop packets based on IP Precedence values, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# policy-map policy-map`	Specifies the name of the traffic policy to be created or modified.
Step 2	`Router(config-pmap)# class class-name`	Specifies the name of a traffic class to associate with the traffic policy
Step 3	`Router(config-pmap-c)# random-detect exponential-weighting-constant exponent`	Configures the exponential weight factor used in calculating the average queue length.
Step 4	`Router(config-pmap-c)# random-detect precedence precedence min-threshold max-threshold mark-prob-denominator`	Configures the parameters for packets with a specific IP Precedence. The minimum threshold for IP Precedence 0 corresponds to half the maximum threshold for the interface. Repeat this command for each precedence.

After configuring the traffic policy with the **policy-map** command, you must still attach the traffic policy to an interface before it is successfully enabled. For information on attaching a traffic policy to an interface, see the "Applying QoS Features Using the MQC" module.

Monitoring and Maintaining DWRED

To display the configuration of a traffic policy and its associated traffic classes, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router# show policy-map`	Displays all configured traffic policies.
`Router# show policy-map policy-map-name`	Displays the user-specified traffic policy.
`Router# show policy-map interface`	Displays statistics and configurations of all input and output policies attached to an interface.
`Router# show policy-map interface interface-spec`	Displays configuration and statistics of the input and output policies attached to a particular interface.
`Router# show policy-map interface interface-spec input`	Displays configuration and statistics of the input policy attached to an interface.
`Router# show policy-map interface interface-spec output`	Displays configuration statistics of the output policy attached to an interface.
`Router# show policy-map [interface [interface-spec [input | output] [class class-name]]]`	Displays the configuration and statistics for the class name configured in the policy.

Flow-Based WRED Configuration Task List

To configure flow-based WRED on an interface, perform the required task described in the "Configuring Flow-Based WRED" section.

See the end of this chapter for the section "Flow-Based WRED Configuration Example."

Configuring Flow-Based WRED

Before you can configure flow-based WRED, you must enable WRED and configure it. For information on how to configure WRED, see the section "Weighted Random Early Detection Configuration Task List" earlier in this chapter.

To configure an interface for flow-based WRED, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# random-detect flow`	Enables flow-based WRED.
Step 2	`Router(config-if)# random-detect flow average-depth-factor` *scaling-factor*	Sets the flow threshold multiplier for flow-based WRED.
Step 3	`Router(config-if)# random-detect flow count` *number*	Sets the maximum flow count for flow-based WRED.

DiffServ Compliant WRED Configuration Task List

To configure the DiffServ Compliant Weighted Random Early Detection feature, perform the tasks described in the following sections. The task in the first section is required; the task in the remaining section is optional.

- Configuring WRED to Use the Differentiated Services Code Point Value (Required)
- Verifying the DSCP Value Configuration (Optional)

See the end of this chapter for the section "DiffServ Compliant WRED Configuration Examples."

Configuring WRED to Use the Differentiated Services Code Point Value

The commands used to configure WRED to use the differentiated services code point (DSCP) value vary according to whether WRED is used at the interface level, the per-VC level, or the class level.

WRED at the Interface Level

To configure WRED to use the DSCP value when it calculates the drop probability, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# random-detect dscp-based`	Indicates that WRED is to use the DSCP value when it calculates the drop probability for the packet.
Step 2	`Router(config-if)# random-detect dscp dscpvalue min-threshold max-threshold [mark-probability-denominator]`	Specifies the minimum and maximum thresholds, and, optionally, the mark-probability denominator for the specified DSCP value.

WRED at the per-VC Level

To configure WRED to use the DSCP value when it calculates the drop probability, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# random-detect-group group-name dscp-based`	Indicates that WRED is to use the DSCP value when it calculates the drop probability for the packet.
Step 2	`Router(cfg-red-grp)# dscp dscpvalue min-threshold max-threshold [mark-probability-denominator]`	Specifies the DSCP value, the minimum and maximum packet thresholds and, optionally, the mark-probability denominator for the DSCP value.
Step 3	`Router(config-atm-vc)# random-detect [attach group-name]`	Enables per-VC WRED or per-VC VIP-DWRED.

WRED at the Class Level

To configure WRED to use the DSCP value when it calculates the drop probability, use the following commands beginning in interface configuration mode. These are the commands to use at the class level, within policy maps.

	Command	Purpose
Step 1	`Router(config-if)# class-map class-map-name`	Creates a class map to be used for matching packets to a specified class.
Step 2	`Router(config-cmap)# match match criterion`	Configures the match criteria for a class map. For more information about match criteria, see the "Applying QoS Features Using the MQC" module.
Step 3	`Router(config-if)# policy-map policy-map`	Creates or modifies a policy map that can be attached to one or more interfaces to specify a traffic policy.
Step 4	`Router(config-pmap)# class class-map-name`	Specifies the QoS actions for the default class.
Step 5	`Router(config-pmap-c)# bandwidth {bandwidth-kbps | percent percent}`	Specifies or modifies the bandwidth allocated for a class belonging to a policy map.

	Command	Purpose
Step 6	`Router(config-pmap-c)# random-detect dscp-based`	Indicates that WRED is to use the DSCP value when it calculates the drop probability for the packet.
Step 7	`Router(config-pmap-c)# random-detect dscp` *dscpvalue min-threshold max-threshold* [*mark-probability-denominator*]	Specifies the minimum and maximum packet thresholds and, optionally, the mark-probability denominator for the DSCP value.
Step 8	`Router(config-if)# service-policy output` *policy-map*	Attaches a policy map to an output interface or VC to be used as the traffic policy for that interface or VC.

Verifying the DSCP Value Configuration

To verify the DSCP value configuration, use the following commands in global configuration mode, as needed:

Command	Purpose
`Router# show queueing interface`	Displays the queueing statistics of an interface or VC.
`Router# show policy-map interface`	Displays the configuration of classes configured for traffic policies on the specified interface or permanent virtual circuit (PVC).

WRED Configuration Examples

The following sections provide WRED and DWRED configuration examples:

- WRED Configuration Example
- Parameter-Setting DWRED Example
- Parameter-Setting WRED Example

For information on how to configure WRED, see the section "Weighted Random Early Detection Configuration Task List" in this chapter.

WRED Configuration Example

The following example enables WRED with default parameter values:

```
interface Serial5/0
 description to qos1-75a
 ip address 200.200.14.250 255.255.255.252
 random-detect
```

Use the **show interfaces** command output to verify the configuration. Notice that the "Queueing strategy" report lists "random early detection (RED)."

```
Router# show interfaces serial 5/0
```

```
Serial5/0 is up, line protocol is up
  Hardware is M4T
  Description: to qos1-75a
  Internet address is 200.200.14.250/30
  MTU 1500 bytes, BW 128 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 237/255
  Encapsulation HDLC, crc 16, loopback not set
  Keepalive not set
  Last input 00:00:15, output 00:00:00, output hang never
  Last clearing of "show interface" counters 00:05:08
  Input queue: 0/75/0 (size/max/drops); Total output drops: 1036
  Queueing strategy: random early detection(RED)
  5 minutes input rate 0 bits/sec, 2 packets/sec
  5 minutes output rate 119000 bits/sec, 126 packets/sec
     594 packets input, 37115 bytes, 0 no buffer
     Received 5 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     37525 packets output, 4428684 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions     DCD=up  DSR=up  DTR=up  RTS=up  CTS=up
```

Use the **show queue** command output to view the current contents of the interface queue. Notice that there is only a single queue into which packets from all IP precedences are placed after dropping has taken place. The output has been truncated to show only three of the five packets.

```
Router# show queue serial 5/0

Output queue for Serial5/0 is 5/0

Packet 1, linktype: ip, length: 118, flags: 0x288
  source: 190.1.3.4, destination: 190.1.2.2, id: 0x0001, ttl: 254,
  TOS: 128 prot: 17, source port 11111, destination port 22222
    data: 0x2B67 0x56CE 0x005E 0xE89A 0xCBA9 0x8765 0x4321
          0x0FED 0xCBA9 0x8765 0x4321 0x0FED 0xCBA9 0x8765

Packet 2, linktype: ip, length: 118, flags: 0x288
  source: 190.1.3.5, destination: 190.1.2.2, id: 0x0001, ttl: 254,
  TOS: 160 prot: 17, source port 11111, destination port 22222
    data: 0x2B67 0x56CE 0x005E 0xE89A 0xCBA9 0x8765 0x4321
          0x0FED 0xCBA9 0x8765 0x4321 0x0FED 0xCBA9 0x8765

Packet 3, linktype: ip, length: 118, flags: 0x280
  source: 190.1.3.6, destination: 190.1.2.2, id: 0x0001, ttl: 254,
  TOS: 192 prot: 17, source port 11111, destination port 22222
    data: 0x2B67 0x56CE 0x005E 0xE89A 0xCBA9 0x8765 0x4321
          0x0FED 0xCBA9 0x8765 0x4321 0x0FED 0xCBA9 0x8765
```

Use the **show queueing** command output to view the current settings for each of the precedences. Also notice that the default minimum thresholds are spaced evenly between half and the entire maximum threshold. Thresholds are specified in terms of packet count.

```
Router# show queueing

Current random-detect configuration:
     Serial5/0
       Queueing strategy:random early detection (WRED)
       Exp-weight-constant:9 (1/512)
       Mean queue depth:28
```

```
  Class   Random       Tail    Minimum    Maximum     Mark
          drop         drop  threshold  threshold  probability
    0       330          0         20         40     1/10
    1       267          0         22         40     1/10
    2       217          0         24         40     1/10
    3       156          0         26         40     1/10
    4        61          0         28         40     1/10
    5         6          0         31         40     1/10
    6         0          0         33         40     1/10
    7         0          0         35         40     1/10
  rsvp        0          0         37         40     1/10
```

Parameter-Setting DWRED Example

The following example specifies the same parameters for each IP precedence. Thus, all IP precedences receive the same treatment. Start by enabling DWRED.

```
interface FastEthernet1/0/0
 ip address 200.200.14.250 255.255.255.252
 random-detect
```

Next, enter the **show queueing random-detect** command to determine reasonable values to use for the precedence-specific parameters.

```
Router# show queueing random-detect

Current random-detect configuration:
  FastEthernet2/0/0
    Queueing strategy:fifo
    Packet drop strategy:VIP-based random early detection (DWRED)
    Exp-weight-constant:9 (1/512)
    Mean queue depth:0
    Queue size:0       Maximum available buffers:6308
    Output packets:5  WRED drops:0  No buffer:0

    Class   Random       Tail    Minimum    Maximum     Mark       Output
              drop       drop  threshold  threshold  probability  Packets
      0          0          0        109        218      1/10            5
      1          0          0        122        218      1/10            0
      2          0          0        135        218      1/10            0
      3          0          0        148        218      1/10            0
      4          0          0        161        218      1/10            0
      5          0          0        174        218      1/10            0
      6          0          0        187        218      1/10            0
      7          0          0        200        218      1/10            0
```

Complete the configuration by assigning the same parameter values to each precedence. Use the values obtained from the **show queueing random-detect** command output to choose reasonable parameter values.

```
interface FastEthernet1/0/0
 random-detect precedence 0 100 218 10
 random-detect precedence 1 100 218 10
 random-detect precedence 2 100 218 10
 random-detect precedence 3 100 218 10
 random-detect precedence 4 100 218 10
 random-detect precedence 5 100 218 10
 random-detect precedence 6 100 218 10
 random-detect precedence 7 100 218 10
```

Parameter-Setting WRED Example

The following example enables WRED on the interface and specifies parameters for the different IP precedences:

```
interface Hssi0/0/0
 description 45Mbps to R1
 ip address 10.200.14.250 255.255.255.252
 random-detect
 random-detect precedence 0 32 256 100
 random-detect precedence 1 64 256 100
 random-detect precedence 2 96 256 100
 random-detect precedence 3 120 256 100
 random-detect precedence 4 140 256 100
 random-detect precedence 5 170 256 100
 random-detect precedence 6 290 256 100
 random-detect precedence 7 210 256 100
 random-detect precedence rsvp 230 256 100
```

DWRED Configuration Examples

The following sections provide DWRED configuration examples:

- DWRED on an Interface: Example, page 11
- Modular QoS CLI: Example, page 11
- Configuring DWRED in Traffic Policy: Example, page 12

For information on how to configure DWRED, see the section "DWRED Configuration Task List" in this chapter.

DWRED on an Interface: Example

The following example configures DWRED on an interface with a weight factor of 10:

```
Router(config)# interface hssi0/0/0
Router(config-if)# description 45mbps to R1
Router(config-if)# ip address 192.168.14.250 255.255.255.252
Router(config-if)# random-detect
Router(config-if)# random-detect exponential-weighting-constant 10
```

Modular QoS CLI: Example

The following example enables DWRED using the Legacy CLI (non-Modular QoS Command-Line Interface) feature on the interface and specifies parameters for the different IP precedences:

```
interface Hssi0/0/0
 description 45Mbps to R1
 ip address 200.200.14.250 255.255.255.252
 random-detect
 random-detect precedence 0 32 256 100
 random-detect precedence 1 64 256 100
 random-detect precedence 2 96 256 100
 random-detect precedence 3 120 256 100
 random-detect precedence 4 140 256 100
 random-detect precedence 5 170 256 100
```

```
 random-detect precedence 6 290 256 100
 random-detect precedence 7 210 256 100
 random-detect precedence rsvp 230 256 100
```

The following example uses the Modular QoS CLI to configure a traffic policy called policy10. For congestion avoidance, WRED packet drop is used, not tail drop. IP Precedence is reset for levels 0 through 5.

```
policy-map policy10
 class acl10
 bandwidth 2000
 random-detect exponential-weighting-constant 10
 random-detect precedence 0 32 256 100
 random-detect precedence 1 64 256 100
 random-detect precedence 2 96 256 100
 random-detect precedence 3 120 256 100
 random-detect precedence 4 140 256 100
 random-detect precedence 5 170 256 100
```

Configuring DWRED in Traffic Policy: Example

The following example configures policy for a traffic class named int10 to configure the exponential weight factor as 12. This is the weight factor used for the average queue size calculation for the queue for traffic class int10. WRED packet drop is used forcongestion avoidance for traffic class int10, not tail drop.

```
policy-map policy12
 class int10
 bandwidth 2000
 random-detect exponential-weighting-constant 12
```

Flow-Based WRED Configuration Example

The following example enables WRED on the serial interface 1 and configures flow-based WRED. The **random-detect** interface configuration command is used to enable WRED. Once WRED is enabled, the **random-detect flow** command is used to enable flow-based WRED.

After flow-based WRED is enabled, the **random-detect flow average-depth-factor** command is used to set the scaling factor to 8 and the **random-detect flow count** command is used to set the flow count to 16. The scaling factor is used to scale the number of buffers available per flow and to determine the number of packets allowed in the output queue for each active flow.

```
configure terminal
interface Serial1
 random-detect
 random-detect flow
 random-detect flow average-depth-factor 8
 random-detect flow count 16
end
```

The following part of the example shows a sample configuration file after the previous flow-based WRED commands are issued:

```
Router# more system:running-config

Building configuration...
Current configuration:
!
```

```
version 12.0
service timestamps debug datetime msec localtime
service timestamps log uptime
no service password-encryption
service tcp-small-servers
!
no logging console
enable password lab
!
clock timezone PST -8
clock summer-time PDT recurring
ip subnet-zero
no ip domain-lookup
!
interface Ethernet0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 shutdown
!
interface Serial0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 no keepalive
 shutdown
!
interface Serial1
 ip address 190.1.2.1 255.255.255.0
 no ip directed-broadcast
 load-interval 30
 no keepalive
 random-detect
 random-detect flow
 random-detect flow count 16
 random-detect flow average-depth-factor 8
!
router igrp 8
 network 190.1.0.0
!
ip classless
no ip http server
!
line con 0
 transport input none
line 1 16
 transport input all
line aux 0
 transport input all
line vty 0 4
 password lab
 login
!
end
```

DiffServ Compliant WRED Configuration Examples

This following sections provide DiffServ Compliant WRED configuration examples:

For information on how to configure DiffServ compliant WRED, see the section "DiffServ Compliant WRED Configuration Task List" in this chapter.

WRED Configured to Use the DSCP Value: Example

The following example configures WRED to use the DSCP value 8. The minimum threshold for the DSCP value 8 is 24 and the maximum threshold is 40. This configuration was performed at the interface level.

```
Router(config-if)# interface seo/0
Router(config-if)# random-detect dscp-based
Router(config-if)# random-detect dscp 8 24 40
```

The following example enables WRED to use the DSCP value 9. The minimum threshold for the DSCP value 9 is 20 and the maximum threshold is 50. This configuration can be attached to other VCs, as required.

```
Router(config)# random-detect-group sanjose dscp-based
Router(cfg-red-grp)# dscp 9 20 50
Router(config-subif-vc)# random-detect attach sanjose
```

The following example enables WRED to use the DSCP value 8 for the class c1. The minimum threshold for the DSCP value 8 is 24 and the maximum threshold is 40. The last line attaches the traffic policy to the output interface or VC p1.

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

DSCP Value Configuration Verification: Example

When WRED has been configured to use the DSCP value when it calculates the drop probability of a packet, all entries of the DSCP table are initialized with the appropriate default values. The example in the following section are samples of the **show policy interface** command for WRED at the class level.

This example displays packet statistics along with the entries of the DSCP table, confirming that WRED has been enabled to use the DSCP value when it calculates the drop probability for a packet.

```
Router# show policy interface Serial6/3

 Serial6/3

  Service-policy output: test

    Class-map: c1 (match-any)
      0 packets, 0 bytes
```

```
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: protocol ip
        0 packets, 0 bytes
        5 minute rate 0 bps
      Weighted Fair Queueing
        Output Queue: Conversation 265
        Bandwidth 20 (%)
        Bandwidth 308 (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
          pkts/bytes        pkts/bytes       pkts/bytes    thresh  thresh  prob
af11          0/0               0/0              0/0          32      40  1/10
af12          0/0               0/0              0/0          28      40  1/10
af13          0/0               0/0              0/0          24      40  1/10
af21          0/0               0/0              0/0          32      40  1/10
af22          0/0               0/0              0/0          28      40  1/10
af23          0/0               0/0              0/0          24      40  1/10
af31          0/0               0/0              0/0          32      40  1/10
af32          0/0               0/0              0/0          28      40  1/10
af33          0/0               0/0              0/0          24      40  1/10
af41          0/0               0/0              0/0          32      40  1/10
af42          0/0               0/0              0/0          28      40  1/10
af43          0/0               0/0              0/0          24      40  1/10
cs1           0/0               0/0              0/0          22      40  1/10
cs2           0/0               0/0              0/0          24      40  1/10
cs3           0/0               0/0              0/0          26      40  1/10
cs4           0/0               0/0              0/0          28      40  1/10
cs5           0/0               0/0              0/0          30      40  1/10
cs6           0/0               0/0              0/0          32      40  1/10
cs7           0/0               0/0              0/0          34      40  1/10
ef            0/0               0/0              0/0          36      40  1/10
rsvp          0/0               0/0              0/0          36      40  1/10
default       0/0               0/0              0/0          20      40  1/10
```

Policing and Shaping

Policing and Shaping Overview

Cisco IOS QoS offers two kinds of traffic regulation mechanisms—policing and shaping.

The rate-limiting features of committed access rate (CAR) and the Traffic Policing feature provide the functionality for policing traffic. The features of Generic Traffic Shaping (GTS), Class-Based Traffic Shaping, Distributed Traffic Shaping (DTS), and Frame Relay Traffic Shaping (FRTS) provide the functionality for shaping traffic.

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

You can deploy these features throughout your network to ensure that a packet, or data source, adheres to a stipulated contract and to determine the QoS to render the packet. Both policing and shaping mechanisms use the traffic descriptor for a packet—indicated by the classification of the packet—to ensure adherence and service. (See the "Classification Overview" module for a description of a traffic descriptor.)

Policers and shapers usually identify traffic descriptor violations in an identical manner. They usually differ, however, in the way they respond to violations, for example:

- A policer typically drops traffic. (For example, the CAR rate-limiting policer will either drop the packet or rewrite its IP precedence, resetting the type of service bits in the packet header.)
- A shaper typically delays excess traffic using a buffer, or queueing mechanism, to hold packets and shape the flow when the data rate of the source is higher than expected. (For example, GTS and Class-Based Shaping use a weighted fair queue to delay packets in order to shape the flow, and DTS and FRTS use either a priority queue, a custom queue, or a FIFO queue for the same, depending on how you configure it.)

Traffic shaping and policing can work in tandem. For example, a good traffic shaping scheme should make it easy for nodes inside the network to detect misbehaving flows. This activity is sometimes called policing the traffic of the flow.

This chapter gives a brief description of the Cisco IOS QoS traffic policing and shaping mechanisms. Because policing and shaping all use the token bucket mechanism, this chapter first explains how a token bucket works. This chapter includes the following sections:

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

What Is a Token Bucket?

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
```

Here are some definitions of these terms:

- Mean rate—Also called the committed information rate (CIR), it specifies how much data can be sent or forwarded per unit time on average.
- Burst size—Also called the Committed Burst (Bc) size, it specifies in bits (or bytes) per burst how much traffic can be sent within a given unit of time to not create scheduling concerns. (For a shaper, such as GTS, it specifies bits per burst; for a policer, such as CAR, it specifies bytes per burst.)
- Time interval—Also called the measurement interval, it specifies the time quantum in seconds per burst.

By definition, over any integral multiple of the interval, the bit rate of the interface will not exceed the mean rate. The bit rate, however, may be arbitrarily fast within the interval.

A token bucket is used to manage a device that regulates the data in a flow. For example, the regulator might be a traffic policer, such as CAR, or a traffic shaper, such as FRTS or GTS. A token bucket itself has no discard or priority policy. Rather, a token bucket discards tokens and leaves to the flow the problem of managing its transmission queue if the flow overdrives the regulator. (Neither CAR nor FRTS and GTS implement either a true token bucket or true leaky bucket.)

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

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

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

For traffic shaping, a token bucket permits burstiness but bounds it. It guarantees that the burstiness is bounded so that the flow will never send faster than the token bucket's capacity, divided by the time interval, plus the established rate at which tokens are placed in the token bucket. See the following formula:

```
(token bucket capacity in bits / time interval in seconds) + established rate in bps =
maximum flow speed in bps
```

This method of bounding burstiness also guarantees that the long-term transmission rate will not exceed the established rate at which tokens are placed in the bucket.

Policing with CAR

CAR embodies a rate-limiting feature for policing traffic, in addition to its packet classification feature discussed in the "Classification Overview" module. The rate-limiting feature of CAR manages the access bandwidth policy for a network by ensuring that traffic falling within specified rate parameters is sent, while dropping packets that exceed the acceptable amount of traffic or sending them with a different priority. The exceed action for CAR is to drop or mark down packets.

The rate-limiting function of CAR does the following:

- Allows you to control the maximum rate of traffic sent or received on an interface.
- Gives you the ability to define Layer 3 aggregate or granular incoming or outgoing (ingress or egress) bandwidth rate limits and to specify traffic handling policies when the traffic either conforms to or exceeds the specified rate limits.

  Aggregate bandwidth rate limits match all of the packets on an interface or subinterface. Granular bandwidth rate limits match a particular type of traffic based on precedence, MAC address, or other parameters.

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

How It Works

CAR examines traffic received on an interface or a subset of that traffic selected by access list criteria. It then compares the rate of the traffic to a configured token bucket and takes action based on the result. For example, CAR will drop the packet or rewrite the IP precedence by resetting the type of service (ToS) bits. You can configure CAR to send, drop, or set precedence.

Aspects of CAR rate limiting are explained in the following sections:

- Matching Criteria
- Rate Limits
- Conform and Exceed Actions
- Multiple Rate Policies

CAR utilizes a token bucket measurement. Tokens are inserted into the bucket at the committed rate. The depth of the bucket is the burst size. Traffic arriving at the bucket when sufficient tokens are available is said to conform, and the corresponding number of tokens are removed from the bucket. If a sufficient number of tokens are not available, then the traffic is said to exceed.

Matching Criteria

Traffic matching entails identification of traffic of interest for rate limiting, precedence setting, or both. Rate policies can be associated with one of the following qualities:

- Incoming interface
- All IP traffic
- IP precedence (defined by a rate-limit access list)
- MAC address (defined by a rate-limit access list)

- Multiprotocol Label Switching (MPLS) experimental (EXP) value (defined by a rate-limit access list)
- IP access list (standard and extended)

CAR provides configurable actions, such as send, drop, or set precedence when traffic conforms to or exceeds the rate limit.

Note Matching to IP access lists is more processor-intensive than matching based on other criteria.

Rate Limits

CAR propagates bursts. It does no smoothing or shaping of traffic, and therefore does no buffering and adds no delay. CAR is highly optimized to run on high-speed links—DS3, for example—in distributed mode on Versatile Interface Processors (VIPs) on the Cisco 7500 series.

CAR rate limits may be implemented either on input or output interfaces or subinterfaces including Frame Relay and ATM subinterfaces.

What Rate Limits Define

Rate limits define which packets conform to or exceed the defined rate based on the following three parameters:

- Average rate. The average rate determines the long-term average transmission rate. Traffic that falls under this rate will always conform.
- Normal burst size. The normal burst size determines how large traffic bursts can be before some traffic exceeds the rate limit.
- Excess Burst size. The Excess Burst (Be) size determines how large traffic bursts can be before all traffic exceeds the rate limit. Traffic that falls between the normal burst size and the Excess Burst size exceeds the rate limit with a probability that increases as the burst size increases.

The maximum number of tokens that a bucket can contain is determined by the normal burst size configured for the token bucket.

When the CAR rate limit is applied to apacket, CAR removes from the bucket tokens that are equivalent in number to the byte size of the packet. If a packet arrives and the byte size of the packet is greater than the number of tokens available in the standard token bucket, extended burst capability is engaged if it is configured.

Extended Burst Value

Extended burst is configured by setting the extended burst value greater than the normal burst value. Setting the extended burst value equal to the normal burst value excludes the extended burst capability. If extended burst is not configured, given the example scenario, the exceed action of CAR takes effect because a sufficient number of tokens are not available.

When extended burst is configured and this scenario occurs, the flow is allowed to borrow the needed tokens to allow the packet to be sent. This capability exists so as to avoid tail-drop behavior, and, instead, engage behavior like that of Random Early Detection (RED).

How Extended Burst Capability Works

Here is how the extended burst capability works. If a packet arrives and needs to borrow *n* number of tokens because the token bucket contains fewer tokens than its packet size requires, then CAR compares the following two values:

- Extended burst parameter value.
- Compounded debt. Compounded debt is computed as the sum over all *ai*:
  - *a* indicates the actual debt value of the flow after packet *i* is sent. Actual debt is simply a count of how many tokens the flow has currently borrowed.
  - *i* indicates the *i*th packet that attempts to borrow tokens since the last time a packet was dropped.

If the compounded debt is greater than the extended burst value, the exceed action of CAR takes effect. After a packet is dropped, the compounded debt is effectively set to 0. CAR will compute a new compounded debt value equal to the actual debt for the next packet that needs to borrow tokens.

If the actual debt is greater than the extended limit, all packets will be dropped until the actual debt is reduced through accumulation of tokens in the token bucket.

Dropped packets do not count against any rate or burst limit. That is, when a packet is dropped, no tokens are removed from the token bucket.

Note Though it is true the entire compounded debt is forgiven when a packet is dropped, the actual debt is not forgiven, and the next packet to arrive to insufficient tokens is immediately assigned a new compounded debt value equal to the current actual debt. In this way, actual debt can continue to grow until it is so large that no compounding is needed to cause a packet to be dropped. In effect, at this time, the compounded debt is not really forgiven. This scenario would lead to excessive drops on streams that continually exceed normal burst. (See the example in the following section, "Actual and Compounded Debt Example."

Testing of TCP traffic suggests that the chosen normal and extended burst values should be on the order of several seconds worth of traffic at the configured average rate. That is, if the average rate is 10 Mbps, then a normal burst size of 10 to 20 Mbps and an Excess Burst size of 20 to 40 Mbps would be appropriate.

Recommended Burst Values

Cisco recommends the following values for the normal and extended burst parameters:

```
normal burst = configured rate * (1 byte)/(8 bits) * 1.5 seconds
extended burst = 2 * normal burst
```

With the listed choices for parameters, extensive test results have shown CAR to achieve the configured rate. If the burst values are too low, then the achieved rate is often much lower than the configured rate.

Actual and Compounded Debt Example

This example shows how the compounded debt is forgiven, but the actual debt accumulates.

For this example, assume the following parameters:

- Token rate is 1 data unit per time unit
- Normal burst size is 2 data units

- Extended burst size is 4 data units
- 2 data units arrive per time unit

After 2 time units, the stream has used up its normal burst and must begin borrowing one data unit per time unit, beginning at time unit 3:

```
Time    DU arrivals     Actual Debt     Compounded Debt
-------------------------------------------------------
1       2               0               0
2       2               0               0
3       2               1               1
4       2               2               3
5       2               3 (temporary)   6 (temporary)
```

At this time a packet is dropped because the new compounded debt (6) would exceed the extended burst limit (4). When the packet is dropped, the compounded debt effectively becomes 0, and the actual debt is 2. (The values 3 and 6 were only temporary and do not remain valid in the case where a packet is dropped.) The final values for time unit 5 follow. The stream begins borrowing again at time unit 6.

```
Time    DU arrivals     Actual Debt     Compounded Debt
-------------------------------------------------------
5       2               2               0
6       2               3               3
7       2               4 (temporary)   7 (temporary)
```

At time unit 6, another packet is dropped and the debt values are adjusted accordingly.

```
Time    DU arrivals     Actual Debt     Compounded Debt
-------------------------------------------------------
7       2               3               0
```

Conform and Exceed Actions

CAR utilizes a token bucket, thus CAR can pass temporary bursts that exceed the rate limit as long as tokens are available.

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

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

For VIP-based platforms, two more actions are possible:

- Set QoS group and transmit—The packet is assigned to a QoS group and sent.
- Set QoS group and continue—The packet is assigned to a QoS group and then evaluated using the next rate policy. If there is not another rate policy, the packet is sent.

Multiple Rate Policies

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

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

Cascading of rate policies allows a series of rate limits to be applied to packets to specify more granular policies (for example, you could rate limit total traffic on an access link to a specified subrate bandwidth and then rate limit World Wide Web traffic on the same link to a given proportion of the subrate limit) or to match packets against an ordered sequence of policies until an applicable rate limit is encountered (for example, rate limiting several MAC addresses with different bandwidth allocations at an exchange point). You can configure up to a 100 rate policies on a subinterface.

Restrictions

CAR and VIP-distributed CAR can only be used with IP traffic. Non-IP traffic is not rate limited.

CAR or VIP-distributed CAR can be configured on an interface or subinterface. However, CAR and VIP-distributed CAR are not supported on the following interfaces:

- Fast EtherChannel
- Tunnel
- PRI
- Any interface that does not support Cisco Express Forwarding (CEF)

CAR is only supported on ATM subinterfaces with the following encapsulations: aal5snap, aal5mux, and aal5nlpid.

Note CAR provides rate limiting and does not guarantee bandwidth. CAR should be used with other QoS features, such as distributed weighted fair queueing (WFQ) (DWFQ), if premium bandwidth assurances are required.

Traffic Policing

Traffic policing allows you to control the maximum rate of traffic sent or received on an interface, and to partition a network into multiple priority levels or class of service (CoS).

The Traffic Policing feature manages the maximum rate of traffic through a token bucket algorithm. The token bucket algorithm can use the user-configured values to determine the maximum rate of traffic allowed on an interface at a given moment in time. The token bucket algorithm is affected by all traffic entering or leaving (depending on where the traffic policy with Traffic Policing configured) and is useful in managing network bandwidth in cases where several large packets are sent in the same traffic stream.

The token bucket algorithm provides users with three actions for each packet: a conform action, an exceed action, and an optional violate action. Traffic entering the interface with Traffic Policing configured is placed in to one of these categories. Within these three categories, users can decide packet

treatments. For instance, 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.

Traffic Policing is 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. Users can change these configuration options to suit their network needs.

The Traffic Policing feature supports the following MIBs:

- CISCO-CLASS-BASED-QOS-MIB
- CISCO-CLASS-BASED-QOS-CAPABILITY-MIB

This feature also supports RFC 2697, *A Single Rate Three Color Marker.*

For information on how to configure the Traffic Policing feature, see the "Configuring Traffic Policing" module.

Benefits

Bandwidth Management Through Rate Limiting

Traffic policing allows you 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 that falls within the rate parameters is sent, whereas traffic that exceeds the parameters is dropped or sent with a different priority.

Packet Marking Through IP Precedence, QoS Group, and DSCP Value Setting

Packet marking allows you to partition your network into multiple priority levels or classes of service (CoS), as follows:

- Use traffic policing to set the IP precedence or differentiated services code point (DSCP) values for packets entering the network. Networking devices within your network can then use the adjusted IP Precedence values to determine how the traffic should be treated. For example, the DWRED feature uses the IP Precedence values to determine the probability that a packet will be dropped.
- Use traffic policing to assign packets to a QoS group. The router uses the QoS group to determine how to prioritize packets.

Restrictions

The following restrictions apply to the Traffic Policing feature:

- On a Cisco 7500 series router, traffic policing can monitor CEF switching paths only. In order to use the Traffic Policing feature, CEF must be configured on both the interface receiving the packet and the interface sending the packet.
- On a Cisco 7500 series router, traffic policing cannot be applied to packets that originated from or are destined to a router.
- Traffic policing can be configured on an interface or a subinterface.

- Traffic policing is not supported on the following interfaces:
  - Fast EtherChannel
  - Tunnel
  - PRI
  - Any interface on a Cisco 7500 series router that does not support CEF

Prerequisites

On a Cisco 7500 series router, CEF must be configured on the interface before traffic policing can be used.

For additional information on CEF, see the "Cisco Express Forwarding Features Roadmap" module.

Traffic Shaping (Regulating Packet Flow)

Regulating the packet flow (that is, the flow of traffic) on the network is also known as traffic shaping. Traffic shaping allows you to control the speed of traffic leaving an interface. This way, you can match the lfow of the traffic to the speed of the interface receiving the packet.

Cisco provides three mechanisms for regulating or shaping traffic: Class-Based Traffic Shaping, Generic Traffic Shaping (GTS), and Frame Relay Traffic Shaping (FRTS).

For more information about traffic shaping, see the "Regulating Packet Flow Using Traffic Shaping" module.

For information on configuring Frame Relay and FRTS, see the "Configuring Frame Relay" module and the "MQC-Based Frame Relay Traffic Shaping" module, respectively.

MQC—Traffic Shaping Overhead Accounting for ATM

First Published: December 4, 2006
Last Updated: April 18, 2008

The MQC—Traffic Shaping Overhead Accounting for ATM feature enables a broadband aggregation system (BRAS) to account for various encapsulation types when applying quality of service (QoS) functionality to packets. Typically, in Ethernet digital subscriber line (DSL) environments, the encapsulation from the router to the digital subscriber line access multiplexer (DSLAM) is Gigabit Ethernet and the encapsulation from the DSLAM to the customer premises equipment (CPE) is ATM. ATM overhead accounting enables the router to account for ATM encapsulation on the subscriber line and for the overhead added by cell segmentation. This functionality enables the service provider to prevent overruns at the subscriber line and ensures that the router executes QoS features on the actual bandwidth used by ATM packets.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for MQC—Traffic Shaping Overhead Accounting for ATM" section on page 17.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Traffic Shaping Overhead Accounting for ATM

Traffic classes must be configured using the **class-map** command.

Restrictions for Traffic Shaping Overhead Accounting for ATM

- The router supports ATM overhead accounting only for the **shape** and **bandwidth** commands.
- If you enable ATM overhead accounting on a child policy, then you must enable ATM overhead accounting on the parent policy.
- In a policy map, you must either enable ATM overhead accounting for all classes in the policy or disable overhead accounting for all classes in the policy. You cannot enable overhead accounting for some classes and disable overhead accounting for other classes in the same policy.
- The encapsulation type used within a policy map and between the parent policy map and the child policy map (in a hierarchical policy map structure) must be consistent.
- When you enter the **show policy-map session** command, the resulting classification byte counts and the queuing feature byte counts do not match. This is because the classification byte count does not consider overhead, whereas the queuing features do consider overhead.

  Note This restriction applies to the Cisco 10000 series router only. This restriction does not apply to the Cisco 7600 series router.

- You must attach a policy map that is configured with ATM overhead accounting to only an Ethernet interface (or an IP session on an Ethernet interface).

Information About Traffic Shaping Overhead Accounting for ATM

Before configuring traffic shaping overhead accounting for ATM, you should understand the following concepts:

Benefits of Traffic Shaping Overhead Accounting for ATM

The Traffic Shaping Overhead Accounting for ATM feature enables the broadband aggregation system (BRAS) to account for various encapsulation types when applying QoS to packets. Typically, in Ethernet digital subscriber line (DSL) environments, the encapsulation from the BRAS to the DSLAM is Gigabit Ethernet and the encapsulation from the DSLAM to the CPE is ATM. ATM overhead accounting enables the BRAS to account for ATM encapsulation on the subscriber line and for the overhead added by cell segmentation. This functionality enables the service provider to prevent overruns at the subscriber line and ensures that the router executes QoS features on the actual bandwidth used by ATM subscriber traffic.

BRAS and Encapsulation Types

Broadband aggregation system (BRAS) uses the encapsulation type that is configured for the DSLAM-CPE side to calculate the ATM overhead per packet.

DSLAM-CPE encapsulation types are based on Subnetwork Access Protocol (SNAP) and multiplexer (MUX) formats of ATM adaptation layer 5 (AAL5), followed by routed bridge (RBE), x-1483, x-dot1q-rbe, IP, PPP over Ethernet (PPPoE), or PPP over ATM (PPPoA) encapsulations. Because the DSLAM treats IP and PPPoE packets as payload, the BRAS does not account for IP and PPPoE encapsulations.

On the BRAS-DSLAM side, encapsulation is IEEE 802.1Q VLAN or Q-in-Q (qinq). However, because the DSLAM removes the BRAS-DSLAM encapsulation, the BRAS does not account for 802.1Q or qinq encapsulation.

AAL5 segmentation processing adds the additional overhead of the 5-byte cell headers, the AAL5 Common Part Convergence Sublayer (CPCS) padding, and the AAL5 trailer. For more information, see the "ATM Overhead Calculation" section on page 4.

Subscriber Line Encapsulation Types

The router supports the following subscriber line encapsulation types:

- snap-rbe
- mux-rbe
- snap-dot1q-rbe
- mux-dot1q-rbe
- snap-pppoa
- mux-pppoa
- snap-1483routed
- mux-1483routed

Note The encapsulation types listed above are for AAL5, qinq, and dot1q encapsulations. User-defined encapsulations with offsets based on the platform in use are also supported. (For the Cisco 10000 series router, valid offsets are –63 to +63. For the Cisco 7600 series router, valid offsets are –48 to +48.)

ATM Overhead Calculation

The Traffic Shaping Overhead Accounting for ATM feature prevents oversubscription of a subscriber line by accounting for the ATM encapsulation overhead at the BRAS. When calculating the ATM overhead, the Traffic Shaping Overhead Accounting for ATM feature considers the following:

- The encapsulation type used by the BRAS
- The CPCS trailer overhead
- The encapsulation type used between the DSLAM and the CPE

The offset size (a parameter used to calculate ATM overhead accounting) is calculated using the following formula:

Offset size in bytes = (CPCS trailer overhead) + (DSLAM to CPE) - (BRAS encapsulation type)

See Table 1 for the offset sizes, in bytes, derived from this formula.

This offset size, along with the packet size and packet assembler/disassembler (PAD) byte overhead in the CPCS, is used by the router to calculate the ATM overhead accounting rate.

Note A CPCS trailer overhead of 8 bytes corresponds to AAL5. A CPCS trailer overhead of 4 bytes corresponds to AAL3, but AAL3 is not supported.

Table 1 Offset Sizes, in Bytes, Used for ATM Overhead Calculation

Encapsulation Type in Use	BRAS	CPCS Trailer Overhead	DSLAM to CPE	Offset Size
dot1q mux-1483routed	18	8	3	–7
dot1q snap-1483routed	18	8	6	–4
dot1q mux-rbe	18	8	14	4
dot1q snap-rbe	18	8	24	14
dot1q mux-dot1q-rbe	18	8	18	8
dot1q snap-dot1q-rbe	18	8	28	18
qot1q mux-pppoa	18 + 6	8	2	–14
qot1q snap-pppoa	18 + 6	8	4	–12
qinq mux-1483routed	22	8	3	–11
qinq snap-1483routed	22	8	6	–8
qinq mux-rbe	22	8	14	0
qinq snap-rbe	22	8	24	10
qinq mux-dot1q-rbe	22	8	18	4
qing snap-dot1q-rbe	22	8	28	14
qinq mux-pppoa	22 + 6	8	2	–18
qinq snap-pppoa	22 + 6	8	4	–16

ATM Overhead Accounting and Hierarchical Policies

In hierarchical policies, you can enable ATM overhead accounting for shaping and bandwidth on parent policies and child policies. You are not required to enable ATM overhead accounting on a traffic class that does not contain the **bandwidth** or **shape** command. If you enable ATM overhead accounting on a child policy, then you must enable ATM overhead accounting on the parent policy. The parent and child classes must specify the same encapsulation type when ATM overhead accounting is enabled.

How to Configure Traffic Shaping Overhead Accounting for ATM

This section contains the following tasks.

- Configuring Traffic Shaping Overhead Accounting for ATM in a Hierarchical Policy, page 5 (required)
- Verifying the Configuration of Traffic Shaping Overhead Accounting for ATM, page 10 (optional)

Configuring Traffic Shaping Overhead Accounting for ATM in a Hierarchical Policy

To configure traffic shaping overhead accounting for ATM in a hierarchical policy map structure, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** *class-map-name*
5. **bandwidth** {*bandwidth-kbps* | **percent** *percentage* | **remaining percent** *percentage*} [**account** {**qinq** | **dot1q**} [**aal5**] {*subscriber-encapsulation* | **user-defined** *offset*}]
6. **bandwidth remaining ratio** *ratio* [**account** {**qinq** | **dot1q**} [**aal5**] {*subscriber-encapsulation* | **user-defined** *offset*}]
7. **shape** [**average** | **peak**] *mean-rate* [*burst-size*] [*excess-burst-size*] [**account** {**qinq** | **dot1q**} [**aal5**] {*subscriber-encapsulation* | **user-defined** *offset*}]
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map-name* **Example:** `Router(config)# policy-map Business`	Creates or modifies the child policy and enters policy-map configuration mode. • Enter the policy map name. This is the name of the child policy and can be a maximum of 40 alphanumeric characters.
Step 4	**class** *class-map-name* **Example:** `Router(config-pmap)# class video`	Assigns the traffic class that you specify for the policy map and enters policy-map class configuration mode. • Enter the traffic class name. This is the name of the previously configured class map and can be a maximum of 40 alphanumeric characters.

	Command or Action	Purpose
Step 5	**bandwidth** {*bandwidth-kbps* | **percent** *percentage* | **remaining percent** *percentage*} [**account** {**qinq** | **dot1q**} [**aal5**] {*subscriber-encapsulation* | **user-defined** *offset*}] **Example:** `Router(config-pmap-c)# bandwidth 8000 account dot1q aal5 snap-pppoa`	Enables Class-Based Weighted Fair Queuing (CBWFQ) on the basis of the keywords and arguments specified, such as the following: • *bandwidth-kbps*—Specifies or modifies the minimum bandwidth allocated for a class that belongs to a policy map. Valid values are from 8 to 2488320, which represents from 1 to 99 percent of the link bandwidth. • **percent** *percentage*—Specifies or modifies the minimum percentage of the link bandwidth allocated for a class that belongs to a policy map. Valid values are from 1 to 99. • **remaining percent** *percentage*—Specifies or modifies the minimum percentage of unused link bandwidth allocated for a class that belongs to a policy map. Valid values are from 1 to 99. • **account**—Enables ATM overhead accounting. • **qinq**—Specifies queue-in-queue encapsulation as the BRAS-DSLAM encapsulation type. • **dot1q**—Specifies IEEE 802.1Q VLAN encapsulation as the BRAS-DSLAM encapsulation type. • **aal5**—Specifies the ATM adaptation layer 5 that supports connection-oriented variable bit rate (VBR) services. • *subscriber-encapsulation*—Specifies the encapsulation type at the subscriber line. For more information, see the "Subscriber Line Encapsulation Types" section on page 3. • **user-defined**—Specifies the offset size that the router uses when calculating the ATM overhead. • *offset*—Specifies the offset size when calculating ATM overhead. Valid values are from –63 to +63 bytes. **Note** For the Cisco 7600 series router, valid values are from –48 to +48 bytes.

	Command or Action	Purpose
Step 6	**bandwidth remaining ratio** *ratio* [**account** {**qinq** | **dot1q**} [**aal5**] {*subscriber-encapsulation* | **user-defined** *offset*}] **Example:** `Router(config-pmap-c)# bandwidth remaining ratio 10 account dot1q aal5 snap-pppo`	(Optional) Specifies the bandwidth-remaining ratio for the subinterface along with ATM accounting parameters: • *ratio*—Specifies the bandwidth-remaining ratio for the subinterface. Valid values are 1 to 100. The default value is 1. **Note** For the Cisco 7600 series router, valid values are from 1 to 10000. The default value is 1. • **account**—Enables ATM overhead accounting. • **qinq**—Specifies queue-in-queue encapsulation as the BRAS-DSLAM encapsulation type. • **dot1q**—Specifies IEEE 802.1Q VLAN encapsulation as the BRAS-DSLAM encapsulation type. • **aal5**—Specifies the ATM adaptation layer 5 that supports connection-oriented VBR services. • *subscriber-encapsulation*—Specifies the encapsulation type at the subscriber line. For more information, see the "Subscriber Line Encapsulation Types" section on page 3. • **user-defined**—Specifies the offset size that the router uses when calculating the ATM overhead. • *offset*—Specifies the offset size, in bytes, when calculating ATM overhead. Valid values are from –63 to +63. **Note** For the Cisco 7600 series router, valid values are from –48 to +48.

	Command or Action	Purpose
Step 7	**shape** [**average** | **peak**] *mean-rate* [*burst-size*] [*excess-burst-size*][**account** {**qinq** | **dot1q**} [**aal5**] {*subscriber-encapsulation* | **user-defined** *offset*}] **Example:** `Router(config-pmap-c)# shape 8000 account qinq aal5 snap-dot1q-rbe`	Shapes traffic to the indicated bit rate and enables ATM overhead accounting on the basis of the keywords and arguments specified, such as the following: • **average**—(Optional) The committed burst (Bc) that specifies the maximum number of bits sent out in each interval. • **peak**—(Optional) Specifies the maximum number of bits sent out in each interval (the Bc + excess burst [Be]). The Cisco 10000 router and the SIP400 (on the Cisco 7600 series router) do not support this option. • *mean-rate*—Also called committed information rate (CIR). Indicates the bit rate used to shape the traffic, in bits per second. • *burst-size*—(Optional) The number of bits in a measurement interval (Bc). • *excess-burst-size*—(Optional) The acceptable number of bits permitted to go over the Be. • **account**—Enables ATM overhead accounting. • **qinq**—Specifies queue-in-queue encapsulation as the BRAS-DSLAM encapsulation type. • **dot1q**—Specifies IEEE 802.1Q VLAN encapsulation as the BRAS-DSLAM encapsulation type. • **aal5**—The ATM adaptation layer 5 that supports connection-oriented variable bit rate (VBR) services. • *subscriber-encapsulation*—Specifies the encapsulation type at the subscriber line. For more information, see the "Subscriber Line Encapsulation Types" section on page 3. • **user-defined**—Specifies the offset size that the router uses when calculating the ATM overhead. • *offset*—Specifies the offset size when calculating ATM overhead. Valid values are from –63 to +63 bytes. **Note** For the Cisco 7600 series router, valid values are from –48 to +48 bytes.
Step 8	**end** **Example:** `Router(config-pmap-c)# end`	Exits policy-map class configuration mode.

Verifying the Configuration of Traffic Shaping Overhead Accounting for ATM

To verify the configuration of traffic shaping overhead accounting for ATM, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show policy-map** [*policy-map-name*]
3. **show policy-map session**
4. **show running-config**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show policy-map** [*policy-map-name*] **Example:** `Router# show policy-map unit-test`	(Optional) Displays the configuration of all classes for a specified policy map or of all classes for all existing policy maps. • (Optional) Enter the policy map name. The name can be a maximum of 40 alphanumeric characters.
Step 3	**show policy-map session** **Example:** `Router# show policy-map session`	(Optional) Displays the QoS policy map in effect for a IPoE/PPPoE session.
Step 4	**show running-config** **Example:** `Router# show running-config`	(Optional) Displays the contents of the currently running configuration file.
Step 5	**exit** **Example:** `Router# exit`	Exits privileged EXEC mode.

Configuration Examples for Traffic Shaping Overhead Accounting for ATM

This section provides the following configuration examples:

Enabling Traffic Shaping Overhead Accounting for ATM: Example

The following example shows how to enable ATM overhead accounting using a hierarchical policy map structure. The Child policy map has two classes: Business and Non-Business. The Business class has priority and is policed at 128,000 kbps. The Non-Business class has ATM overhead accounting enabled and has a bandwidth of 20 percent of the available bandwidth. The Parent policy map shapes the aggregate traffic to 256,000 kbps and enables ATM overhead accounting.

Notice that Layer 2 overhead accounting is not explicitly configured for the Business traffic class. If the class-default class of a parent policy has ATM overhead accounting enabled, you are not required to enable ATM overhead accounting on a child traffic class that does not contain the **bandwidth** or **shape** command. Therefore, in this example, the Business priority queue implicitly has ATM overhead accounting enabled because its parent class-default class has overhead accounting enabled.

```
policy-map Child
 class Business
  priority
  police 128000
 class Non-Business
  bandwidth percent 20 account dot1q aal5 snap-rbe-dot1q
  exit
 exit
policy-map Parent
 class class-default
  shape 256000 account dot1q aal5 snap-rbe-dot1q
  service-policy Child
```

In the following example, overhead accounting is enabled for bandwidth on the gaming and class-default class of the child policy map named subscriber_classes and on the class-default class of the parent policy map named subscriber_line. The voip and video classes do not have accounting explicitly enabled; these classes have ATM overhead accounting implicitly enabled because the parent policy has overhead accounting enabled. Notice that the features in the parent and child policies use the same encapsulation type.

```
policy-map subscriber_classes
 class voip
  priority level 1
  police 8000
 class video
  priority level 2
  police 8000
 class gaming
  bandwidth remaining percent 80 account dot1q aal5 snap-rbe-dot1q
 class class-default
  bandwidth remaining percent 20 account dot1q aal5 snap-rbe-dot1q
policy-map subscriber_line
```

```
 class class-default
  bandwidth remaining ratio 10 account dot1q aal5 snap-rbe-dot1q
  shape average 512 account aal5 dot1q snap-rbe-dot1q
  service policy subscriber_classes
```

Verifying Traffic Shaping Overhead Accounting for ATM: Example

The following output from the **show policy-map interface** command indicates that ATM overhead accounting is enabled for shaping and disabled for bandwidth:

```
Router# show policy-map interface


Service-policy output:unit-test

    Class-map: class-default (match-any)
    100 packets, 1000 bytes
    30 second offered rate 800 bps, drop rate 0 bps
    Match: any
    shape (average) cir 154400, bc 7720, be 7720
    target shape rate 154400
    overhead accounting: enabled
    bandwidth 30% (463 kbps)
    overhead accounting: disabled

    queue limit 64 packets
    (queue depth/total drops/no-buffer drops) 0/0/0
    (packets output/bytes output) 100/1000
```

The following output from the **show policy-map session** command indicates that ATM overhead accounting is enabled for shaping.

```
Router# show policy-map session output

SSS session identifier 2 -

Service-policy output:  ATM_OH_POLICY


    Class-map: class-default (match-any)
      0 packets, 0 bytes
      30 second offered rate 0 bps, drop rate 0 bps
      Match: any
      Queueing
      queue limit 2500 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 0/0
      shape (average) cir 10000000, bc 40000, be 40000
      target shape rate 10000000
       Overhead Accounting Enabled
```

The following output from the **show running-config** command indicates that ATM overhead accounting is enabled for shaping. The BRAS-DSLAM encapsulation is dot1q and the subscriber line encapsulation is snap-rbe based on the AAL5 service.

```
subscriber policy recording rules limit 64
no mpls traffic-eng auto-bw timers frequency 0
call rsvp-sync
!
controller T1 2/0
    framing sf
    linecode ami
```

```
!
controller T1 2/1
    framing sf
    linecode ami
!
!
policy-map unit-test
    class class-default
        shape average percent 10 account dot1q aal5 snap-rbe
!
```

Additional References

The following sections provide references related to traffic shaping overhead accounting for ATM.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC), hierarchical policies, policy maps	"Applying QoS Features Using the MQC" module
Policing and shaping traffic	"Policing and Shaping Overview" module

Standards

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

MIBs

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

RFCs

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

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Quality of Service Solutions Command Reference* at http://www.cisco.com/en/US/docs/ios/qos/command/reference/qos_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

- **bandwidth (policy-map class)**
- **bandwidth remaining ratio**
- **shape (policy-map class)**
- **show policy-map interface**
- **show policy-map session**
- **show running-config**

Feature Information for MQC—Traffic Shaping Overhead Accounting for ATM

Table 2 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 2 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 2 Feature Information for MQC—Traffic Shaping Overhead Accounting for ATM

Feature Name	Releases	Feature Information
MQC—Traffic Shaping Overhead Accounting for ATM	12.2(31)SB2 12.2(33)SRC 12.2(33)SB	The MQC—Traffic Shaping Overhead Accounting for ATM feature enables a broadband aggregation system (BRAS) to account for various encapsulation types when applying QoS functionality to packets. In Release 12.2(31)SB2, this feature was introduced and implemented on the Cisco 10000 series router for the PRE3. In Release 12.2(33)SRC, support was added for the Cisco 7600 series router. In Release 12.2(33)SB, support was added for the Cisco 7300 series router. The following commands were introduced or modified: **bandwidth (policy-map class)**, **bandwidth remaining ratio**, **shape (policy-map class)**, **show policy-map interface**, **show policy-map session**, **show running-config**.

Configuring Traffic Policing

Traffic Policing

This feature module describes the Traffic Policing feature. It includes information on the benefits of the feature, supported platforms, related documents, and so forth.

For complete conceptual information, see the "Policing and Shaping Overview" module.

For a complete description of the Traffic Policing commands mentioned in this module, refer to the *Cisco IOS Quality of Service Solutions Command Reference*. To locate documentation of other commands that appear in this module, use the command reference master index or search online.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

Contents

This document includes the following sections:

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Feature Overview

The Traffic Policing feature performs the following functions:

- Limits the input or output transmission rate of a class of traffic based on user-defined criteria
- Marks packets by setting the ATM Cell Loss Priority (CLP) bit, Frame Relay Discard Eligibility (DE) bit, IP precedence value, IP differentiated services code point (DSCP) value, MPLS experimental value, and Quality of Service (QoS) group.

Traffic policing allows you to control the maximum rate of traffic transmitted or received on an interface. The Traffic Policing feature is applied when you attach a traffic policy contain the Traffic Policing configuration to an interface. A traffic policy is configured using the Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC). For information about the MQC, see the "Applying QoS Features Using the MQC" module.

Table 1 lists the feature history.

Table 1 Feature History

Cisco IOS Release	Enhancement
12.1(5)T	This command was introduced for Cisco IOS Release 12.1 T. A new Traffic Policing algorithm was introduced. The **violate-action** option became available. This feature became available on Cisco 2600, 3600, 4500, 7200, and 7500 series routers.
12.2(2)T	The **set-clp-transmit** option for the *action* argument was added to the **police** command. The **set-frde-transmit** option for the *action* argument was added to the **police** command. However, the **set-frde-transmit** option is not supported for Any Transport over Multiprotocol Label Switching (MPLS) (AToM) traffic in this release. The **set-mpls-exp-transmit** option for the *action* argument was added to the **police** command.
Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.

Benefits

Bandwidth Management Through Rate Limiting

Traffic policing allows you to control the maximum rate of traffic transmitted 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. In most Traffic Policing configurations, traffic that falls within the rate parameters is transmitted, whereas traffic that exceeds the parameters is dropped or transmitted with a different priority.

Packet Marking

Packet marking allows you to partition your network into multiple priority levels or classes of service (CoS). A packet is marked and these markings can be used to identify and classify traffic for downstream devices. In some cases, such as ATM Cell Loss Priority (CLP) marking or Frame Relay Discard Eligibility (DE) marking, the marking is used to classify traffic.

- Use traffic policing to set the IP precedence or DSCP values for packets entering the network. Networking devices within your network can then use the adjusted IP precedence values to determine how the traffic should be treated. For example, the Weighted Random Early Detection (WRED) feature uses the IP precedence values to determine the probability that a packet will be dropped.
- Use traffic policing to assign packets to a QoS group. The router uses the QoS group to determine how to prioritize packets within the router.

Traffic can be marked without using the Traffic Policing feature. If you want to mark traffic but do not want to use Traffic Policing, see the "Marking Network Traffic" module.

Packet Prioritization for Frame Relay Frames

The Traffic Policing feature allows users to mark the Frame Relay DE bit of the Frame Relay frame. The Frame Relay DE bit is one bit and, therefore, can be set to either 0 or 1. In congested environments, frames with the DE bit set to 1 are discarded before frames with the DE bit set to 0.

Packet Prioritization for ATM Cells

The Traffic Policing feature allows users to mark the ATM CLP bit in ATM cells. The ATM CLP bit is used to prioritize packets in ATM networks. The ATM CLP bit is one bit and, therefore, can be set to either 0 or 1. In congested environments, cells with the ATM CLP bit set to 1 are discarded before cells with the ATM CLP bit set to 0.

Restrictions

- On a Cisco 7500 series router, traffic policing can monitor Cisco Express Forwarding (CEF) switching paths only. In order to use the Traffic Policing feature, Cisco Express Forwarding must be configured on both the interface receiving the packet and the interface sending the packet.
- On a Cisco 7500 series router, traffic policing cannot be applied to packets that originated from or are destined to a router.
- Traffic policing can be configured on an interface or a subinterface.
- Traffic policing is not supported on the following interfaces:
  - Fast EtherChannel
  - Tunnel

Note Traffic policing is supported on tunnels that are using the Cisco generic routing encapsulation (GRE) tunneling protocol.

  - PRI
  - Any interface on a Cisco 7500 series router that does not support Cisco Express Forwarding

Related Features and Technologies

- Modular Quality of Service Command-Line Interface
- Class-Based Weighted Fair Queueing (CBWFQ)
- Class-Based Marking

Related Documents

- "Applying QoS Features Using the MQC" module
- "Configuring Committed Access Rate" module
- "Marking Network Traffic" module
- "Configuring Weighted Fair Queueing" module
- *Cisco IOS Quality of Service Solutions Command Reference*

Supported Platforms

- Cisco 2500 series

Note Cisco IOS Release 12.2(2)T or later does not run on Cisco 2500 series routers.

- Cisco 2600 series
- Cisco 3640 routers
- Cisco 4500 series
- Cisco 7000 series with RSP7000
- Cisco 7100 series
- Cisco 7200 series
- Cisco 7500 series

Note To use the **set-clp-transmit** action available with this feature, the Enhanced ATM Port Adapter (PA-A3) is required. Therefore, the **set-clp-transmit** action is not supported on any platform that does not support the PA-A3 adapter (such as the Cisco 2600 series router, the Cisco 3640 router, and the 4500 series router). For more information, see the documentation for your specific router.

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

Class-Based Quality of Service MIB

- CISCO-CLASS-BASED-QOS-MIB
- CISCO-CLASS-BASED-QOS-CAPABILITY-MIB

For descriptions of supported MIBs and how to use MIBs, see the Cisco MIB web site on CCO at http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml.

RFCs

- RFC 2697, *A Single Rate Three Color Marker*

Prerequisites

On a Cisco 7500 series router, Cisco Express Forwarding (CEF) must be configured on the interface before traffic policing can be used.

For additional information on CEF, see the "Cisco Express Forwarding Features Roadmap" module.

Configuration Tasks

See the following sections for configuration tasks for the Traffic Policing feature. Each task in the list indicates if the task is optional or required.

- Configuring Traffic Policing (Required)

Configuring Traffic Policing

To successfully configure the Traffic Policing feature, a traffic class and a traffic policy must be created, and the traffic policy must be attached to a specified interface. These tasks are performed using the MQC. For information about the MQC, see the "Applying QoS Features Using the MQC" module.

The Traffic Policing feature is configured in the traffic policy. To configure the Traffic Policing feature, use the following command in policy map configuration mode:

Command	Purpose
`Router(config-pmap-c)#` **police** *bps burst-normal burst-max* **conform-action** *action* **exceed-action** *action* **violate-action** *action*	Specifies a maximum bandwidth usage by a traffic class.

For more information about the **police** command, see the *Cisco IOS Quality of Service Solutions Command Reference.*

The Traffic Policing feature works with a token bucket mechanism. There are currently two types of token bucket algorithms: a single token bucket algorithm and a two token bucket algorithm. A single token bucket system is used when the **violate-action** option is not specified, and a two token bucket system is used when the **violate-action** option is specified.

For more information about token bucket mechanisms, see the "Policing and Shaping Overview" module.

Troubleshooting Tips

- Check the interface type. Verify that your interface is not mentioned in the nonsupported interface description in the "Restrictions" section of this module.
- For input traffic policing on a Cisco 7500 series router, verify that CEF is configured on the interface where traffic policing is configured.
- For output traffic policing on a Cisco 7500 series router, ensure that the incoming traffic is CEF-switched. Traffic policing cannot be used on the switching path unless CEF switching is enabled.

Monitoring and Maintaining Traffic Policing

To monitor and maintain the Traffic Policing feature, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router#` **show policy-map**	Displays all configured policy maps.
`Router#` **show policy-map** *policy-map-name*	Displays the user-specified policy map.
`Router#` **show policy-map interface**	Displays statistics and configurations of all input and output policies that are attached to an interface.

For more information about the **show policy-map** and **show policy-map interface** commands and how to interpret the information displayed, see the *Cisco IOS Quality of Service Solutions Command Reference*.

Configuration Examples

This section provides the following configuration example:

- Configuring a Service Policy that Includes Traffic Policing: Example, page 6

Configuring a Service Policy that Includes Traffic Policing: Example

The following configuration shows how to define a traffic class (with the **class-map** command) and associate that traffic class with a traffic policy (with the **policy-map** command). Traffic policing is applied in the traffic policy. The **service-policy** command is then used to attach the traffic policy to the interface.

For additional information on configuring traffic classes and traffic policies, see the "Applying QoS Features Using the MQC" module.

In this particular example, traffic policing is configured with the average rate at 8000 bits per second, the normal burst size at 2000 bytes, and the excess burst size at 4000 bytes. Packets coming into Fast Ethernet interface 0/0 are evaluated by the token bucket algorithm to analyze whether packets conform exceed, or violate the specified parameters. Packets that conform are transmitted, packets that exceed are assigned a QoS group value of 4 and are transmitted, and packets that violate are dropped.

For more information about token bucket mechanisms, see the "Policing and Shaping Overview" module.

```
Router(config)# class-map acgroup2
Router(config-cmap)# match access-group 2
Router(config-cmap)# exit
Router(config)# policy-map police
Router(config-pmap)# class acgroup2
Router(config-pmap-c)# police 8000 2000 4000 conform-action transmit exceed-action
set-qos-transmit 4 violate-action drop
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface fastethernet 0/0
Router(config-if)# service-policy input police
```

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Quality of Service Solutions Command Reference* at http://www.cisco.com/en/US/docs/ios/qos/command/reference/qos_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

- **police**
- **show policy-map**
- **show policy-map interface**

Glossary

average rate—Maximum long-term average rate of conforming traffic.

conform action—Action to take on packets with a burst size below the rate allowed by the rate limit.

DSCP—differentiated services code point

exceed action—Action to take on packets that exceed the rate limit.

excess burst size—Bytes allowed in a burst before all packets will exceed the rate limit.

normal burst size—Bytes allowed in a burst before some packets will exceed the rate limit. Larger bursts are more likely to exceed the rate limit.

QoS group—Internal QoS group ID for a packet used to determine weighted fair queuing characteristics for that packet.

policing policy—Rate limit, conform actions, and exceed actions that apply to traffic matching a certain criteria.

Versatile Interface Processor (VIP)—Interface card used by Cisco 7500 series and Cisco 7000 series with RSP7000 routers.

Two-Rate Policer

First Published: October 15, 2001
Last Updated: May 5, 2008

This document describes the Two-Rate Policer feature and explains how to configure the feature.

History for the Two-Rate Policer Feature

Release	Modification
12.2(4)T	This feature was introduced.
12.2(4)T3	Support for the Cisco 7500 series routers was added.
12.0(26)S	This feature was integrated into Cisco IOS Release 12.0(26)S for the Cisco 7200 and 7500 series routers.
12.2(28)SB	This feature was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This feature was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This feature was integrated into Cisco IOS Release 12.2(33)SXH.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Feature Overview

Networks police traffic by limiting the input or output transmission rate of a class of traffic based on user-defined criteria. Policing traffic allows you to control the maximum rate of traffic sent or received on an interface and to partition a network into multiple priority levels or class of service (CoS).

The Two-Rate Policer performs the following functions:

- Limits the input or output transmission rate of a class of traffic based on user-defined criteria.
- Marks packets by setting the IP precedence value, IP differentiated services code point (DSCP) value, Multiprotocol Label Switching (MPLS) experimental value, Quality of Service (QoS) group, ATM Cell Loss Priority (CLP) bit, and the Frame Relay Discard Eligibility (DE) bit.

With the Two-Rate Policer, you can enforce traffic policing according to two separate rates—committed information rate (CIR) and peak information rate (PIR). You can specify the use of these two rates, along with their corresponding values, by using two keywords, **cir** and **pir**, of the **police** command.

The Two-Rate Policer manages the maximum rate of traffic through a token bucket algorithm. The token bucket algorithm can use the user-configured values to determine the maximum rate of traffic allowed on an interface at a given moment in time. The token bucket algorithm is affected by all traffic entering or leaving the interface (depending on the location of the interface on which the Two-Rate Policer is configured) and is useful in managing network bandwidth in cases where several large packets are sent in the same traffic stream.

The token bucket algorithm provides users with three actions for each packet: a conform action, an exceed action, and an optional violate action. Traffic entering the interface with Two-Rate Policer configured is placed in to one of these categories. Within these three categories, users can decide packet treatments. For instance, packets that conform can be configured to be sent, packets that exceed can be configured to be sent with a decreased priority, and packets that violate can be configured to be dropped.

The Two-Rate Policer is 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 configurations, traffic that conforms is sent and traffic that exceeds is sent with a decreased priority or is dropped. Users can change these configuration options to suit their network needs.

Note Additionally, the Two-Rate Policer enables you to implement Differentiated Services (DiffServ) Assured Forwarding (AF) Per-Hop Behavior (PHB) traffic conditioning. For more information about DiffServ, see the “Implementing DiffServ for End-to-End Quality of Service Overview” module.

Note Starting with Cisco IOS Release 12.1(5)T, you can police traffic by using the Traffic Policing feature (sometimes referred to as the single-rate policer). The Two-Rate Policer (available with Cisco IOS Release 12.2(4)T) is in addition to the Traffic Policing feature, and it provides additional functionality. For more information about the Traffic Policing feature, see the “Traffic Policing” module

Benefits

Bandwidth Management Through Rate Limiting

This feature provides improved bandwidth management through rate limiting. Before this feature was available, you could police traffic with the single-rate Traffic Policing feature. The Traffic Policing feature provided a certain amount of bandwidth management by allowing you to set the peak burst size

(be). The Two-Rate Policer supports a higher level of bandwidth management and supports a sustained excess rate. With the Two-Rate Policer, you can enforce traffic policing according to two separate rates—CIR and PIR—specified in bits per second (bps).

Packet Marking Through IP Precedence, DSCP Value, MPLS Experimental Value, and the QoS Group Setting

In addition to rate-limiting, the Two-Rate Policer allows you to independently mark the packet according to whether the packet conforms, exceeds, or violates a specified rate. Packet marking also allows you to partition your network into multiple priority levels or classes of service (CoS).

- Use the Two-Rate Policer to set the IP precedence value, the IP DSCP value, or the MPLS experimental value for packets that enter the network. Then networking devices within your network can use the this setting to determine how the traffic should be treated. For example, the Weighted Random Early Detection (WRED) feature uses the IP precedence value to determine the probability that a packet will be dropped.
- Use the Two-Rate Policer to assign packets to a QoS group. The router uses the QoS group to determine how to prioritize packets within the router.

If you want to mark traffic but do not want to use the Two-Rate Policer, see the "Marking Network Traffic" module.

Packet Marking for Frame Relay Frames

The Two-Rate Policer allows users to mark the Frame Relay DE bit of the Frame Relay frame. The Frame Relay DE bit is one bit and, therefore, can be set to either 0 or 1. In congested environments, frames that have the DE bit set to 1 are discarded before frames that have the DE bit set to 0.

Packet Marking for ATM Cells

The Two-Rate Policer allows users to mark the ATM CLP bit in ATM cells. The ATM CLP bit is used to prioritize packets in ATM networks. The ATM CLP bit is one bit and, therefore, can be set to either 0 or 1. In congested environments, cells that have the ATM CLP bit set to 1 are discarded before cells that have the ATM CLP bit set to 0.

Restrictions

The following restrictions apply to the Two-Rate Policer:

- On a Cisco 7500 series router, traffic policing can monitor Cisco Express Forwarding (CEF) or Distributed CEF (dCEF) switching paths only. To use the Two-Rate Policer, CEF or dCEF must be configured on both the interface receiving the packet and the interface sending the packet.
- On a Cisco 7500 series router, traffic policing cannot be applied to packets that originated from or are destined to a router.
- Two-rate policing can be configured on an interface, a subinterface, a Frame Relay data-link connection identifier (DLCI), and an ATM permanent virtual circuit (PVC).
- Two-rate policing is not supported on the following interfaces:
  - Fast EtherChannel
  - PRI
  - Any interface on a Cisco 7500 series router that does not support CEF or dCEF

Prerequisites

Supported Platforms

- Cisco 2600 series
- Cisco 3620
- Cisco 3640
- Cisco 7100 series
- Cisco 7200 series
- Cisco 7500 series (VIP-based platform only)

Note To use the *set-clp-transmit* action available with this feature, the Enhanced ATM Port Adapter (PA-A3) is required. Therefore, the *set-clp-transmit* action is not supported on any platform that does not support the PA-A3 adapter (such as the Cisco 2600 series router, the Cisco 3620 router, and the 3640 router). For more information, see the documentation for your specific router.

- On a Cisco 7500 series router, CEF or dCEF must be configured on the interface before you can use the Two-Rate Policer. For additional information on CEF or dCEF, see the "Cisco Express Forwarding Features Roadmap" module.
- To configure the Two-Rate Policer, a traffic class and a service policy must be created, and the service policy must be attached to a specified interface. These tasks are performed using the Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC). For information on the MQC, see the "Applying QoS Features Using the MQC" module.

Configuration Tasks

See the following sections for configuration tasks for the Two-Rate Policer feature. Each task in the list is identified as either required or optional.

- Configuring the Two-Rate Policer (required)
- Verifying the Two-Rate Policer Configuration (optional)

Configuring the Two-Rate Policer

The Two-Rate Policer is configured in the service policy. To configure the Two-Rate Policer, use the following command in policy-map class configuration mode.

Command	Purpose
`Router(config-pmap-c)# police cir` *cir* `[bc` *conform-burst*`] pir` *pir* `[be` *peak-burst*`]`	Specifies that both the CIR and the PIR are to be used for two-rate traffic policing. The **bc** and **be** keywords and their associated arguments (*conform-burst* and *peak-burst*, respectively) are optional.

Although not required for configuring the Two-Rate Policer, the command syntax of the **police** command also allows you to specify the action to be taken on a packet when you enable an optional *action* argument. The resulting action corresponding to the keyword choices are listed in Table 1.

Table 1 police Command Action Keywords

Keyword	Resulting Action
drop	Drops the packet.
set-clp-transmit	Sets the ATM CLP bit from 0 to 1 on the ATM cell and sends the packet with the ATM CLP bit set to 1.
set-dscp-transmit *new-dscp*	Sets the IP DSCP value and sends the packet with the new IP DSCP value setting.
set-frde-transmit	Sets the Frame Relay DE bit from 0 to 1 on the Frame Relay frame and sends the packet with the DE bit set to 1.
set-mpls-exp-transmit	Sets the MPLS experimental bits from 0 to 7 and sends the packet with the new MPLS experimental bit value setting.
set-prec-transmit *new-prec*	Sets the IP precedence and sends the packet with the new IP precedence value setting.
set-qos-transmit *new-qos*	Sets the QoS group value and sends the packet with the new QoS group value setting.
transmit	Sends the packet with no alteration.

The Two-Rate Policer works by using a token bucket mechanism. There are currently two types of token bucket algorithms: a single token bucket algorithm (available through the Traffic Policing feature) and a two token bucket algorithm (available through the Two-Rate Policer).

For more information about token bucket mechanisms, see the "Policing and Shaping Overview" module.

Verifying the Two-Rate Policer Configuration

To verify that the Two-Rate Policer is configured on your interface, use the following command in user EXEC or privileged EXEC mode:

Command	Purpose
`Router# show policy-map interface`	Displays statistics and configurations of all input and output policies attached to an interface.

Troubleshooting Tips

- Check the interface type. Verify that your interface is not listed as a nonsupported interface in the Restrictions section of this module.
- For input traffic policing on a Cisco 7500 series router, verify that CEF or dCEF is configured on the interface on which traffic policing is configured.
- For output traffic policing on a Cisco 7500 series router, ensure that the incoming traffic is CEF-switched or dCEF-switched. Traffic policing cannot be used on the switching path unless CEF or dCEF switching is enabled.

Monitoring and Maintaining the Two-Rate Policer

To monitor and maintain the Two-Rate Policer, use the following user EXEC or privileged EXEC mode commands.

Command	Purpose
`Router# show policy-map`	Displays all configured policy maps.
`Router# show policy-map` *policy-map-name*	Displays the user-specified policy map.
`Router# show policy-map interface`	Displays statistics and configurations of all input and output policies that are attached to an interface.

Configuration Examples

This section provides the following configuration example:

- Limiting the Traffic Using a Policer Class: Example

Limiting the Traffic Using a Policer Class: Example

In this example, the Two-Rate Policer is configured on a class to limit traffic to an average committed rate of 500 kbps and a peak rate of 1 Mbps.

```
Router(config)# class-map police
Router(config-cmap)# match access-group 101
Router(config-cmap)# policy-map policy1
Router(config-pmap)# class police
Router(config-pmap-c)# police cir 500000 bc 10000 pir 1000000 be 10000 conform-action
transmit exceed-action set-prec-transmit 2 violate-action drop

Router(config)# interface serial3/0
Router(config-if)# service-policy output policy1
Router(config-if)# end

Router# show policy-map policy1

 Policy Map policy1
  Class police
   police cir 500000 conform-burst 10000 pir 1000000 peak-burst 10000 conform-action
transmit exceed-action set-prec-transmit 2 violate-action drop
```

Traffic marked as conforming to the average committed rate (500 kbps) will be sent as is. Traffic marked as exceeding 500 kbps, but not exceeding 1 Mbps, will be marked with IP Precedence 2 and then sent. All traffic exceeding 1 Mbps will be dropped. The burst parameters are set to 10000 bytes.

In the following example, 1.25 Mbps of traffic is sent ("offered") to a *policer* class.

```
Router# show policy-map interface serial3/0

 Serial3/0

  Service-policy output: policy1

   Class-map: police (match all)
    148803 packets, 36605538 bytes
    30 second offered rate 1249000 bps, drop rate 249000 bps
    Match: access-group 101
    police:
     cir 500000 bps, conform-burst 10000, pir 1000000, peak-burst 100000
     conformed 59538 packets, 14646348 bytes; action: transmit
     exceeded 59538 packets, 14646348 bytes; action: set-prec-transmit 2
     violated 29731 packets, 7313826 bytes; action: drop
     conformed 499000 bps, exceed 500000 bps violate 249000 bps

   Class-map: class-default (match-any)
    19 packets, 1990 bytes
    30 seconds offered rate 0 bps, drop rate 0 bps
    Match: any
```

The Two-Rate Policer marks 500 kbps of traffic as conforming, 500 kbps of traffic as exceeding, and 250 kbps of traffic as violating the specified rate. Packets marked as conforming will be sent as is, and packets marked as exceeding will be marked with IP Precedence 2 and then sent. Packets marked as violating the specified rate are dropped.

Additional References

The following sections provide references related to the Two-Rate Policer feature.

Related Documents

Related Topic	Document Title
MQC	• "Applying QoS Features Using the MQC" module
QoS features such as class-based weighted fair queueing (CBWFQ), traffic marking, and traffic policing	• "Configuring Weighted Fair Queueing" module • "Marking Network Traffic" module • "Traffic Policing" module

Standards

Standard	Title
None	—

MIBs

MIB	MIBs Link
• CISCO-CLASS-BASED-QOS-MIB • CISCO-CLASS-BASED-QOS-CAPABILITY-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2698	*A Two Rate Three Color Marker*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

This feature uses no new or modified commands.

Control Plane Policing

First Published: January 19, 2006
Last Updated: February 27, 2009

The Control Plane Policing feature allows users to configure a quality of service (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 denial-of-service (DoS) attacks. In this way, the control plane (CP) can help maintain packet forwarding and protocol states despite an attack or heavy traffic load on the router or switch.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "Feature Information for Control Plane Policing" section on page 20.

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

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Control Plane Policing

The Modular Quality of Service (QoS) Command-Line interface (CLI) (MQC) is used to configure the packet classification and policing functionality of the Control Plane Policing feature.

Before configuring Control Plane Policing (CoPP), you should understand the procedures for using the MQC. For information about the MQC, see the "Applying QoS Features Using the MQC" module.

Restrictions for Control Plane Policing

Aggregate and Distributed Control Plane Policing

Aggregate policing is supported in Cisco IOS Release 12.0(29)S, Cisco IOS Release 12.2(18)S, Cisco IOS Release 12.3(4)T, and later releases.

Distributed policing is supported only in Cisco IOS Release 12.0(30)S and later Cisco IOS 12.0S releases.

Output Rate-Limiting Support

Output rate-limiting is performed in silent (packet discard) mode. Silent mode enables a router to silently discard packets using policy maps applied to output control plane traffic with the **service-policy output** command. For more information, see the "Output Rate-Limiting and Silent Mode Operation" section on page 10.

Output rate-limiting (policing) in silent mode is supported only in:

- Cisco IOS Release 12.2(25)S and later Cisco IOS 12.2S releases
- Cisco IOS Release 12.3(4)T and later Cisco IOS 12.3T releases

Output rate-limiting is not supported for distributed control plane services in Cisco IOS 12.0S releases or in Cisco IOS 12.2SX releases.

Output rate-limiting is not supported on the Cisco 7500 series and Cisco 10720 Internet router.

MQC Restrictions

The Control Plane Policing feature requires the MQC to configure packet classification and policing. All restrictions that apply when you use the MQC to configure policing also apply when you configure control plane policing. Only two MQC actions are supported in policy maps—**police** and **drop**.

Note On the Cisco 10720 Internet router, only the **police** command, not the **drop** command, is supported in policy maps. In addition, in a QoS service policy that is attached to the Cisco 10720 control plane, the **police** command does not support **set** actions as arguments in **conform-action**, **exceed-action**, and **violate-action** parameters.

Features that require Network-Based Application Recognition (NBAR) classification may not work well at the control plane level. The following classification (match) criteria are supported on all platforms:

- Standard and extended IP access lists (ACLs).
- In class-map configuration mode: **match ip dscp**, **match ip precedence**, and **match protocol arp,** and **match protocol pppoe** commands.

Note In the Cisco IOS 12.2SX release, the **match protocol arp** command is not supported.

On the Cisco 10720 Internet router, the following MQC commands are also supported in class-map configuration mode: **match input-interface**, **match mpls experimental**, **match protocol ipv6**, and **match qos-group**. When using these commands for control plane policing on the Cisco 10720 Internet router, note the following restrictions:

- Packet classification using match criteria is not supported for packets that cannot be classified in the Cisco 10720 data path, such as unknown Layer 2 encapsulation and IP options.
- The following IPv6 fields are not be supported in packet classification for IPv6 QoS on the Cisco 10720 Internet router and are, therefore, not supported for control plane policing:
  - IPv6 source and destination addresses
  - Layer 2 class of service (CoS)
  - IPv6 routing header flag
  - IPv6 undetermined transport flag
  - IPv6 flow label
  - IP Real-Time transport Protocol (RTP)

Note Packets that are not supported for QoS packet classification on the Cisco 10720 Internet router are not policed in the default traffic class for control plane policing.

CISCO-CLASS-BASED-QOS-MIB Control Plane Support

In Cisco IOS Release 12.3(7)T and later Cisco IOS 12.3T releases, the CISCO-CLASS-BASED-QOS-MIB is extended to manage control plane QoS policies and provide information about the control plane.

Cisco IOS Release 12.2(18)SXD1

In Cisco IOS Release 12.2(18)SXD1 and later releases, Hardware Control Plane Interface for Control Plane Policing has the following restrictions:

- Supported only with Supervisor Engine 720. Not supported with Supervisor Engine 2.
- Does not support CoPP output rate-limiting (policing).
- Does not support the CoPP silent operation mode.
- Cisco IOS Release 12.2(18)SXD1 and later releases automatically install the CoPP service policy on all DFC-equipped switching modules.

For more information about control plane policing in Cisco IOS Release 12.2(18)SXD1 and later releases, see either of these publications:

- For Catalyst 6500 series switches, see the "Configuring Control Plane Policing (CoPP)" module.
- For Cisco 7600 series routers, see the "Configuring Denial of Service Protection" module.

Information About Control Plane Policing

To configure the Control Plane Policing feature, you should understand the following concepts:

Benefits of Control Plane Policing

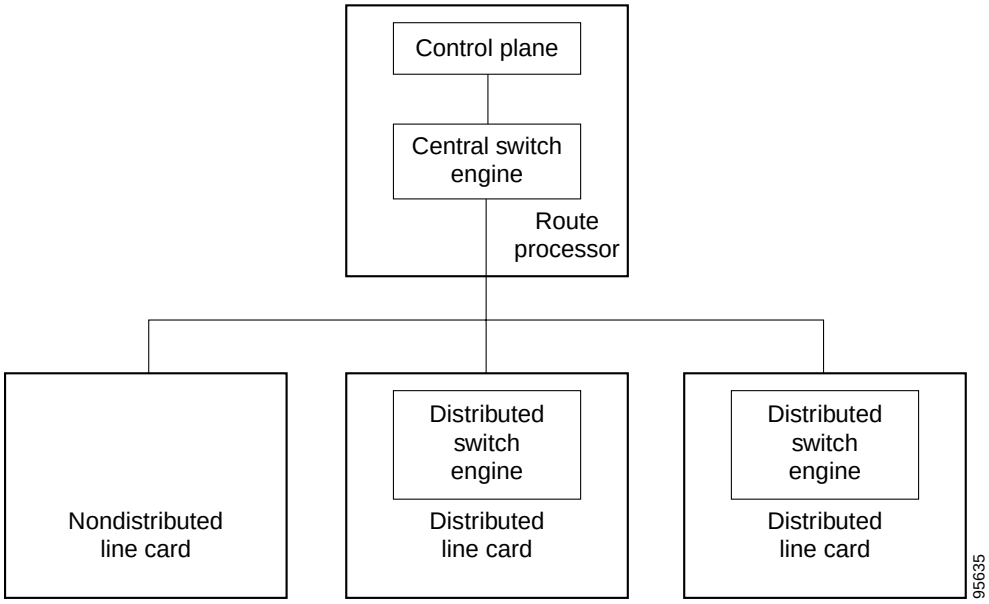
Configuring the Control Plane Policing feature on your Cisco router or switch provides the following benefits:

- Protection against DoS attacks at infrastructure routers and switches
- QoS control for packets that are destined to the control plane of Cisco routers or switches
- Ease of configuration for control plane policies
- Better platform reliability and availability

Terms to Understand

Because different platforms can have different architectures, the following set of terms is defined. Figure 1 illustrates how control plane policing works.

Figure 1 Layout of Control Plane, Central Switch Engine, Distributed Switch Engines, and Line Cards on a Router



- Control plane (CP)—A collection of processes that run at the process level on the route processor (RP). These processes collectively provide high-level control for most Cisco IOS functions.
- Central switch engine—A device that is responsible for high-speed routing of IP packets. It also typically performs high-speed input and output services for nondistributed interfaces. (See nondistributed line cards.) The central switch engine is used to implement aggregate CP protection for all interfaces on the router.

Note All IP packets that are destined for the CP should pass through the central switch engine before they are forwarded to the process level.

On the Cisco 10720 Internet router, control plane policing is implemented on Cisco Parallel eXpress Forwarding (PXF) in a Toaster-based architecture. PXF is a hardware-based central switch engine that can filter traffic at a higher rate than the route processor. PXF switches all data traffic separately from the route processor. PXF packet processing occurs at an intermediate step between the nondistributed line cards and the route processor shown in Figure 1. In addition to the regular punting, PXF also punts certain types of packets (such as unknown Layer 2 encapsulation and packets with IP options) to the RP for further processing at interrupt level.

Note On the Cisco 10720 Internet router, you can configure enhanced RP protection by using the **ip option drop** command to drop IPv4 packets with IP options that are punted to the RP by PXF. Tunneled IPv4 packets and IPv4 packets with an unsupported encapsulation method are not dropped. For more information, see the "ACL IP Options Selective Drop" module.

- Distributed switch engine—A device that is responsible for high-speed switching of IP packets on distributed line cards without using resources from the central switch engine. It also typically performs input and output services for the line card. Each distributed switch engine is used to implement distributed CP services for all ports on a line card. Input CP services distribute the processing load across multiple line cards and conserve vital central switch engine resources. Distributed CP services are optional; however, they provide a more refined level of service than aggregate services.
- Nondistributed line cards—Line cards that are responsible for receiving packets and occasionally performing input and output services. All packets must be forwarded to the central switch engine for a routing or switching decision. Aggregate CP services provide coverage for nondistributed line cards.

Note Distributed CP services are supported only in Cisco IOS Release 12.0(30)S and later 12.0S releases.

Control Plane Security and Packet QoS Overview

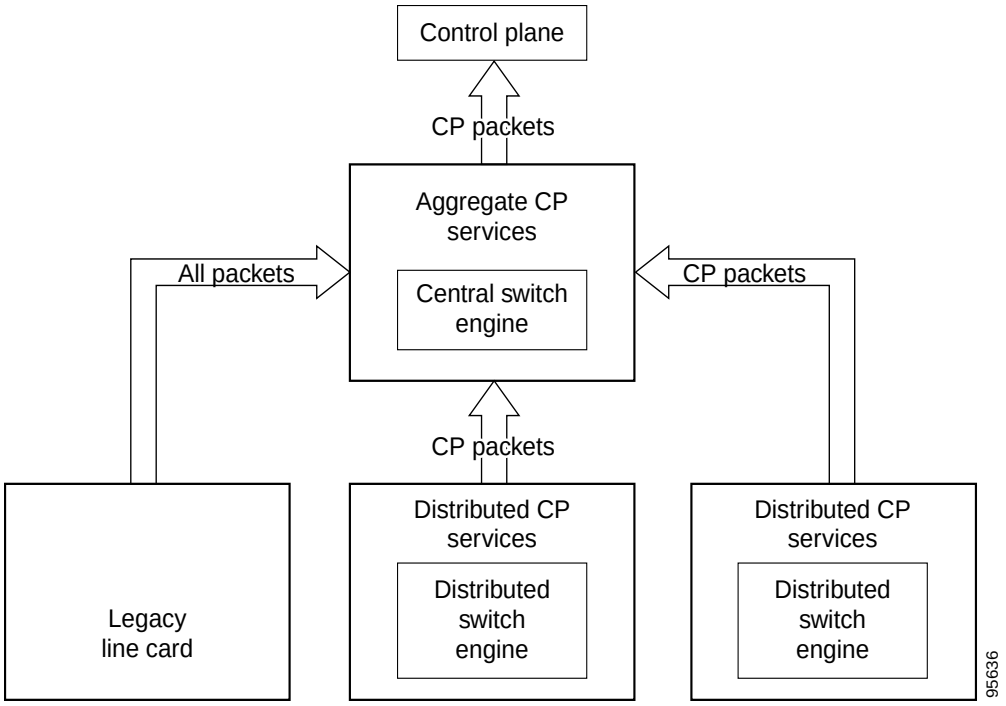
To protect the CP on a router from DoS attacks and to provide packet QoS, the Control Plane Policing feature treats the CP as a separate entity with its own ingress (input) and egress (output) ports, which are like ports on a router and switch. Because the Control Plane Policing feature treats the CP as a separate entity, a set of rules can be established and associated with the ingress and egress ports of the CP.

These rules are applied only after the packet has been determined to have the CP as its destination or when a packet exits the CP. Thereafter, you can configure a service policy to prevent unwanted packets from progressing after a specified rate limit has been reached; for example, a system administrator can limit all TCP/SYN packets that are destined for the CP to a maximum rate of 1 megabit per second.

Input CP services are executed after router input port services have been performed and after a routing decision on the input path has been made. As shown in Figure 2, CP security and packet QoS are applied on:

- An aggregate level by the central switch engine and applied to all CP packets received from all line cards on the router (see the "Aggregate Control Plane Services" section on page 7).
- A distributed level by the distributed switch engine of a line card and applied to all CP packets received from all interfaces on the line card (see the "Distributed Control Plane Services" section on page 8).

Figure 2 Input Control Plane Services: Aggregate and Distributed Services



The following types of Layer 3 packets are forwarded to the control plane and processed by aggregate and distributed control plane policing:

- Routing protocol control packets
- Packets destined for the local IP address of the router
- Packets from management protocols (such as Simple Network Management Protocol [SNMP], Telnet, and secure shell [SSH])

Note Ensure that Layer 3 control packets have priority over other packet types that are destined for the control plane.

Aggregate Control Plane Services

Aggregate control plane services provide control plane policing for all CP packets that are received from all line-card interfaces on the router.

The central switch engine executes normal input port services and makes routing decisions for an incoming packet: if the packet is destined for the CP, aggregate services are performed. Because CP traffic from all line cards must pass through aggregate CP services, these services manage the cumulative amount of CP traffic that reaches the CP.

Aggregate CP service steps are as follows:

1. The line card receives a packet and delivers it to the central switch engine.

Note Before the packet is sent to the central switch engine, additional processing may be necessary for platforms that support hardware-level policing or platform-specific aggregate policing. It is possible that the packet may undergo multiple checks before it undergoes the generic Cisco IOS check.

2. The interfaces perform normal (interface-level) input port services and QoS.
3. The central switch engine performs Layer 3 switching or makes a routing decision, determining whether or not the packet is destined for the CP.
4. The central switch engine performs aggregate CP services for all CP packets.
5. On the basis of the results of the aggregate CP services, the central switch engine either drops the packet or delivers the packet to the CP for final processing.

Functionality Highlights of Aggregate CP Services

The following list highlights the functionality of aggregate CP services:

- Aggregate CP services are defined for a single input interface, such as the CP, and represent an aggregate for all ports on a router.
- Modular QoS is used to define CP services. Class maps and policy maps for both DoS protection and packet QoS are defined for a single aggregate CP service policy.
- Modular QoS does not prevent a single bad port from consuming all allocated bandwidth. Class maps that match an interface or subinterface may be able to constrain the contribution of each interface through an interface-specific policy map.

Distributed Control Plane Services

Distributed control plane services provide control plane policing for all CP packets that are received from the interfaces on a line card.

A distributed switch engine executes normal input port services and makes routing decisions for a packet: if the packet is destined for the CP, distributed CP services are performed. Afterwards, CP traffic from each line card is forwarded to the central switch engine where aggregate CP services are applied.

Note Distributed CP services may also forward conditioned packets to the central switch engine. In this case, aggregate CP services are also performed on the conditioned CP traffic.

Distributed CP service steps are as follows:

1. A line card receives a packet and delivers it to the distributed switch engine.
2. The distributed switch engine performs normal (interface-level) input port services and QoS.
3. The distributed switch engine performs Layer 2 or Layer 3 switching or makes a routing decision, determining whether the packet is destined for the CP.
4. The distributed switch engine performs distributed CP services for all CP packets.

5. On the basis of the results of the distributed CP services, the distributed switch engine either drops the packet or marks the packet and delivers it to the central switch engine for further processing.
6. The central switch engine performs aggregate CP services and delivers the packet to the CP for final processing.

Functionality Highlights of Distributed CP Services

The following list highlights the functionality of distributed CP services:

- Distributed CP services are defined for a single input interface, such as the distributed CP, and represent an aggregate for all ports on a line card.
- The MQC is used to define CP services. Class maps and policy maps for both DoS protection and packet QoS are defined for a single distributed CP service policy. Each line card may have a unique CP service policy that applies traffic classifications, QoS policies, and DoS services to packets received from all ports on the line card in an aggregate way.
- The MQC does not prevent one bad port from consuming all allocated bandwidth on a line card. Class maps that match an interface or subinterface may be able to constrain the contribution of each interface through an interface-specific policy map.
- Distributed CP services allow you to limit the number of CP packets forwarded from a line card to the central switch engine. The total number of CP packets received from all line cards on a router may exceed aggregate CP levels.

Usage of Distributed CP Services

The purpose of CP protection and packet QoS is to apply sufficient control to the packets that reach the control plane. To successfully configure this level of CP protection, you must:

- Apply traditional QoS services using the MQC to CP packets.
- Protect the path to the control plane against indiscriminate packet dropping due to resource exhaustion. If packets are not dropped according to user-defined QoS policies, but are dropped due to a resource limitation, the QoS policy is not maintained.

Distributed CP services allow you to configure specific CP services that are enforced at the line-card level and are required for the following reasons:

- While under a DoS attack, line-card resources may be consumed. In this case, you must configure a drop policy to identify important packets. The drop policy ensures that all important packets arrive to the central switch engine for aggregate CP protection and arrive later to the CP. Distributed CP services allow routers to apply the appropriate drop policy when resources are consumed and therefore maintain the desired QoS priorities. If a line card indiscriminately drops packets, the aggregate CP filter becomes ineffective and the QoS priorities are no longer maintained.
- It is not possible to prevent one interface from consuming all aggregate CP resources. A DoS attack on one port may negatively impact CP processing of traffic from other ports. Distributed CP services allow you to limit the amount of important traffic that is forwarded by a line card to the CP. For example, you can configure a layered approach in which the combined rates of all line cards are over-subscribed compared to the aggregate rate. The rate of each individual line card would be below the aggregate rate, but combined together, the rates of all line cards exceed it. This over-subscription model is commonly used for other resource-related functions and helps limit the contribution of CP packets from any one line card.

- Distributed CP services provide for slot-level (line-card) filtering. Customer-facing interfaces may have greater security requirements (with more restrictions or for billing reasons) than network-facing interfaces to backbone devices.
- Because distributed CP protection allows you to configure packet filters on a per-line-card basis, processing cycles on line cards may offload aggregate level processing. You can configure Border Gateway Protocol (BGP) filtering at the distributed level for interfaces that use BGP, allowing the aggregate level to filter packets with the remaining filter requirements. Or you can configure identical filters for distributed and aggregate CP services with a distributed packet marking scheme that informs the aggregate filter that a packet has already been checked. Distributed CP service processing further reduces aggregate processing and can significantly reduce the load on aggregate CP services.

Output Rate-Limiting and Silent Mode Operation

A router is automatically enabled to silently discard packets when you configure output policing on control plane traffic using the **service-policy output** *policy-map-name* command.

Rate-limiting (policing) of output traffic from the CP is performed in silent mode. In silent mode, a router that is running Cisco IOS software operates without sending any system messages. If a packet that is exiting the control plane is discarded for output policing, you do not receive an error message.

When control plane policing is configured for output traffic, error messages are not generated in the following cases:

- Traffic that is being transmitted to a port to which the router is not listening
- A connection to a legitimate address and port that is rejected because of a malformed request

Note The silent mode functionality and output policing on CP traffic are supported only in:

- Cisco IOS Release 12.2(25)S and later Cisco IOS 12.2S releases
- Cisco IOS Release 12.3(4)T and later Cisco IOS 12.3T releases

Silent mode and output policing on CP traffic are not supported for distributed control plane services.

How to Use Control Plane Policing

This section documents the following procedures:

- Defining Aggregate Control Plane Services, page 11 (required)
- Defining Distributed Control Plane Services, page 12 (required)
- Verifying Aggregate Control Plane Services, page 13 (optional)
- Verifying Distributed Control Plane Services, page 15 (optional)

Defining Aggregate Control Plane Services

To configure aggregate CP services, such as packet rate control and silent packet discard, for the active route processor, complete the following steps.

Prerequisites

Before you enter control-plane configuration mode to attach an existing QoS policy to the control plane, you must first create the policy using MQC to define a class mapand policy map for control plane traffic.

For information about how to classify traffic and create a QoS policy, see the "Applying QoS Features Using the MQC" module.

Restrictions

- Platform-specific restrictions, if any, are checked when the service policy is applied to the control plane interface.
- Support for output policing is available only in Cisco IOS Release 12.3(4)T and later T-train releases. (Note that output policing does not provide any performance benefits. It simply controls the information that is leaving the device.)

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **control-plane**
4. **service-policy** {**input** | **output**} *policy-map-name*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`control-plane` **Example:** `Router(config)# control-plane`	Enters control-plane configuration mode (a prerequisite for Step 4).

	Command or Action	Purpose
Step 4	**service-policy** {**input** | **output**} *policy-map-name* **Example:** `Router(config-cp)# service-policy input control-plane-policy`	Attaches a QoS service policy to the control plane. Note the following points: • **input**—Applies the specified service policy to packets received on the control plane. • **output**—Applies the specified service policy to packets transmitted from the control plane and enables the router to silently discard packets. • *policy-map-name*—Name of a service policy map (created using the **policy-map** command) to be attached. The name can be a maximum of 40 alphanumeric characters.
Step 5	**end** **Example:** `Router(config-cp)# end`	(Optional) Returns to privileged EXEC mode.

Defining Distributed Control Plane Services

To configure distributed CP services, such as packet rate control, for packets that are destined for the CP and sent from the interfaces on a line card, complete the following steps.

Prerequisites

Before you enter control-plane configuration mode to attach an existing QoS policy for performing distributed control-plane services, you must first create the policy using MQC to define a class map and policy map for control-plane traffic.

For information about how to classify traffic and create a QoS policy, see the "Applying QoS Features Using the MQC" module.

Restrictions

- Platform-specific restrictions, if any, are checked when the service policy is applied to the control plane interface.
- Support for output policing is available only in Cisco IOS Release 12.3(4)T and later T-train releases. (Note that output policing does not provide any performance benefits. It simply controls the information that is leaving the device.)
- With Cisco IOS 12.2SX releases, the Supervisor Engine 720 automatically installs the service policy on all DFC-equipped switching modules.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **control-plane** [**slot** *slot-number*]

4. **service-policy input** *policy-map-name*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**control-plane** [**slot** *slot-number*] **Example:** `Router(config)# control-plane slot 3`	Enters control-plane configuration mode, which allows you to optionally attach a QoS policy (used to manage CP traffic) to the specified slot. • Enter the **slot** keyword and the slot number, as applicable.
Step 4	**service-policy input** *policy-map-name* **Example:** `Router(config-cp)# service-policy input control-plane-policy`	Attaches a QoS policy map to filter and manage CP traffic on a specified line card before the aggregate CP policy is applied. Note the following points: • **input**—Applies the specified policy map using the distributed switch engine to CP packets that are received from all interfaces on the line card. • *policy-map-name*—Name of a service policy map (created using the **policy-map** command) to be attached. The name can be a maximum of 40 alphanumeric characters. **Note** The **service-policy output** *policy-map-name* command is not supported for applying a QoS policy map for distributed control plane services.
Step 5	**end** **Example:** `Router(config-cp)# end`	(Optional) Returns to privileged EXEC mode.

Verifying Aggregate Control Plane Services

To display information about the service policy attached to the control plane for aggregate CP services, complete the following steps.

SUMMARY STEPS

1. **enable**
2. **show policy-map control-plane** [**all**] [**input** [**class** *class-name*] | **output** [**class** *class-name*]]

3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show policy-map control-plane** [**all**] [**input** [**class** *class-name*] | **output** [**class** *class-name*]] **Example:** `Router# show policy-map control-plane all`	Displays information about the control plane. Note the following points: • **all**—(Optional) Service policy information about all QoS policies used in aggregate and distributed CP services. • **input**—(Optional) Statistics for the attached input policy. • **output**—(Optional) Statistics for the attached output policy. • **class** *class-name*—(Optional) Name of the traffic class whose configuration and statistics are displayed.
Step 3	**exit** **Example:** `Router(config-cp)# exit`	(Optional) Exits privileged EXEC mode.

Examples

The following example shows that the policy map TEST is associated with the control plane. This policy map polices traffic that matches the class map TEST, while allowing all other traffic (that matches the class map "class-default") to go through as is.

```
Router# show policy-map control-plane

Control Plane

Service-policy input:TEST

Class-map:TEST (match-all)
      20 packets, 11280 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match:access-group 101
      police:
        8000 bps, 1500 limit, 1500 extended limit
        conformed 15 packets, 6210 bytes; action:transmit
        exceeded 5 packets, 5070 bytes; action:drop
        violated 0 packets, 0 bytes; action:drop
        conformed 0 bps, exceed 0 bps, violate 0 bps

Class-map:class-default (match-any)
      105325 packets, 11415151 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match:any
```

Verifying Distributed Control Plane Services

To display information about the service policy attached to the control plane to perform distributed CP services, complete the following steps.

SUMMARY STEPS

1. **enable**
2. **show policy-map control-plane** [**all** | **slot** *slot-number*] [**input** [**class** *class-name*] | **output** [**class** *class-name*]]
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show policy-map control-plane** [**all**][**slot** *slot-number*] [**input** [**class** *class-name*]	**output** [**class** *class-name*]] **Example:** `Router# show policy-map control-plane slot 2`	Displays information about the service policy used to apply distributed CP services on the router. Note the following points: • **all**—(Optional) Service policy information about all QoS policies used in aggregate and distributed CP services. • **slot** *slot-number*—(Optional) Service policy information about the QoS policy map used to perform distributed CP services on the specified line card. • **input**—(Optional) Statistics for the attached input policy map. • **output**—(Optional) Statistics for the attached output policy map. • **class** *class-name*—(Optional) Name of the traffic class whose configuration and statistics are displayed.
Step 3	`exit` **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

Examples

The following example shows how to display information about the classes of CP traffic received from all interfaces on the line card in slot 1 to which the policy map TESTII is applied for distributed CP services. This policy map polices traffic that matches the traffic class TESTII, while allowing all other traffic (that matches the class map "class-default") to go through as is.

```
Router# show policy-map control-plane slot 1
```

```
Control Plane - slot 1

Service-policy input: TESTII (1048)

Class-map: TESTII (match-all) (1049/4)
        0 packets, 0 bytes
        5 minute offered rate 0 bps, drop rate 0 bps
        Match: protocol arp (1050)
        police:
            cir 8000 bps, bc 4470 bytes, be 4470 bytes
          conformed 0 packets, 0 bytes; actions:
            transmit
          exceeded 0 packets, 0 bytes; actions:
            drop
          violated 0 packets, 0 bytes; actions:
            drop
          conformed 0 bps, exceed 0 bps, violate 0 bps

Class-map: class-default (match-any) (1052/0)
        0 packets, 0 bytes
        5 minute offered rate 0 bps, drop rate 0 bps
        Match: any  (1053)
```

Configuration Examples for Control Plane Policing

This section contains examples that shows how to configure aggregate control plane services on both an input and an output interface:

- Configuring Control Plane Policing on Input Telnet Traffic: Example, page 16
- Configuring Control Plane Policing on Output ICMP Traffic: Example, page 17

Configuring Control Plane Policing on Input Telnet Traffic: Example

The following example shows how to apply a QoS policy for aggregate CP services to Telnet traffic that is received on the control plane. Trusted hosts with source addresses 10.1.1.1 and 10.1.1.2 forward Telnet packets to the control plane without constraint, while allowing all remaining Telnet packets to be policed at the specified rate.

```
! Allow 10.1.1.1 trusted host traffic.
Router(config)# access-list 140 deny tcp host 10.1.1.1 any eq telnet
! Allow 10.1.1.2 trusted host traffic.
Router(config)# access-list 140 deny tcp host 10.1.1.2 any eq telnet
! Rate-limit all other Telnet traffic.
Router(config)# access-list 140 permit tcp any any eq telnet
! Define class-map “telnet-class.”
Router(config)# class-map telnet-class
Router(config-cmap)# match access-group 140
Router(config-cmap)# exit
Router(config)# policy-map control-plane-in
Router(config-pmap)# class telnet-class
Router(config-pmap-c)# police 80000 conform transmit exceed drop
Router(config-pmap-c)# exit
Router(config-pmap)# exit
! Define aggregate control plane service for the active route processor.
Router(config)# control-plane
Router(config-cp)# service-policy input control-plane-in
Router(config-cp)# end
```

Configuring Control Plane Policing on Output ICMP Traffic: Example

The following example shows how to apply a QoS policy for aggregate CP services to Telnet traffic transmitted from the control plane. Trusted networks with source addresses 10.0.0.0 and 10.0.0.1 receive Internet Control Management Protocol (ICMP) port-unreachable responses without constraint, while allowing all remaining ICMP port-unreachable responses to be dropped:

```
! Allow 10.0.0.0 trusted network traffic.
Router(config)# access-list 141 deny icmp 10.0.0.0 0.0.0.255 any port-unreachable
! Allow 10.0.0.1 trusted network traffic.
Router(config)# access-list 141 deny icmp 10.0.0.1 0.0.0.255 any port-unreachable
! Rate-limit all other ICMP traffic.
Router(config)# access-list 141 permit icmp any any port-unreachable
Router(config)# class-map icmp-class
Router(config-cmap)# match access-group 141
Router(config-cmap)# exit
Router(config)# policy-map control-plane-out
! Drop all traffic that matches the class "icmp-class."
Router(config-pmap)# class icmp-class
Router(config-pmap-c)# drop
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# control-plane
! Define aggregate control plane service for the active route processor.
Router(config-cp)# service-policy output control-plane-out
Router(config-cp)# end
```

Additional References

The following sections provide references related to the Control Plane Policing feature.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
QoS features overview	"Quality of Service Overview" module
MQC	"Applying QoS Features Using the MQC" module
Security features overview	"Security Overview" module
Control plane policing in Cisco IOS Release 12.2(18)SXD1 and later releases	For Catalyst 6500 series switches, see the "Configuring Control Plane Policing (CoPP)" module. For Cisco 7600 series routers, see the "Configuring Denial of Service Protection" module.
Enhanced RP protection	"ACL IP Options Selective Drop" module

Standards

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

MIBs

MIB	MIBs Link
• CISCO-CLASS-BASED-QOS-MIB **Note** Supported only in Cisco IOS Release 12.3(7)T.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator, found at the following URL: http://www.cisco.com/go/mibs

RFCs

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

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Quality of Service Solutions Command Reference* at http://www.cisco.com/en/US/docs/ios/qos/command/reference/qos_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html.

- **control-plane**
- **service-policy** (control-plane)
- **show policy-map control-plane**

Feature Information for Control Plane Policing

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS, Catalyst OS, and Cisco IOS XE software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Control Plane Policing

Feature Name	Releases	Feature Information
Control Plane Policing	12.2(18)S 12.3(4)T 12.3(7)T 12.0(29)S 12.2(18)SXD1 12.0(30)S 12.2(27)SBC 12.0(32)S 12.3(31)SB2	The Control Plane Policing feature allows users to configure a quality of service (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 denial-of-service (DoS) attacks. For Release 12.2(18)S, this feature was introduced. For Release 12.3(4)T, this feature was integrated into Cisco IOS Release 12.3(4)T, and the output rate-limiting (silent mode operation) feature was added. For Release 12.3(7)T, the CISCO-CLASS-BASED-QOS-MIB was extended to manage control plane QoS policies, and the **police rate** command was introduced to support traffic policing on the basis of packets per second for control plane traffic. For Release 12.0(29)S, this feature was integrated into Cisco IOS Release 12.0(29)S. For Release 12.2(18)SXD1, this feature was integrated into Cisco IOS Release 12.2(18)SXD1. For Release 12.0(30)S, this feature was modified to include support for distributed control plane services on the Cisco 12000 series Internet router. For Release 12.2(27)SBC, this feature was integrated into Cisco IOS Release 12.2(27)SBC. For Release 12.0(32)S, this feature was modified to include support for aggregate control plane services on the Cisco 10720 Internet router. For Release 12.3(31)SB2, this feature was implemented on the Cisco 10000 series router for the PRE3.

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

Class-Based Policing

Feature History

Release	Modification
12.1(5)T	This command was introduced for Cisco IOS Release 12.1 T. A new Class-Based Policing algorithm was introduced. The **violate-action** option became available. This feature became available on Cisco 2600, 3600, 4500, 7200, and 7500 series routers.
12.2(2)T	The set-clp-transmit option for the *action* argument was added to the **police** command. The set-frde-transmit option for the *action* argument was added to the **police** command. The set-mpls-exp-transmit option for the *action* argument was added to the **police** command.
12.0(26)S	This feature was integrated into Cisco IOS Release 12.0(26)S for the Cisco 7200 and 7500 series routers. The name of the feature changed from *Traffic Policing* to *Class-Based Policing*.

Finding Support Information for Platforms and Cisco IOS Software Images

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

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Feature Overview

This feature module describes the Class-Based Policing feature. It includes information on the benefits of the feature, supported platforms, related documents, and so forth.

The Class-Based Policing feature performs the following functions:

- Limits the input or output transmission rate of a class of traffic based on user-defined criteria
- Marks packets by setting the ATM Cell Loss Priority (CLP) bit, Frame Relay Discard Eligibility (DE) bit, IP precedence value, IP differentiated services code point (DSCP) value, MPLS experimental value, and Quality of Service (QoS) group.

Class-Based Policing allows you to control the maximum rate of traffic transmitted or received on an interface. The Class-Based Policing feature is applied when you attach a traffic policy contain the Class-Based Policing configuration to an interface. A traffic policy is configured using the Modular Quality of Service Command-Line Interface (Modular QoS CLI). For information on configuring the Modular QoS CLI, see the "Applying QoS Features Using the MQC" module.

Benefits

Bandwidth Management Through Rate Limiting

Class-Based Policing allows you to control the maximum rate of traffic transmitted or received on an interface. Class-Based Policing is often configured on interfaces at the edge of a network to limit traffic into or out of the network. In most Class-Based Policing configurations, traffic that falls within the rate parameters is transmitted, whereas traffic that exceeds the parameters is dropped or transmitted with a different priority.

Packet Marking

Packet marking allows you to partition your network into multiple priority levels or classes of service (CoS). A packet is marked and these markings can be used to identify and classify traffic for downstream devices. In some cases, such as ATM Cell Loss Priority (CLP) marking or Frame Relay Discard Eligibility (DE) marking, the marking is used to classify traffic.

- Use Class-Based Policing to set the IP precedence or DSCP values for packets entering the network. Networking devices within your network can then use the adjusted IP precedence values to determine how the traffic should be treated. For example, the Weighted Random Early Detection (WRED) feature uses the IP precedence values to determine the probability that a packet will be dropped.
- Use Class-Based Policing to assign packets to a QoS group. The router uses the QoS group to determine how to prioritize packets within the router.

Traffic can be marked without using the Class-Based Policing feature. If you want to mark traffic but do not want to use Class-Based Policing, see the "Marking Network Traffic" module.

Packet Prioritization for Frame Relay Frames

The Class-Based Policing feature allows users to mark the Frame Relay DE bit of the Frame Relay frame. The Frame Relay DE bit is one bit and, therefore, can be set to either 0 or 1. In congested environments, frames with the DE bit set to 1 are discarded before frames with the DE bit set to 0.

Packet Prioritization for ATM Cells

The Class-Based Policing feature allows users to mark the ATM CLP bit in ATM cells. The ATM CLP bit is used to prioritize packets in ATM networks. The ATM CLP bit is one bit and, therefore, can be set to either 0 or 1. In congested environments, cells with the ATM CLP bit set to 1 are discarded before cells with the ATM CLP bit set to 0.

Restrictions

- To use the *set-clp-transmit* action available with this feature, the Enhanced ATM Port Adapter (PA-A3) is required. Therefore, the *set-clp-transmit* action is not supported on any platform that does not support the PA-A3 adapter (such as the Cisco 2600 series router, the Cisco 3640 router, and the 4500 series router). For more information, see the documentation for your specific router.
- On a Cisco 7500 series router, Class-Based Policing can monitor Cisco Express Forwarding (CEF) switching paths only. In order to use the Class-Based Policing feature, Cisco Express Forwarding must be configured on both the interface receiving the packet and the interface sending the packet.
- On a Cisco 7500 series router, Class-Based Policing cannot be applied to packets that originated from or are destined to a router.
- Class-Based Policing can be configured on an interface or a subinterface.
- Class-Based Policing is not supported on the following interfaces:
  - Fast EtherChannel
  - Tunnel

Note Class-Based Policing is supported on tunnels that are using the Cisco generic routing encapsulation (GRE) tunneling protocol.

  - PRI
  - Any interface on a Cisco 7500 series router that does not support Cisco Express Forwarding

Prerequisites

On a Cisco 7500 series router, Cisco Express Forwarding (CEF) must be configured on the interface before Class-Based Policing can be used.

For additional information on Cisco Express Forwarding, see the "Cisco Express Forwarding Features Roadmap" module.

Configuration Tasks

See the following sections for configuration tasks for the Class-Based Policing feature. Each task in the list indicates if the task is optional or required.

Configuring Traffic Policing

To successfully configure the Class-Based Policing feature, a traffic class and a traffic policy must be created, and the traffic policy must be attached to a specified interface. These tasks are performed using the Modular QoS CLI. For information on the Modular QoS CLI, see the "Applying QoS Features Using the MQC" module.

The Class-Based Policing feature is configured in the traffic policy. To configure the Class-Based Policing feature, use the following command in policy map configuration mode:

Command	Purpose
`Router(config-pmap-c)#` **police** *bps burst-normal burst-max* **conform-action** *action* **exceed-action** *action* **violate-action** *action*	Specifies a maximum bandwidth usage by a traffic class.

The Class-Based Policing feature works with a token bucket mechanism. There are currently two types of token bucket algorithms: a single token bucket algorithm and a two token bucket algorithm. A single token bucket system is used when the **violate-action** option is not specified, and a two token bucket system is used when the **violate-action** option is specified.

For more information about token bucket mechanisms, see the "Policing and Shaping Overview" module.

Verifying Traffic Policing

Use the **show policy-map interface** EXEC command to verify that the Class-Based Policing feature is configured on your interface. If the feature is configured on your interface, the **show policy-map interface** command output displays policing statistics:

```
Router# show policy-map interface
 Ethernet1/7
  service-policy output: x
    class-map: a (match-all)
      0 packets, 0 bytes
      5 minute rate 0 bps
      match: ip precedence 0
      police:
        1000000 bps, 10000 limit, 10000 extended limit
        conformed 0 packets, 0 bytes; action: transmit
        exceeded 0 packets, 0 bytes; action: drop
        conformed 0 bps, exceed 0 bps, violate 0 bps
```

Troubleshooting Tips

- Check the interface type. Verify that your interface is not mentioned in the nonsupported interface description in the "Restrictions" section of this module.
- For input Class-Based Policing on a Cisco 7500 series router, verify that CEF is configured on the interface where Class-Based Policing is configured.
- For output Class-Based Policing on a Cisco 7500 series router, ensure that the incoming traffic is CEF-switched. Class-Based Policing cannot be used on the switching path unless CEF switching is enabled.

Monitoring and Maintaining Traffic Policing

To monitor and maintain the Class-Based Policing feature, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router#` **show policy-map**	Displays all configured policy maps.
`Router#` **show policy-map** *policy-map-name*	Displays the user-specified policy map.
`Router#` **show policy-map interface**	Displays statistics and configurations of all input and output policies that are attached to an interface.

Configuration Examples

This section provides the following configuration example:

- Configuring a Service Policy that Includes Traffic Policing: Example, page 5

Configuring a Service Policy that Includes Traffic Policing: Example

In the following example, Class-Based Policing is configured with the average rate at 8000 bits per second, the normal burst size at 1000 bytes, and the excess burst size at 1000 bytes for all packets leaving Fast Ethernet interface 0/0.

For additional information on configuring traffic classes and traffic policies, see the "Applying QoS Features Using the MQC" module.

For more information about token bucket mechanisms, see the "Policing and Shaping Overview" module.

```
class-map access-match
    match access-group 1
    exit
policy-map police-setting
    class access-match
        police 8000 1000 1000 conform-action transmit exceed-action set-qos-transmit 1
        violate-action drop
        exit
    exit
    service-policy output police-setting
```

The treatment of a series of packets leaving Fast Ethernet interface 0/0 depends on the size of the packet and the number of bytes remaining in the conform and exceed token buckets. The series of packets are policed based on the following rules:

- If the previous arrival of the packet was at T1 and the current arrival of the packet is at T, the bucket is updated with T -T1 worth of bits based on the token arrival rate. The refill tokens are placed in the conform bucket. If the tokens overflow the conform bucket, the overflow tokens are placed in the exceed bucket. The token arrival rate is calculated as follows:

  (time between packets <which is equal to T - T1> * policer rate)/8 bytes

- If the number of bytes in the conform bucket B is greater than or equal to 0, the packet conforms and the conform action is taken on the packet. If the packet conforms, B bytes are removed from the conform bucket and the conform action is taken. The exceed bucket is unaffected in this scenario.
- If the number of bytes in the conform bucket B is less than 0, the excess token bucket is checked for bytes by the packet. If the number of bytes in the exceed bucket B is greater than or equal to 0, the exceed action is taken and B bytes are removed from the exceed token bucket. No bytes are removed from the conform bucket in this scenario.
- If the number bytes in the exceed bucket B is fewer than 0, the packet violates the rate and the violate action is taken. The action is complete for the packet.

In this example, the initial token buckets starts full at 1000 bytes. If a 450-byte packet arrives, the packet conforms because enough bytes are available in the conform token bucket. The conform action (send) is taken by the packet and 450 bytes are removed from the conform token bucket (leaving 550 bytes).

If the next packet arrives 0.25 seconds later, 250 bytes are added to the conform token bucket

((0.25 * 8000)/8), leaving 800 bytes in the conform token bucket. If the next packet is 900 bytes, the packet does not conform because only 800 bytes are available in the conform token bucket.

The exceed token bucket, which starts full at 1000 bytes (as specified by the excess burst size) is then checked for available bytes. Because enough bytes are available in the exceed token bucket, the exceed action (set the QoS transmit value of 1) is taken and 900 bytes are taken from the exceed bucket (leaving 100 bytes in the exceed token bucket.

If the next packet arrives 0.40 seconds later, 400 bytes are added to the token buckets ((.40 * 8000)/8). Therefore, the conform token bucket now has 1000 bytes (the maximum number of tokens available in the conform bucket) and 200 bytes overflow the conform token bucket (because it only 200 bytes were needed to fill the conform token bucket to capacity). These overflow bytes are placed in the exceed token bucket, giving the exceed token bucket 300 bytes.

If the arriving packet is 1000 bytes, the packet conforms because enough bytes are available in the conform token bucket. The conform action (transmit) is taken by the packet, and 1000 bytes are removed from the conform token bucket (leaving 0 bytes).

If the next packet arrives 0.20 seconds later, 200 bytes are added to the token bucket ((.20 * 8000)/8). Therefore, the conform bucket now has 200 bytes. If the arriving packet is 400 bytes, the packet does not conform because only 200 bytes are available in the conform bucket. Similarly, the packet does not exceed because only 300 bytes are available in the exceed bucket. Therefore, the packet violates and the violate action (drop) is taken.

Additional References

The following sections provide references related to Traffic Policing.

Related Documents

Related Topic	Document Title
Traffic policing	"Traffic Policing" module
Modular Quality of Service Command-Line Interface (MQC)	"Applying QoS Features Using the MQC" module

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
Class-Based Quality of Service MIB • CISCO-CLASS-BASED-QOS-MIB • CISCO-CLASS-BASED-QOS-CAPABILITY-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2697	*A Single Rate Three Color Marker*

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Quality of Service Solutions Command Reference* at http://www.cisco.com/en/US/docs/ios/qos/command/reference/qos_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

- **police**

Glossary

average rate—Maximum long-term average rate of conforming traffic.

conform action—Action to take on packets with a burst size below the rate allowed by the rate limit.

DSCP—differentiated services code point

exceed action—Action to take on packets that exceed the rate limit.

excess burst size—Bytes allowed in a burst before all packets will exceed the rate limit.

normal burst size—Bytes allowed in a burst before some packets will exceed the rate limit. Larger bursts are more likely to exceed the rate limit.

QoS group—Internal QoS group ID for a packet used to determine weighted fair queuing characteristics for that packet.

policing policy—Rate limit, conform actions, and exceed actions that apply to traffic matching a certain criteria.

Versatile Interface Processor (VIP)—Interface card used by Cisco 7500 series and Cisco 7000 series with RSP7000 routers.

QoS: Percentage-Based Policing

First Published: December 4, 2006
Last Updated: February 28, 2007

The QoS: Percentage-Based Policing feature allows you to configure traffic policing and traffic shaping on the basis of a *percentage* of bandwidth available on the interface. This feature also allows you to specify the committed burst (bc) size and the excess burst (be) size (used for configuring traffic policing) in milliseconds (ms). Configuring traffic policing in this manner enables you to use the same policy map for multiple interfaces with differing amounts of bandwidth.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "Feature Information for QoS: Percentage-Based Policing" section on page 14.

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

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for QoS: Percentage-Based Policing

- For input traffic policing on a Cisco 7500 series router, verify that distributed Cisco Express Forwarding (dCEF) is enabled on the interface on which traffic policing is configured.
- For output traffic policing on a Cisco 7500 series router, ensure that the incoming traffic is dCEF-switched. Traffic policing cannot be used on the switching path unless dCEF switching is enabled.

Restrictions for QoS: Percentage-Based Policing

The **shape** (percent) command, when used in "child" (nested) policy maps, is not supported on the Cisco 7500, the Cisco 7200, or lower series routers. Therefore, the **shape** (percent) command cannot be configured for use in nested policy maps on these routers.

Information About QoS: Percentage-Based Policing

To configure QoS: Percentage-Based Policing feature, you should understand the following concepts:

Benefits for QoS: Percentage-Based Policing

Increased Flexibility and Ease-of-Use

This feature provides the ability to configure traffic policing and traffic shaping on the basis of a *percentage* of bandwidth available on an interface, and it allows you to specify burst sizes in milliseconds. Configuring traffic policing and traffic shaping in this manner enables you to use the same policy map for multiple interfaces with differing amounts of bandwidth. That is, you do not have to recalculate the bandwidth for each interface or configure a different policy map for each type of interface.

Defining Class and Policy Maps for QoS: Percentage-Based Policing

To configure the QoS: Percentage-Based Policing feature, you must define a traffic class, configure a policy map, and then attach that policy map to the appropriate interface. These three tasks can be accomplished by using the Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC).

The MQC is a command-line interface that allows you to define traffic classes, create and configure traffic policies (policy maps), and then attach these traffic policies to interfaces.

In the MQC, the **class-map** command is used to define a traffic class (which is then associated with a traffic policy). The purpose of a traffic class is to classify traffic.

The MQC consists of the following three processes:

- Defining a traffic class with the **class-map** command.
- Creating a traffic policy by associating the traffic class with one or more QoS features (using the **policy-map** command).
- Attaching the traffic policy to the interface with the **service-policy** command.

A traffic class contains three major elements: a name,a series of match commands, and, if more than one **match** command exists in the traffic class, an instruction on how to evaluate these **match** commands (that is, match-all or match-any). The traffic class is named in the **class-map** command line; for example, if you enter the **class-map cisco** command while configuring the traffic class in the CLI, the traffic class would be named "cisco".

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.

Traffic Regulation Mechanisms and Bandwidth Percentages

Cisco IOS quality of service (QoS) offers two kinds of traffic regulation mechanisms—traffic policing and traffic shaping. A traffic policer typically drops traffic that violates a specific rate. A traffic shaper typically delays excess traffic using a buffer to hold packets and shapes the flow when the data rate to a queue is higher than expected.

Traffic shaping and traffic policing can work in tandem and can be configured in a class map. Class maps organize data packets into specific categories ("classes") that can, in turn, receive a user-defined QoS treatment when used in policy maps (sometimes referred to as "service policies").

Before this feature, traffic policing and traffic shaping were configured on the basis of a user-specified amount of bandwidth available on the interface. Policy maps were then configured on the basis of that specific amount of bandwidth, meaning that separate policy maps were required for each interface.

This feature provides the ability to configure traffic policing and traffic shaping on the basis of a *percentage* of bandwidth available on the interface. Configuring traffic policing and traffic shaping in this manner enables customers to use the same policy map for multiple interfaces with differing amounts of bandwidth.

Configuring traffic policing and shaping on the basis of a percentage of bandwidth is accomplished by using the **police** (percent) and **shape** (percent) commands. For more information about these commands, see the *Cisco IOS Quality of Service Solutions Command Reference*.

Burst Size in Milliseconds Option

The purpose of the burst parameters (bc and be) is to drop packets gradually, as is done with Weighted Random Early Detection (WRED), and to avoid tail drop. Setting sufficiently high burst values helps to ensure good throughput.

This feature allows you the option of specifying the committed burst (bc) size and the extended burst (be) as milliseconds (ms) of the class bandwidth when you configure traffic policing. The number of milliseconds is used to calculate the number of bytes that will be used by the QoS: Percentage-Based Policing feature.

Specifying these burst sizes in milliseconds is accomplished by using the **bc** and **be** keywords (and their associated arguments) of the **police** (percent) and **shape** (percent) commands.

For more information about these commands, see the *Cisco IOS Quality of Service Solutions Command Reference*.

How to Configure QoS: Percentage-Based Policing

See the following sections for configuration tasks for the QoS: Percentage-Based Policing feature. Each task in the list is identified as either required or optional.

- Configuring a Class and Policy Map for Percentage-Based Policing, page 4 (required)
- Attaching the Policy Map to an Interface for Percentage-Based Policing, page 5 (required)
- Verifying the Percentage-Based Policing Configuration, page 7 (optional)

Configuring a Class and Policy Map for Percentage-Based Policing

A class map is used to organize traffic into specific categories or classes. These categories or classes of traffic are associated with a traffic policy or policy map. In turn, the policy map is used in conjunction with the class map to apply a specific QoS feature to the traffic. In this instance, the QoS feature of percentage-based policing will be applied.

To configure a class map and associate the class map with a specific policy map, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-name*
4. **class** {*class-name* | **class-default**}
5. **police cir percent** *percentage* [*burst-in-ms*] [**bc** *conform-burst-in-msec* **ms**] [**be** *peak-burst-in-msec* **ms**] [**pir percent** *percent*]
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-name* **Example:** `Router(config)# policy-map policy1`	Specifies the name ofthe policy map to becreated. Enters policy-map configuration mode. • Enter the policy map name.
Step 4	**class** {*class-name* | **class-default**} **Example:** `Router(config-pmap)# class class1`	Specifies the class so that you can configure or modify its policy. Enters policy-map class configuration mode. • Enter the class name or specify the default class (class-default).
Step 5	**police cir percent** *percentage* [*burst-in-ms*] [**bc** *conform-burst-in-msec* **ms**] [**be** *peak-burst-in-msec* **ms**] [**pir percent** *percent*] **Example:** `Router(config-pmap-c)# police cir percent 20 bc 300 ms be 400 ms pir percent 40`	Configures traffic policing on the basis of the specified bandwidth percentage and optional burst sizes. Enters policy-map class police configuration mode. • Enter the bandwidth percentage and optional burst sizes.
Step 6	**exit** **Example:** `Router(config-pmap-c-police)# exit`	Exits policy-map class police configuration mode.

Attaching the Policy Map to an Interface for Percentage-Based Policing

After a policy map is created, the next step is to attach the policy map to an interface. Policy maps can be attached to either the input or output direction of the interface.

Note Depending on the needs of your network, you may need to attach the policy map to a subinterface, an ATM PVC, a Frame Relay DLCI, or other type of interface.

To attach the policy map to an interface, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **pvc** [*name*] *vpi*/*vci* [**ilmi** | **qsaal** | **smds**]
5. **service-policy** {**input** | **output**} *policy-map-name*
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface serial4/0`	Configures an interface (or subinterface) type and enters interface configuration mode. • Enter the interface type number.
Step 4	**pvc** [*name*] *vpi*/*vci* [**ilmi** | **qsaal** | **smds**] **Example:** `Router(config-if)# pvc cisco 0/16 ilmi`	(Optional) Creates or assigns a name to an ATM PVC and specifies the encapsulation type on an ATM PVC. Enters ATM VC configuration mode. **Note** This step is required only if you are attaching the policy map to an ATM PVC. If you are not attaching the policy map to an ATM PVC, skip this step and proceed with Step 5.

	Command or Action	Purpose
Step 5	**service-policy** {**input** | **output**} *policy-map-name* **Example:** `Router(config-if)# service-policy input policy1`	Specifies the name of the policy map to be attached to the input *or* output direction of the interface. **Note** Policy maps can be configured on ingress or egress routers. They can also be attached in the input or output direction of an interface. The direction (input or output) and the router (ingress or egress) to which the policy map should be attached varies according your network configuration. When using the **service-policy** command to attach the policy map to an interface, be sure to choose the router and the interface direction that are appropriate for your network configuration. • Enter the policy map name.
Step 6	**exit** **Example:** `Router(config-if)# exit`	(Optional) Exits interface configuration mode.

Verifying the Percentage-Based Policing Configuration

To verify the configuration, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show class-map** [*class-map-name*]

   or

   show policy-map interface *interface-name*
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show class-map** [*class-map-name*] **Example:** `Router# show class-map class1`	Displays all information about a class map, including the match criterion. • Enter class map name.
	or	

	Command or Action	Purpose
	show policy-map interface *interface-name* **Example:** `Router# show policy-map interface serial4/0`	Displays the packet statistics of all classes that are configured for all service policies either on the specified interface or subinterface or on a specific PVC on the interface. • Enter the interface name.
Step 3	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

Troubleshooting Tips for Percentage-Based Policing

The commands in the "Verifying the Percentage-Based Policing Configuration" section allow you to verify that you achieved the intended configuration and that the feature is functioning correctly. If, after using the **show** commands listed above, you find that the configuration is not correct or the feature is not functioning as expected, perform these operations:

If the configuration is not the one you intended, complete the following procedures:

1. Use the **show running-config** command and analyze the output of the command.
2. If the policy map does not appear in the output of the **show running-config** command, enable the **logging console** command.
3. Attach the policy map to the interface again.

If the packets are not being matched correctly (for example, the packet counters are not incrementing correctly), complete the following procedures:

1. Run the **show policy-map** command and analyze the output of the command.
2. Run the **show running-config** command and analyze the output of the command.
3. Use the **show policy-map interface** command and analyze the output of the command. Check the the following findings:
   a. If a policy map applies queueing, and the packets are matching the correct class, but you see unexpected results, compare the number of the packets in the queue with the number of the packets matched.
   b. If the interface is congested, and only a small number of the packets are being matched, check the tuning of the transmission (tx) ring, and evaluate whether the queueing is happening on the tx ring. To do this, use the **show controllers** command, and look at the value of the tx count in the output of the command.

Configuration Examples for QoS: Percentage-Based Policing

This section provides the following configuration examples:

- Specifying Traffic Policing on the Basis of a Bandwidth Percentage: Example, page 9
- Verifying the Percentage-Based Policing Configuration, page 7

Specifying Traffic Policing on the Basis of a Bandwidth Percentage: Example

The following example configures traffic policing using a committed information rate (CIR) and a peak information rate (PIR) on the basis of a percentage of bandwidth. In this example, a CIR of 20 percent and a PIR of 40 percent have been specified. Additionally, an optional bc value and be value (300 ms and 400 ms, respectively) have been specified.

```
Router> enable
Router# configure terminal
Router(config)# policy-map policy1
Router(config-pmap)# class class1
Router(config-pmap-c)# police cir percent 20 bc 300 ms be 400 ms pir percent 40
Router(config-pmap-c-police)# exit
```

After the policy map and class maps are configured, the policy map is attached to interface as shown in the following example.

```
Router> enable
Router# configure terminal
Router(config-if)# interface serial4/0
Router(config-if)# service-policy input policy1
Router(config-if)# exit
```

Verifying the Percentage-Based Policing Configuration: Example

This section contains sample output from the **show policy-map interface** command and the **show policy-map** command. The output from these commands can be used to verify and monitor the feature configuration on your network.

The following is sample output from the **show policy-map** command. This sample output displays the contents of a policy map called "policy1." In policy 1, traffic policing on the basis of a CIR of 20 percent has been configured, and the bc and be have been specified in milliseconds. As part of the traffic policing configuration, optional conform, exceed, and violate actions have been specified.

```
Router# show policy-map policy1

  Policy Map policy1
    Class class1
     police cir percent 20 bc 300 ms pir percent 40 be 400 ms
       conform-action transmit
       exceed-action drop
       violate-action drop
```

The following is sample output from the **show policy-map interface** command. This sample displays the statistics for the serial 2/0 interface on which traffic policing has been enabled. The committed burst (bc) and excess burst (be) are specified in milliseconds (ms).

```
Router# show policy-map interface serial2/0

 Serial2/0

  Service-policy output: policy1 (1050)

    Class-map: class1 (match-all) (1051/1)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: ip precedence 0  (1052)
      police:
          cir 20 % bc 300 ms
          cir 409500 bps, bc 15360 bytes
```

```
        pir 40 % be 400 ms
        pir 819000 bps, be 40960 bytes
      conformed 0 packets, 0 bytes; actions:
        transmit
      exceeded 0 packets, 0 bytes; actions:
        drop
      violated 0 packets, 0 bytes; actions:
        drop
      conformed 0 bps, exceed 0 bps, violate 0 bps

  Class-map: class-default (match-any) (1054/0)
    0 packets, 0 bytes
    5 minute offered rate 0 bps, drop rate 0 bps
    Match: any  (1055)
      0 packets, 0 bytes
      5 minute rate 0 bps
```

In this example, the CIR and PIR are displayed in bps, and both the committed burst (bc) and excess burst (be) are displayed in bits.

The CIR, PIR bc, and be are calculated on the basis of the formulas described below.

Formula for Calculating the CIR

When calculating the CIR, the following formula is used:

- CIR percentage specified (as shown in the output of the **show policy-map** command) * bandwidth (BW) of the interface (as shown in the output of the **show interfaces** command) = total bits per second

On serial interface 2/0, the bandwidth (BW) is 2048 kbps. To see the bandwidth of the interface, use the **show interfaces** command. A sample is shown below:

```
Router# show interfaces serial2/0

Serial2/0 is administratively down, line protocol is down
  Hardware is M4T
  MTU 1500 bytes, BW 2048 Kbit, DLY 20000 usec, rely 255/255, load 1/255
```

The following values are used for calculating the CI:.

20 % * 2048 kbps = 409600 bps

Formula for Calculating the PIR

When calculating the PIR, the following formula is used:

- PIR percentage specified (as shown in the output of the **show policy-map** command) * bandwidth (BW) of the interface (as shown in the output of the **show interfaces** command) = total bits per second

On serial interface 2/0, the bandwidth (BW) is 2048 kbps. To see the bandwidth of the interface, use the **show interfaces** command. A sample is shown below:

```
Router# show interfaces serial2/0

Serial2/0 is administratively down, line protocol is down
  Hardware is M4T
  MTU 1500 bytes, BW 2048 Kbit, DLY 20000 usec, rely 255/255, load 1/255
```

The following values are used for calculating the PIR:

40 % * 2048 kbps = 819200 bps

Note Discrepancies between this total and the total shown in the output of the **show policy-map interface** command can be attributed to a rounding calculation or to differences associated with the specific interface configuration.

Formula for Calculating the Committed Burst (bc)

When calculating the bc, the following formula is used:

- The bc in milliseconds (as shown in the **show policy-map** command) * the CIR in bits per seconds = total number bytes

The following values are used for calculating the bc:

300 ms * 409600 bps = 15360 bytes

Formula for Calculating the Excess Burst (be)

When calculating the bc and the be, the following formula is used:

- The be in milliseconds (as shown in the **show policy-map** command) * the PIR in bits per seconds = total number bytes

The following values are used for calculating the be:

400 ms * 819200 bps = 40960 bytes

Additional References

The following sections provide references related to the QoS: Percentage-Based Policing feature.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
Modular QoS Command-Line Interface (CLI) (MQC), including information about attaching policy maps	"Applying QoS Features Using the MQC" module
Traffic shaping and traffic policing	"Policing and Shaping Overview" module
dCEF	"Cisco Express Forwarding Features Roadmap" module
Commands related to dCEF	*Cisco IOS Switching Command Reference*

Standard

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

MIB

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

RFCs

RFC	Title
RFC 2697	*A Single Rate Three Color Marker*
RFC 2698	*A Two Rate Three Color Marker*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Quality of Service Solutions Command Reference* at http://www.cisco.com/en/US/docs/ios/qos/command/reference/qos_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

- **police (percent)**
- **shape (percent)**
- **show policy-map**
- **show policy-map interface**

Feature Information for QoS: Percentage-Based Policing

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for QoS: Percentage-Based Policing

Feature Name	Releases	Feature Information
QoS: Percentage-Based Policing	12.2(13)T 12.0(28)S 12.2(28)SB	The QoS: Percentage-Based Policing feature allows you to configure traffic policing and traffic shaping on the basis of a *percentage* of bandwidth available on the interface. This feature also allows you to specify the committed burst (bc) size and the excess burst (be) size (used for configuring traffic policing) in milliseconds (ms). Configuring traffic policing in this manner enables you to use the same policy map for multiple interfaces with differing amounts of bandwidth. In Release 12.2(13)T, this feature was introduced. In Release 12.0(28)S, the option of specifying committed (conform) burst (bc) and excess (peak) burst (be) sizes in milliseconds was added. In Release 12.2(28)SB, this feature was integrated in Cisco IOS Release 12.2(28)SB. The following commands were introduced or modified: **police (percent)**, **shape (percent)**, **show policy-map**, **show policy-map interface**.

Regulating Packet Flow

Regulating Packet Flow Roadmap

First Published: May 2, 2005
Last Updated: June 30, 2008

This feature roadmap lists the Cisco IOS features related to traffic shaping (that is, regulating packet flow) documented in the *Cisco IOS Quality of Service Solutions Configuration Guide* and maps them to the documents in which they appear. The roadmap is organized so that you can select your release train and see the features in that release. Find the feature name you are searching for and click on the URL in the "Where Documented" column to access the document containing that feature.

Feature and Release Support

Table 1 lists traffic shaping (that is, regulating packet flow) feature support for the following Cisco IOS software release trains:

- Cisco IOS Releases 12.2T, 12.3, and 12.3T
- Cisco IOS XE Release 2.1

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 lists the most recent release of each software train first and the features in alphabetical order within the release.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Table 1 Supported Traffic Shaping-Related Features

Release	Feature Name	Feature Description	Where Documented
Cisco IOS Releases 12.2T, 12.3, and 12.3T			
12.2(8)T	Distributed Traffic Shaping	Distributed Traffic Shaping (DTS) is a legacy method for regulating the flow of packets going out an interface. Class-Based Traffic Shaping should be used instead of DTS. Class-Based Traffic Shaping can and should be used on the Cisco 7500 series router with a versatile interface processor (VIP2)-40, VIP2-50 or greater processor.	"Regulating Packet Flow on a Per-Class Basis — Using Class-Based Traffic Shaping"

Regulating Packet Flow Using Traffic Shaping

This module contains overview information about regulating the packet flow on a network. Regulating the packet flow (that is, the flow of traffic) on the network is also known as traffic shaping. Traffic shaping allows you to control the speed of traffic leaving an interface. This way, you can match the flow of the traffic to the speed of the interface receiving the packet. Cisco provides three mechanisms for regulating or shaping traffic: Class-Based Traffic Shaping, Generic Traffic Shaping (GTS), and Frame Relay Traffic Shaping (FRTS). Before configuring any of these mechanisms, it is important that you understand the overview information presented in this module.

Module History

This module was first published on May 2, 2005, and last updated on May 2, 2005.

Contents

Information About Traffic Shaping

Before configuring any of the Cisco traffic shaping mechanisms, you should understand the following concepts:

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Benefits of Shaping Traffic on a Network

The benefits of shaping traffic on the network include the following:

- It allows you to control the traffic going out an interface, matching the traffic flow to the speed of the interface.
- It ensures that traffic conforms to the policies contracted for it.
- Traffic shaping helps to ensure that a packet adheres to a stipulated contract and determines the appropriate quality of service to apply to the packet.
- It avoids bottlenecks and data-rate mismatches. For instance, central-to-remote site data speed mismatches.
- Traffic shaping prevents packet loss.

Here are some scenarios for which you would use traffic shaping:

- Control access to bandwidth when, for example, policy dictates that the rate of a given interface should not on the average exceed a certain rate even though the access rate exceeds the speed.
- Configure traffic shaping on an interface if you have a network with differing access rates. Suppose that one end of the link in a Frame Relay network runs at 256 kbps and the other end of the link runs at 128 kbps. Sending packets at 256 kbps could cause failure of the applications using the link.

  A similar, more complicated case would be a link-layer network giving indications of congestion that has differing access rates on different attached data terminal equipment (DTE); the network may be able to deliver more transit speed to a given DTE device at one time than another. (This scenario warrants that the token bucket be derived, and then its rate maintained.)
- If you offer a subrate service. In this case, traffic shaping enables you to use the router to partition your T1 or T3 links into smaller channels.
- Traffic shaping is especially important in Frame Relay networks because the switch cannot determine which packets take precedence, and therefore which packets should be dropped when congestion occurs. Moreover, it is of critical importance for real-time traffic such as Voice over Frame Relay (VoFR) that latency be bounded, thereby bounding the amount of traffic and traffic loss in the data link network at any given time by keeping the data in the router that is making the guarantees. Retaining the data in the router allows the router to prioritize traffic according to the guarantees it is making. (Packet loss can result in detrimental consequences for real-time and interactive applications.)

Cisco Traffic Shaping Mechanisms

Cisco provides three traffic shaping mechanisms: Class-Based Traffic Shaping, GTS, and FRTS.

All three mechanisms are similar in implementation, though their command-line interfaces (CLIs) differ somewhat and they use different types of queues to contain and shape traffic that is deferred. In particular, the underlying code that determines whether a packet is sent or delayed is common to all three mechanisms, and all three mechanism use a token bucket metaphor (see the "Token Bucket and Traffic Shaping" section on page 3).

Table 1 lists the differences between traffic shaping mechanisms.

Table 1 Differences Between Traffic Shaping Mechanisms

Traffic Shaping Mechanism	Class-Based	GTS	FRTS
Command-Line Interface	• Applies configuration on a per-class basis	• Applies configuration on a per interface or subinterface basis • **traffic group** command supported	• Classes of parameters • Applies configuration to all virtual circuits (VCs) on an interface through inheritance mechanism • No **traffic group** group
Queues Supported	• Class-based WFQ (CBWFQ)	• Weighted Fair Queueing (WFQ) per interface or subinterface	• WFQ, strict priority queue with WFQ, custom queue (CQ), priority queue (PQ), first-in first-out (FIFO) per VC
For More Details, See The. . .	"Regulating Packet Flow on a Per-Class Basis — Using Class-Based Traffic Shaping" module	"Regulating Packet Flow on a Per-Interface Basis — Using Generic Traffic Shaping" module	"MQC-Based Frame Relay Traffic Shaping" module

Token Bucket and Traffic Shaping

Traffic shaping uses a token bucket metaphor to shape traffic. 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
```

Here are some definitions of these terms:

- Mean rate—Also called the committed information rate (CIR), it specifies how much data can be sent or forwarded per unit time on average.
- Burst size—Also called the committed burst (Bc) size, it specifies in bits (or bytes) per burst how much traffic can be sent within a given unit of time to not create scheduling concerns. (For a traffic shaper, it specifies bits per burst.)
- Time interval—Also called the measurement interval, it specifies the time quantum in seconds per burst.

By definition, over any integral multiple of the interval, the bit rate of the interface will not exceed the mean rate. The bit rate, however, may be arbitrarily fast within the interval.

A token bucket is used to manage a device that regulates the data in a flow. For example, the regulator might be a traffic shaper. A token bucket itself has no discard or priority policy. Rather, a token bucket discards tokens and leaves to the flow the problem of managing its transmission queue if the flow overdrives the regulator.

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

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

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

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

Traffic Shaping and Rate of Transfer

Traffic shaping limits the rate of transmission of data. You can limit the data transfer to one of the following:

- A specific configured rate
- A derived rate based on the level of congestion

As mentioned, the rate of transfer depends on these three components that constitute the token bucket: burst size, mean rate, time (measurement) interval. The mean rate is equal to the burst size divided by the interval.

When traffic shaping is enabled, the bit rate of the interface will not exceed the mean rate over any integral multiple of the interval. In other words, during every interval, a maximum of burst size can be sent. Within the interval, however, the bit rate may be faster than the mean rate at any given time.

One additional variable applies to traffic shaping: excess burst (Be) size. The Be size corresponds to the number of noncommitted bits—those outside the CIR—that are still accepted by the Frame Relay switch but marked as discard eligible (DE).

In other words, the Be size allows more than the burst size to be sent during a time interval in certain situations. The switch will allow the packets belonging to the excess burst to go through but it will mark them by setting the DE bit. Whether the packets are sent depends on how the switch is configured.

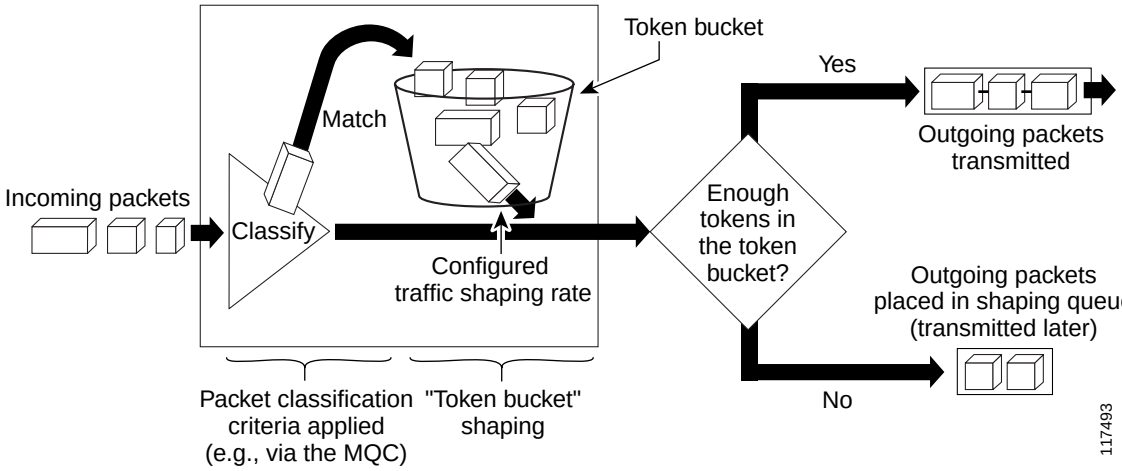
When the Be size equals 0, the interface sends no more than the burst size every interval, achieving an average rate no higher than the mean rate. However, when the Be size is greater than 0, the interface can send as many as Bc plus Be bits in a burst, if in a previous time period the maximum amount was not sent. Whenever less than the burst size is sent during an interval, the remaining number of bits, up to the Be size, can be used to send more than the burst size in a later interval.

How Traffic Shaping Regulates Traffic

As mentioned previously, Cisco provides three mechanisms for shaping traffic: Class-Based Traffic Shaping, GTS, and FRTS. All three mechanisms are similar in implementation, though their CLIs differ somewhat and they use different types of queues to contain and shape traffic that is deferred.

Figure 1 illustrates how a traffic shaping mechanism regulates traffic.

Figure 1 How a Traffic Shaping Mechanism Regulates Traffic



In Figure 1, incoming packets arrive at an interface. The packets are classified using a "classification engine," such as an access control list (ACL) or the Modular Quality of Service Command-Line Interface (MQC). If the packet matches the specified classification, the traffic shaping mechanism continues. Otherwise, no further action is taken.

Packets matching the specified criteria are placed in the token bucket. The maximum size of the token bucket is the Bc size plus the Be size. The token bucket is filled at a constant rate of Bc worth of tokens at every Tc. This is the configured traffic shaping rate.

If the traffic shaping mechanism is active (that is, packets exceeding the configured traffic shaping rate already exist in a transmission queue), at every Tc, the traffic shaper checks to see if the transmission queue contains enough packets to send (that is, up to either Bc (or Bc plus Be) worth of traffic).

If the traffic shaper is not active (that is, there are no packets exceeding the configured traffic shaping rate in the transmission queue), the traffic shaper checks the number of tokens in the token bucket. One of the following occurs:

- If there are enough tokens in the token bucket, the packet is sent (transmitted).
- If there are not enough tokens in the token bucket, the packet is placed in a shaping queue for transmission at a later time.

Traffic Shaping versus Traffic Policing

Although traffic shaping and traffic policing can be implemented together on the same network, there are distinct differences between them, as shown in Table 2.

Table 2 Differences Between Traffic Shaping and Traffic Policing

	Traffic Shaping	Traffic Policing
Triggering Event	• Occurs automatically at regular intervals (Tc). or Occurs whenever a packet arrives at an interface.	• Occurs whenever a packet arrives at an interface.
What it Does	• Classifies packets. • If packet does not meet match criteria, no further action is taken. • Packets meeting match criteria are sent (if there are enough tokens in the token bucket) or Packets are placed in a queue for transmission later. • If the number of packets in the queue exceed the queue limit, the packets are dropped.	• Classifies packets. • If packet does not meet match criteria, no further action is taken. • Packets meeting match criteria and conforming to, exceeding, or violating a specified rate, receive the configured policing action (for example, drop, send, mark then send). • Packets are not placed in queue for transmission later.

For more information about traffic policing, see the following documents:

- "Traffic Policing" module
- "Two-Rate Policer" module
- "Control Plane Policing" module
- "Class-Based Policing" module
- "QoS: Percentage-Based Policing" module
- "Policer Enhancement — Multiple Actions" module
- "QoS: Color-Aware Policer" module

Note The above list of documents related to traffic policing is not all-inclusive. Traffic policing-related features and modules vary by IOS release and platform.

Where to Go Next

To configure Class-Based Traffic Shaping, see the "Regulating Packet Flow on a Per-Class Basis — Using Class-Based Traffic Shaping" module.

To configure GTS, see the "Regulating Packet Flow on a Per-Interface Basis — Using Generic Traffic Shaping" module.

To configure FRTS, see the "MQC-Based Frame Relay Traffic Shaping" module.

Additional References

The following sections provide additional references about traffic shaping.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
Packet classification	"Classifying Network Traffic" module
MQC, policy maps, class maps, and hierarchical policy maps	"Applying QoS Features Using the MQC" module
WFQ, CBWFQ, PQ, CQ, FIFO and other queueing mechanisms	"Congestion Management Overview" module
Class-Based Traffic Shaping	"Regulating Packet Flow on a Per-Class Basis — Using Class-Based Traffic Shaping" module
GTS	"Regulating Packet Flow on a Per-Interface Basis — Using Generic Traffic Shaping" module
FRTS	"MQC-Based Frame Relay Traffic Shaping" module

Standards

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

MIBs

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

RFCs

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

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Regulating Packet Flow on a Per-Class Basis—Using Class-Based Traffic Shaping

First Published: February 25, 2002
Last Updated: June 30, 2008

Packet flow on a network can be regulated using a traffic shaping mechanism. One such traffic shaping mechanism is a Cisco feature called Class-Based Traffic Shaping. Class-Based Traffic Shaping allows you to regulate the flow of packets (on a per-traffic-class basis) going out an interface, matching the packet flow to the speed of the interface. This module describes the concepts and tasks related to configuring Class-Based Traffic Shaping.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported use the "Feature Information for Class-Based Traffic Shaping" section on page 11.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Configuring Class-Based Traffic Shaping

Knowledge

Be familiar with the concepts in the "Regulating Packet Flow Using Traffic Shaping" module.

Platform Support

Use Feature Navigator to determine if the platform in use supports Class-Based Traffic Shaping. Access Cisco Feature Navigator at http://www.cisco.com/go/fn.

Enable dCEF

Distributed Cisco Express Forwarding (dCEF) must be enabled if the customer is using a Versatile Interface Processor (VIP) on the router.

Create Policy Map and Class

A policy map and a class map must be created first using the Modular Quality of Service (QoS) Command-Line Interface (MQC). For information about using the MQC, see the "Applying QoS Features Using the MQC" module.

Restrictions for Configuring Class-Based Traffic Shaping

Adaptive Traffic Shaping

Adaptive traffic shaping for Frame Relay networks is supported for Frame Relay networks only.

Outbound Traffic Only

Class-Based Traffic Shaping applies to outbound traffic only.

Unsupported Commands

Class-Based Traffic Shaping does not support the following commands:

- **traffic-shape adaptive**
- **traffic shape fecn-adaptive**
- **traffic-shape group**
- **traffic-shape rate**

Information About Class-Based Traffic Shaping

To configure Class-Based Traffic Shaping, you should understand the following concepts:

Class-Based Traffic Shaping Functionality

Class-Based Traffic Shaping is a traffic shaping mechanism (also known as a "traffic shaper"). A traffic shaper typically delays excess traffic using a buffer, or queueing mechanism, to hold packets and shape the flow when the data rate of the source is higherthan expected. It holds and shapes traffic to aparticular bit rate by using the token bucket mechanism. For more information about token buckets and traffic shaping, see the "Regulating Packet Flow Using Traffic Shaping" module.

Class-Based Traffic Shaping is the Cisco-recommended traffic shaping mechanism.

Note Class-Based Traffic Shaping should be used instead of what was previously referred to as Distributed Traffic Shaping (DTS). Class-Based Traffic Shaping can and should be used on the Cisco 7500 series router with a VIP2-40, VIP2-50, or greater processor.

Using the Class-Based Traffic Shaping, you can perform the following tasks:

- Configure traffic shaping on a per-traffic-class basis. It allows you to fine-tune traffic shaping for one or more classes and it allows you to configure traffic shaping on a more granular level.
- Specify average rate or peak rate traffic shaping. Specifying peak rate shaping allows you to make better use of available bandwidth by allowing more data than the configured traffic shaping rate to be sent if the bandwidth is available.
- Configure traffic shaping in a hierarchical policy map structure. That is, traffic shaping is configured in a primary-level (parent) policy map and other QoS features (for instance, CBWFQ and traffic policing) can be configured in the secondary-level (child) policy maps. For more information, see the "Hierarchical Policy Map Structure of Class-Based Traffic Shaping" section on page 3.

Benefits of Class-Based Traffic Shaping

All of the benefits associated with traffic shaping also apply to Class-Based Traffic Shaping, but on a more granular level. For information about the benefits of traffic shaping, see the "Regulating Packet Flow Using Traffic Shaping" module.

Hierarchical Policy Map Structure of Class-Based Traffic Shaping

With the Class-Based Traffic Shaping mechanism, traffic shaping can be configured in a hierarchical policy map structure; that is, traffic shaping is enabled in a primary-level (parent) policy map and other QoS features used with traffic shaping, such as CBWFQ and traffic policing, can be enabled in a secondary-level (child) policy map.

Traffic shaping is enabled by using the **shape** command (and specifying a rate) in a policy map. When traffic shaping is enabled, one the following actions occur:

- Packets exceeding the specified rate are placed in a queue using an appropriate queueing mechanism.
- Packets conforming to the specified rate are transmitted.

When packets are placed in a queue, the default queueing mechanism used is weighted fair queueing (WFQ). However, with Class-Based Traffic Shaping, class-based WFQ (CBWFQ) can be configured as an alternative queueing mechanism.

CBWFQ allows you to fine-tune the way traffic is placed in a queue. For instance, you can specify that all voice traffic be placed in a high-priority queue and all traffic from a specific class be placed in a lower-priority queue.

If you want to use CBWFQ with the Class-Based Traffic Shaping mechanism, the following conditions must be met:

- A secondary-level (child) policy map *must* be created. This secondary-level (child) policy map is then used to configure CBWFQ by enabling the **bandwidth** command.
- Traffic shaping *must* be configured in the primary-level (parent) policy map.

Note CBWFQ is supported in both the primary-level (parent) policy map and the secondary-level (child) policy map. However, to use CBWFQ at the secondary-level (child) policy map, traffic shaping *must* be configured in the primary-level (parent) policy map.

The following sample configuration illustrates how the Class-Based Traffic Shaping mechanism is configured in a hierarchical policy map structure:

```
enable
configure terminal
 policy-map policy_parent        ! This is the primary-level policy map.
  class class-default
   shape average 1000000         ! This enables traffic shaping.
   service-policy policy_child   ! This associates the policy maps.
```

Traffic shaping must be configured in the primary-level (parent) policy map. With this configuration, WFQ is used as the default queueing mechanism for placing all the traffic in a queue.

In the following secondary-level (child) policy map, the alternative queueing mechanism CBWFQ is configured:

```
enable
configure terminal
 policy-map policy_child     ! This is the secondary-level policy map.
  class class-default
   bandwidth percent 50      ! This enables CBWFQ.
```

How to Configure Class-Based Traffic Shaping

This section contains the following procedures:

- Configuring Class-Based Traffic Shaping in a Primary-Level (Parent) Policy Map, page 4 (required)
- Configuring the Secondary-Level (Child) Policy Map, page 6 (optional)

Configuring Class-Based Traffic Shaping in a Primary-Level (Parent) Policy Map

Traffic shaping is configured in a policy map. Policy maps determine the specific quality of service (QoS) feature that will be applied to traffic on a network. In this module, the QoS feature being applied is traffic shaping.

Traffic shaping is configured in the primary-level (parent) policy map in the hierarchy.

Note Traffic shaping is supported in the primary-level (parent) policy map *only*.

Prerequisites

Before configuring traffic shaping, you must use the MQC to create a policy map and a class map. For information about using the MQC to create a policy map and a class map, see the "Applying QoS Features Using the MQC" module.

To configure Class-Based Traffic Shaping (after first creating a policy map and class map), complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** {*class-name* | **class-default**}
5. **shape** [**average** | **peak**] *mean-rate* [[*burst-size*] [*excess-burst-size*]]
6. **service-policy** *policy-map-name*
7. **end**
8. **show policy-map**
9. **show policy-map interface** *type number*
10. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map-name* **Example:** `Router(config)# policy-map policy_parent`	Specifies the name of the policy map created earlier and enters policy-map configuration mode. See the "Prerequisites" section on page 5 for more information. • Enter the policy map name.
Step 4	**class** {*class-name* | **class-default**} **Example:** `Router(config-pmap)# class class-default`	Specifies the name of the class whose policy you want to create and enters policy-map class configuration mode. • Enter the name of the class or enter the **class-default** keyword.

	Command or Action	Purpose
Step 5	**shape** [**average** | **peak**] *mean-rate* [*burst-size*] [*excess-burst-size*] **Example:** `Router(config-pmap-c)# shape average 1000000`	Shapes traffic according to the keyword and rate specified. • Enter the keyword and rate.
Step 6	**service-policy** *policy-map-name* **Example:** `Router(config-pmap-c)# service-policy policy_child`	Uses a service policy as a QoS policy within a policy map (called a hierarchical service policy). • Enter the policy map name.
Step 7	**end** **Example:** `Router(config-pmap-c)# end`	Returns to privileged EXEC mode.
Step 8	**show policy-map** **Example:** `Router# show policy-map`	(Optional) Displays all configured policy maps.
Step 9	**show policy-map interface** *type number* **Example:** `Router# show policy-map interface serial4/0`	(Optional) Displays the packet statistics of all classes that are configured for all service policies either on the specified interface or subinterface or on a specific PVC on the interface. • Enter the interface type and number.
Step 10	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

What to Do Next

So far, you have configured Class-Based Traffic Shaping in a primary-level (parent) policy map. To configure a secondary-level (child) policy map in the hierarchical policy map structure (an optional task), proceed with the instructions in "Configuring the Secondary-Level (Child) Policy Map" section on page 6.

Configuring the Secondary-Level (Child) Policy Map

In the secondary-level (child) policy map, additional QoS features used with traffic shaping (for example, CBWFQ and traffic policing) are typically configured. For Class-Based Traffic Shaping, the only two QoS features supported at the secondary-level (child) policy map are CBWFQ and traffic policing.

Note CBWFQ is supported in both the primary-level (parent) policy map and the secondary-level (child) policy map. However, to use CBWFQ in the secondary-level (child) policy map, traffic shaping *must* be configured in the primary-level (parent) policy map. For more information about CBWFQ in a secondary-level (child) policy map, see the "Hierarchical Policy Map Structure of Class-Based Traffic Shaping" section on page 3.

To configure a QoS feature (such as CBWFQ and traffic policing) in a secondary-level (child) policy map, complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** {*class-name* | **class-default**}
5. **bandwidth** {*bandwidth-kbps* | **remaining percent** *percentage* | **percent** *percentage*}
6. **end**
7. **show policy-map**
8. **show policy-map interface** *type number*
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`policy-map` *policy-map-name* **Example:** `Router(config)# policy-map policy1`	Specifies the name of the policy map created earlier and enters policy-map configuration mode. See the "Prerequisites" section on page 5 for more information. • Enter the policy map name.
Step 4	`class` {*class-name* | `class-default`} **Example:** `Router(config-pmap)# class class-default`	Specifies the name of the class whose policy you want to create and enters policy-map class configuration mode. • Enter the name of the class or enter the **class-default** keyword.

	Command or Action	Purpose
Step 5	**bandwidth** {*bandwidth-kbps* | **remaining percent** *percentage* | **percent** *percentage*} **Example:** `Router(config-pmap-c)# bandwidth percent 50`	Specifies or modifies the bandwidth allocated for a class belonging to a policy map. • Enter the amount of bandwidth as a number of kbps, a relative percentage of bandwidth, or an absolute amount of bandwidth. **Note** The **bandwidth** command used here is only an example of a QoS feature than can be configured. The **bandwidth** command configures CBWFQ. You could also use the **police** command to configure traffic policing.
Step 6	**end** **Example:** `Router(config-pmap-c)# end`	Returns to privileged EXEC mode.
Step 7	**show policy-map** **Example:** `Router# show policy-map`	(Optional) Displays all configured policy maps.
Step 8	**show policy-map interface** *type number* **Example:** `Router# show policy-map interface serial4/0`	(Optional) Displays the packet statistics of all classes that are configured for all service policies either on the specified interface or subinterface or on a specific PVC on the interface. • Enter the interface type and number.
Step 9	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

Configuration Examples for Class-Based Traffic Shaping

This section contains the following examples:

• Class-Based Traffic Shaping Configuration: Example, page 8

Class-Based Traffic Shaping Configuration: Example

The following is an example of Class-Based Traffic Shaping configured in a hierarchical policy map structure. In this example, two policy maps have been created; the primary-level (parent) policy map called “policy_parent,” and a secondary-level (child) policy map called “policy_child.” Traffic shaping is configured in the policy_parent policy map, and CBWFQ has been configured in the policy_child policy map.

The **service-policy** command associates the two policy maps in the hierarchical policy map structure.

```
enable
configure terminal
 policy-map policy_parent
```

```
 class class-default
  shape average 1000000         ! This enables traffic shaping.
  service-policy policy_child   ! This associates the policy maps.
  exit
 exit
policy-map policy_child
 class class-default
  bandwidth percent 50   ! This enables CBWFQ.
  end
```

Where to Go Next

To configure Generic Traffic Shaping (GTS), see the "Regulating Packet Flow on a Per-Interface Basis—Using Generic Traffic Shaping" module.

To configure Frame Relay Traffic Shaping (FRTS), see the "MQC-Based Frame Relay Traffic Shaping" module.

Additional References

The following sections provide references related to configuring Class-Based Traffic Shaping.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
Packet classification	"Classifying Network Traffic" module
MQC, policy maps, class maps, and hierarchical policy maps	"Applying QoS Features Using the MQC" module
CBWFQ and other queueing mechanisms	"Configuring Weighted Fair Queueing" module
dCEF	"Cisco Express Forwarding Features Roadmap" module
Overview information about using traffic shaping to regulate packet flow on a network	"Regulating Packet Flow Using Traffic Shaping" module
GTS	"Regulating Packet Flow on a Per-Interface Basis—Using Generic Traffic Shaping" module
FRTS	"MQC-Based Frame Relay Traffic Shaping" module
Information on a feature in this technology that is not documented here	"Regulating Packet Flow Roadmap" module

Standards

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

MIBs

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

RFCs

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

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Class-Based Traffic Shaping

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(1) or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the "Regulating Packet Flow Roadmap" module.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Class-Based Traffic Shaping

Feature Name	Software Releases	Feature Configuration Information
Distributed Traffic Shaping	12.2(8)T	Distributed Traffic Shaping (DTS) is a legacy method for regulating the flow of packets going out an interface. Class-Based Traffic Shaping should be used instead of (DTS). The following sections provide information about Class-Based Traffic Shaping: • Information About Class-Based Traffic Shaping, page 2 • How to Configure Class-Based Traffic Shaping, page 4

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

Regulating Packet Flow on a Per-Interface Basis — Using Generic Traffic Shaping

Packet flow on a network can be regulated using a traffic shaping mechanism. One such traffic shaping mechanism is a Cisco feature called Generic Traffic Shaping (GTS). Generic Traffic Shaping allows you to regulate the flow of packets going out an interface or subinterface, matching the packet flow to the speed of the interface. This module describes the concepts and tasks related to configuring Generic Traffic Shaping.

Module History

This module was first published on May 2, 2005, and last updated on May 2, 2005.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Generic Traffic Shaping" section on page 11.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Configuring Generic Traffic Shaping

Knowledge

- Be familiar with the concepts in the "Regulating Packet Flow Using Traffic Shaping" module.

Platform Support

- Use Feature Navigator to determine if the platform in use supports GTS. Access Cisco Feature Navigator at http://www.cisco.com/go/fn.

Restrictions for Configuring Generic Traffic Shaping

- GTS is not supported on the following interfaces:
  - Multilink PPP (MLP) interfaces
  - Integrated Services Digital Networks (ISDNs), dialer interfaces, or generic routing encapsulation (GRE) tunnel interfaces on the Cisco 7500 series router
- GTS is not supported with flow switching.

Information About Configuring Generic Traffic Shaping

To configure GTS, you should understand the following concepts:

Generic Traffic Shaping Functionality

GTS is a traffic shaping mechanism (also known as a "traffic shaper"). A traffic shaper typically delays excess traffic using a buffer, or queueing mechanism, to hold packets and shape the flow when the data rate of the source is higher than expected. It holds and shapes traffic to a particular bit rate by using the token bucket mechanism. See the "Regulating Packet Flow Using Traffic Shaping" module.

Note GTS is similar to Class-Based Traffic Shaping. Although Class-Based Traffic Shaping is the Cisco-recommended mechanism, GTS is still supported.

GTS supports traffic shaping on most media and encapsulation types on the router.

GTS works with a variety of Layer 2 technologies, including Frame Relay, ATM, Switched Multimegabit Data Service (SMDS), and Ethernet.

GTS performs the following tasks:

- Applies traffic shaping on a per-interface basis and uses access control lists (ACLs) to select the traffic to shape.
- On a Frame Relay subinterface, dynamically adapts to available bandwidth by integrating backward explicit congestion notification (BECN) signals, or shapes to a specified rate. This is known as adaptive GTS.
- On an ATM/ATM Interface Processor (AIP) interface, responds to the Resource Reservation Protocol (RSVP) feature signalled over statically configured ATM permanent virtual circuits (PVCs).

Adaptive Generic Traffic Shaping on Frame Relay Networks

If adaptive GTS is configured on a Frame Relay network using the **traffic-shape rate** command, you can also use the **traffic-shape adaptive** command to specify the minimum bit rate to which the traffic is shaped.

With adaptive GTS, the router uses backward explicit congestion notifications (BECNs) to estimate the available bandwidth and adjust the transmission rate accordingly. The actual maximum transmission rate will be between the rate specified in the **traffic-shape adaptive** command and the rate specified in the **traffic-shape rate** command.

Configure these two commands on both ends of the network link, enabling the router at the high-speed end to detect and adapt to congestion even when traffic is flowing primarily in one direction.

For more information about configuring adaptive GTS, see the "Configuring Adaptive Generic Traffic Shaping for Frame Relay Networks" section on page 7.

Benefits of Generic Traffic Shaping

All of the benefits associated with traffic shaping also apply to GTS. For information about the benefits of traffic shaping, see the "Regulating Packet Flow Using Traffic Shaping" module.

How to Configure Generic Traffic Shaping

This section contains the following procedures. While all three procedures are listed as optional, you must choose either the first or the second procedure.

Configuring Generic Traffic Shaping on an Interface

To configure GTS on an interface, complete the following steps.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **interface** *type number*
4. **traffic-shape rate** *bit-rate* [*burst-size*] [*excess-burst-size*] [*buffer-limit*]
5. **end**
6. **show traffic-shape** [*interface-type interface-number*]
7. **show traffic-shape statistics** [*interface-type interface-number*]
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface s4/0`	Configures an interface (or subinterface) type and enters interface configuration mode. • Enter the interface type number.
Step 4	**traffic-shape rate** *bit-rate* [*burst-size*] [*excess-burst-size*] [*buffer-limit*] **Example:** `Router(config-if)# traffic-shape rate 128000`	Enables traffic shaping for outbound traffic on an interface based on the bit rate specified. • Enter the bit rate.
Step 5	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.
Step 6	**show traffic-shape** [*interface-type interface-number*] **Example:** `Router# show traffic-shape serial4/0`	(Optional) Displays the current traffic-shaping configuration.

	Command or Action	Purpose
	show traffic-shape statistics [*interface-type interface-number*] **Example:** `Router# show traffic-shape statistics serial4/0`	(Optional) Displays the current traffic-shaping statistics.
Step 7	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

Configuring Generic Traffic Shaping Using an Access Control List

To configure GTS for outbound traffic using an access control list (ACL), complete the following steps.

Access Control List Functionality

Access control lists filter network traffic by controlling whether routed packets are forwarded or blocked at the router interface. When configured with GTS, the router examines each packet to determine how to shape the traffic on the basis of the criteria you specified for the access control list.

Access control list criteria could be the source address of the traffic, the destination address of the traffic, the upper-layer protocol, or other information. Note that sophisticated users can sometimes successfully evade or fool basic access control lists because no authentication is required.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*]
4. **interface** *type number*
5. **traffic-shape group** *access-list bit-rate* [*burst-size* [*excess-burst-size*]]
6. **end**
7. **show traffic-shape** [*interface-type interface-number*]
8. **show traffic-shape statistics** [*interface-type interface-number*]
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] **Example:** `Router(config)# access-list 1 permit 192.5.34.0 0.0.0.255`	Shapes traffic according to specified access list. • Enter the access list number, one of the required keywords, and the source information.
Step 4	**interface** *type number* **Example:** `Router(config)# interface s4/0`	Configures an interface (or subinterface) type and enters interface configuration mode. • Enter the interface type number.
Step 5	**traffic-shape group** *access-list bit-rate* [*burst-size* [*excess-burst-size*]] **Example:** `Router(config-if)# traffic-shape group 101 128000`	Enables traffic shaping based on a specific access list for outbound traffic on an interface. • Enter the access list number and the bit rate.
Step 6	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.
Step 7	**show traffic-shape** [*interface-type interface-number*] **Example:** `Router# show traffic-shape serial4/0`	(Optional) Displays the current traffic-shaping configuration.
Step 8	**show traffic-shape statistics** [*interface-type interface-number*] **Example:** `Router# show traffic-shape statistics serial4/0`	(Optional) Displays the current traffic-shaping statistics.
Step 9	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

Note Repeat the above procedure for each additional type of traffic you want to shape.

Configuring Adaptive Generic Traffic Shaping for Frame Relay Networks

To configure adaptive GTS for Frame Relay networks, complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **traffic-shape rate** *bit-rate* [*burst-size*] [*excess-burst-size*] [*buffer-limit*]
5. **traffic-shape adaptive** bit-rate
6. **traffic-shape fecn-adapt**
7. **end**
8. **show traffic-shape** [*interface-type interface-number*]
9. **show traffic-shape statistics** [*interface-type interface-number*]
10. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface s4/0`	Configures an interface (or subinterface) type and enters interface configuration mode. • Enter the interface type number.
Step 4	**traffic-shape rate** *bit-rate* [*burst-size*] [*excess-burst-size*] [*buffer-limit*] **Example:** `Router(config-if)# traffic-shape rate 128000`	Enables traffic shaping for outbound traffic on an interface based on the bit rate specified. • Enter the bit rate.
Step 5	**traffic-shape adaptive** bit-rate **Example:** `Router(config-if)# traffic-shape adaptive 64000`	Configures a Frame Relay subinterface to estimate the available bandwidth when BECNs are received. • Enter the bit rate.

	Command or Action	Purpose
Step 6	**traffic-shape fecn-adapt** **Example:** `Router(config-if)# traffic-shape fecn-adapt`	Configures reflection of forward explicit congestion notifications (FECNs) as BECNs.
Step 7	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.
Step 8	**show traffic-shape** [*interface-type interface-number*] **Example:** `Router# show traffic-shape serial4/0`	(Optional) Displays the current traffic-shaping configuration.
Step 9	**show traffic-shape statistics** [*interface-type interface-number*] **Example:** `Router# show traffic-shape statistics serial4/0`	(Optional) Displays the current traffic-shaping statistics.
Step 10	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

Configuration Examples for Generic Traffic Shaping

This section contains the following examples:

Generic Traffic Shaping on an Interface Configuration: Example

The following is an example of GTS configured on serial interface s4/0:

```
enable
configure terminal
 interface s4/0
  traffic-shape rate 128000
  end
```

Generic Traffic Shaping Using an Access Control List Configuration: Example

The following is an example of GTS configured using an ACL. In this example, GTS is configured for the outbound traffic on ACL 1.

```
enable
configure terminal
 access-list 1 permit 192.5.34.0 0.0.0.255
 interface s4/0
  traffic-shape group 101 128000
  end
```

Adaptive Generic Traffic Shaping for a Frame Relay Network Configuration: Example

The following is an example of adaptive GTS configured on Frame Relay network. In this example, adaptive GTS is configured using the **traffic-shape rate** command. The **traffic-shape adaptive** command specifies the minimum bit rate to which the traffic is shaped. The actual maximum transmission rate will be between the rate specified in the **traffic-shape adaptive** command and the rate specified in the **traffic-shape rate** command.

```
enable
configure terminal
 interface s4/0
  traffic-shape rate 128000
  traffic-shape adaptive 64000
  traffic-shape fecn-adapt
  end
```

Where to Go Next

To configure Class-Based Traffic Shaping, see the "Regulating Packet Flow on a Per-Class Basis — Using Class-Based Traffic Shaping" module.

To configure Frame Relay Traffic Shaping (FRTS), see the "MQC-Based Frame Relay Traffic Shaping" module.

Additional References

The following sections provide additional references related to configuring GTS.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
Overview information about using traffic shaping to regulate packet flow on a network	"Regulating Packet Flow Using Traffic Shaping" module
Class-Based Traffic Shaping	"Regulating Packet Flow on a Per-Class Basis — Using Class-Based Traffic Shaping" module
FRTS	"MQC-Based Frame Relay Traffic Shaping" module

Standards

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

MIBs

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

RFCs

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

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Generic Traffic Shaping

Table 1 lists the release history for this feature

Not all commands may be available in your Cisco IOS software release. For details on when support for specific commands was introduced, see the command reference documents.

For information on a feature in this technology that is not documented here, see the "Regulating Packet Flow Roadmap."

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Generic Traffic Shaping

Feature Name	Software Releases	Feature Configuration Information
This table is intentionally left blank because no features were introduced or modified in Cisco IOS Release 12.2(1) or later. This table will be updated when feature information is added to this module.	—	—

Signalling

Signalling Overview

In the most general sense, QoS signalling is a form of network communication that allows an end station or network node to communicate with, or signal, its neighbors to request special handling of certain traffic. QoS signalling is useful for coordinating the traffic handling techniques provided by other QoS features. It plays a key role in configuring successful overall end-to-end QoS service across your network.

True end-to-end QoS requires that every element in the network path—switch, router, firewall, host, client, and so on—deliver its part of QoS, and that all of these entities be coordinated with QoS signalling.

Many viable QoS signalling solutions provide QoS at some places in the infrastructure; however, they often have limited scope across the network. To achieve end-to-end QoS, signalling must span the entire network.

Cisco IOS QoS software takes advantage of IP to meet the challenge of finding a robust QoS signalling solution that can operate over heterogeneous network infrastructures. It overlays Layer 2 technology-specific QoS signalling solutions with Layer 3 IP QoS signalling methods of the Resource Reservation Protocol (RSVP) and IP Precedence features.

An IP network can achieve end-to-end QoS, for example, by using part of the IP packet header to request special handling of priority or time-sensitive traffic. Given the ubiquity of IP, QoS signalling that takes advantage of IP provides powerful end-to-end signalling. Both RSVP and IP Precedence fit this category.

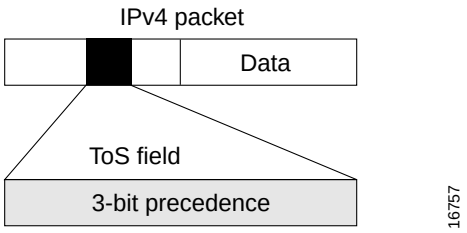
Either in-band (IP Precedence, 802.1p) or out-of-band (RSVP) signalling is used to indicate that a particular QoS is desired for a particular traffic classification. IP Precedence signals for differentiated QoS, and RSVP for guaranteed QoS.

IP Precedence

As shown in Figure 1, the IP Precedence feature utilizes the three precedence bits in the type of service (ToS) field of the IP version 4 (IPv4) header to specify class of service for each packet. You can partition traffic in up to six classes of service using IP precedence. The queueing technologies throughout the network can then use this signal to provide the appropriate expedited handling.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Figure 1 IP Precedence ToS Field



You can use features such as policy-based routing (PBR) and committed access rate (CAR) to set precedence based on extended access list classification. Use of these features allows considerable flexibility of precedence assignment, including assignment by application or user, or by destination or source subnet. Typically, you deploy these features as close to the edge of the network or the administrative domain as possible, so that each subsequent network element can provide service based on the determined policy. IP precedence can also be set in the host or the network client; however, IP precedence can be overridden by policy within the network.

IP precedence enables service classes to be established using existing network queueing mechanisms, such as weighted fair queueing (WFQ) and Weighted Random Early Detection (WRED), with no changes to existing applications and with no complicated network requirements.

Resource Reservation Protocol

RSVP is the first significant industry-standard protocol for dynamically setting up end-to-end QoS across a heterogeneous network. RSVP, which runs over IP, allows an application to dynamically reserve network bandwidth. Using RSVP, applications can request a certain level of QoS for a data flow across a network.

The Cisco IOS QoS implementation allows RSVP to be initiated within the network using configured proxy RSVP. Using this capability, you can take advantage of the benefits of RSVP in the network even for non-RSVP enabled applications and hosts. RSVP is the only standard signalling protocol designed to guarantee network bandwidth from end-to-end for IP networks.

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. This modularity does not prevent RSVP from using other routing services. RSVP provides transparent operation through router nodes that do not support RSVP.

RSVP works in conjunction with, not in place of, current queueing mechanisms. RSVP requests the particular QoS, but it is up to the particular interface queueing mechanism, such as WFQ or WRED, to implement the reservation.

You can use RSVP to make two types of dynamic reservations: controlled load and guaranteed rate services, both of which are briefly described in the chapter "Quality of Service Overview" in this book.

A primary feature of RSVP is its scalability. RSVP scales well using the inherent scalability of multicast. RSVP scales to very large multicast groups because it uses receiver-oriented reservation requests that merge as they progress up the multicast tree. Although RSVP is designed specifically for multicast applications, it may also make unicast reservations. However, it does not scale as well with a large number of unicast reservations.

RSVP is an important QoS feature, but it does not solve all problems addressed by QoS, and it imposes a few hindrances, such as the time required to set up end-to-end reservation.

How It Works

Hosts and routers use RSVP to deliver QoS requests to the routers along the paths of the data stream and to maintain router and host state to provide the requested service, usually bandwidth and latency. RSVP uses a mean data rate—the largest amount of data the router will keep in the queue—and minimum QoS (that is, guarantee of the requested bandwidth specified when you made the reservation using RSVP) to determine bandwidth reservation.

A host uses RSVP to request a specific QoS service from the network on behalf of an application data stream. RSVP requests the particular QoS, but it is up to the interface queueing mechanism to implement the reservation. RSVP carries the request through the network, visiting each node the network uses to carry the stream. At each node, RSVP attempts to make a resource reservation for the stream using its own admission control module, exclusive to RSVP, which determines whether the node has sufficient available resources to supply the requested QoS.

Note For RSVP, an application could send traffic at a rate higher than the requested QoS, but the application is guaranteed only the minimum requested rate. If bandwidth is available, traffic surpassing the requested rate will go through if sent; if bandwidth is not available, the exceeding traffic will be dropped.

If the required resources are available and the user is granted administrative access, the RSVP daemon sets arguments in the packet classifier and packet scheduler to obtain the desired QoS. The classifier determines the QoS class for each packet and the scheduler orders packet transmission to achieve the promised QoS for each stream. If either resource is unavailable or the user is denied administrative permission, the RSVP program returns an error notification to the application process that originated the request.

WFQ or WRED sets up the packet classification and the scheduling required for the reserved flows. Using WFQ, RSVP can deliver an integrated services Guaranteed Rate Service. Using WRED, it can deliver a Controlled Load Service.

For information on how to configure RSVP, see the chapter "Configuring RSVP" in this book.

RSVP Support for Low Latency Queueing

RSVP is a network-control protocol that provides a means for reserving network resources—primarily bandwidth—to guarantee that applications sending end-to-end across networks achieve the desired QoS.

RSVP enables real-time traffic (which includes voice flows) to reserve resources necessary for low latency and bandwidth guarantees.

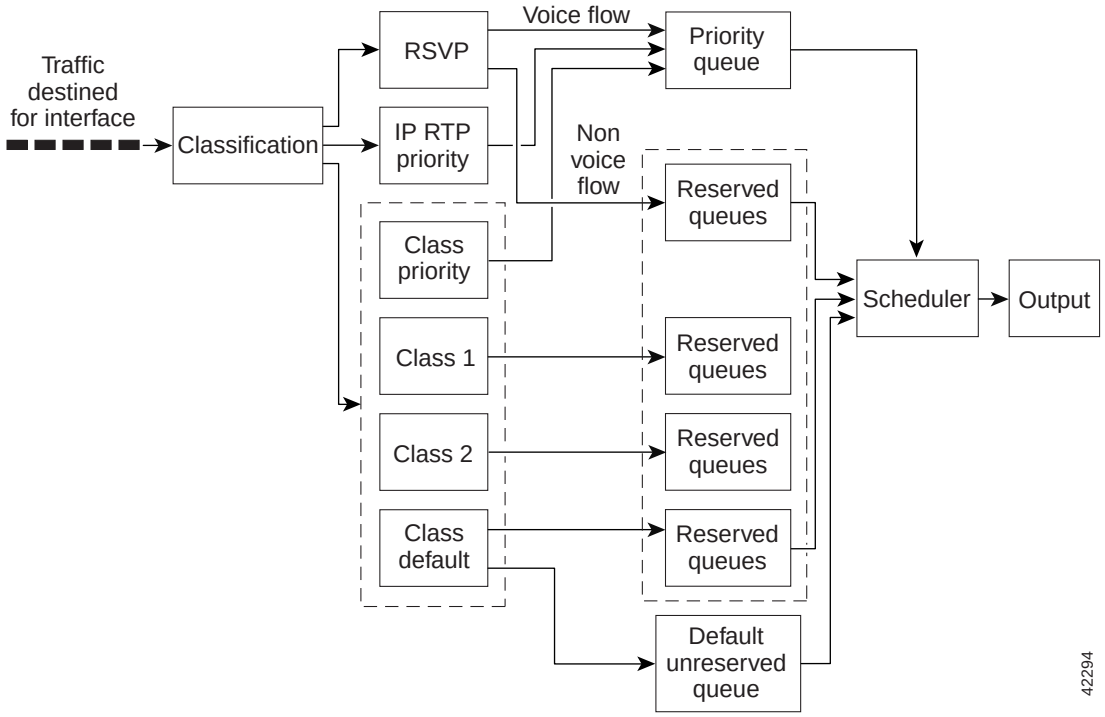
Voice traffic has stringent delay and jitter requirements. It must have very low delay and minimal jitter per hop to avoid degradation of end-to-end QoS. This requirement calls for an efficient queueing implementation, such as low latency queueing (LLQ), that can service voice traffic at almost strict priority in order to minimize delay and jitter.

RSVP uses WFQ to provide fairness among flows and to assign a low weight to a packet to attain priority. However, the preferential treatment provided by RSVP is insufficient to minimize the jitter because of the nature of the queueing algorithm itself. As a result, the low latency and jitter requirements of voice flows might not be met in the prior implementation of RSVP and WFQ.

RSVP provides admission control. However, to provide the bandwidth and delay guarantees for voice traffic and get admission control, RSVP must work with LLQ. The RSVP Support for LLQ feature allows RSVP to classify voice flows and queue them into the priority queue within the LLQ system while simultaneously providing reservations for nonvoice flows by getting a reserved queue.

Figure 2 shows how RSVP operates with other Voice over IP (VoIP) features, such as **ip rtp priority**, using the same queueing mechanism, LLQ.

Figure 2 RSVP Support for LLQ



RSVP is the only protocol that provides admission control based on the availability of network resources such as bandwidth. LLQ provides a means to forward voice traffic with strict priority ahead of other data traffic. When combined, RSVP support for LLQ provides admission control and forwards voice flows with the lowest possible latency and jitter.

High priority nonvoice traffic from mission-critical applications can continue to be sent without being adversely affected by voice traffic.

Nonconformant traffic receives best-effort treatment, thereby avoiding any degradation that might otherwise occur for all traffic.

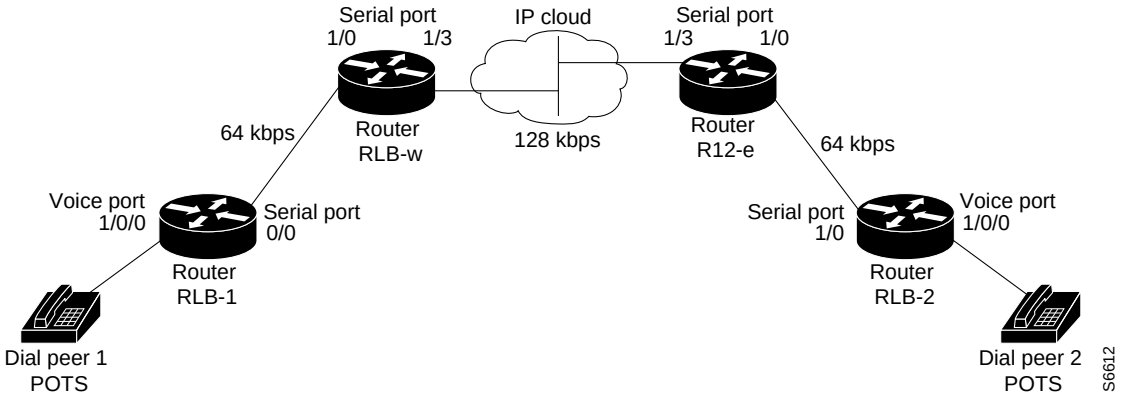
The RSVP Support for LLQ feature supports the following RFCs:

- RFC 2205, *Resource Reservation Protocol*
- RFC 2210, *RSVP with IETF Integrated Services*
- RFC 2211, *Controlled-Load Network Element Service*
- RFC 2212, *Specification of Guaranteed Quality of Service*
- RFC 2215, *General Characterization Parameters for Integrated Service Network Elements*

Figure 3 shows a sample network topology with LLQ running on each interface. This configuration guarantees QoS for voice traffic.

Note If the source is incapable of supporting RSVP, then the router can proxy on behalf of the source.

Figure 3 Topology Showing LLQ on Each Interface



For information on how to configure the RSVP Support for LLQ feature, see the "Configuring RSVP Support for LLQ" module.

Restrictions

The following restrictions apply to the RSVP Support for LLQ feature:

- The LLQ is not supported on any tunnels.
- RSVP support for LLQ is dependent on the priority queue. If LLQ is not available on any interface or platform, then RSVP support for LLQ is not available.

Prerequisites

The network must support the following Cisco IOS features before RSVP support for LLQ is enabled:

- RSVP
- WFQ or LLQ (WFQ with priority queue support)

RSVP Support for Frame Relay

Network administrators use queueing to manage congestion on a router interface or a virtual circuit (VC). In a Frame Relay environment, the congestion point might not be the interface itself, but the VC because of the committed information rate (CIR). For real-time traffic (voice flows) to be sent in atimely manner, the data rate must not exceed the CIR or packets might be dropped, thereby affecting voice quality. Frame Relay Traffic Shaping (FRTS) is configured on the interfaces to control the outbound traffic rate by preventing the router from exceeding the CIR. This type of configuration means that fancy queueing such as class-based WFQ (CBWFQ), LLQ, or WFQ, can run on the VC to provide the QoS guarantees for the traffic.

Previously, RSVP reservations were not constrained by the CIR of the outbound VC of the flow. As a result, oversubscription could occur when the sum of the RSVP traffic and other traffic exceeded the CIR.

The RSVP Support for Frame Relay feature allows RSVP to function with per-VC (data-link connection identifier (DLCI) queueing for voice-like flows. Traffic shaping must be enabled in a Frame Relay environment for accurate admission control of resources (bandwidth and queues) at the congestion point, that is, the VC itself. Specifically, RSVP can function with VCs defined at the interface and subinterface levels. There is no limit to the number of VCs that can be configured per interface or subinterface.

RSVP Bandwidth Allocation and Modular QoS Command Line Interface (CLI)

RSVP can use an interface (or a PVC) queueing algorithm, such as WFQ, to ensure QoS for its data flows.

Admission Control

When WFQ is running, RSVP can co-exist with other QoS features on an interface (or PVC) that also reserve bandwidth and enforce QoS. When you configure multiple bandwidth-reserving features (such as RSVP, LLQ, CB-WFQ, and **ip rtp priority**), portions of the interface's (or PVC's) available bandwidth may be assigned to each of these features for use with flows that they classify.

An internal interface-based (or PVC-based) bandwidth manager prevents the amount of traffic reserved by these features from oversubscribing the interface (or PVC). You can view this pool of available bandwidth using the **show queue** command, and it is configured (as a percentage of the interface’s or PVC's capacity) via the **max-reserved-bandwidth** command.

When you configure features such as LLQ and CB-WFQ, any classes that are assigned a bandwidth reserve their bandwidth at the time of configuration, and deduct this bandwidth from the bandwidth manager. If the configured bandwidth exceeds the interface's capacity, the configuration is rejected.

When RSVP is configured, no bandwidth is reserved. (The amount of bandwidth specified in the **ip rsvp bandwidth** command acts as a strict upper limit, and does **not** guarantee admission of any flows.) Only when an RSVP reservation arrives does RSVP attempt to reserve bandwidth out of the remaining pool of available bandwidth (that is, the bandwidth that has not been dedicated to traffic handled by other features.)

Data Packet Classification

By default, RSVP performs an efficient flow-based, datapacket classification to ensure QoS for its reserved traffic. This classification runs prior to queueing consideration by **ip rtp priority** or CB-WFQ. Thus, the use of a CB-WFQ class or **ip rtp priority** command is **not** required in order for RSVP data flows to be granted QoS. Any **ip rtp priority** or CB-WFQ configuration will not match RSVP flows, but they will reserve additional bandwidth for any non-RSVP flows that may match their classifiers.

Benefits

The benefits of this feature include the following:

- RSVP now provides admission control based on the VC minimum acceptable outgoing (minCIR) value, if defined, instead of the amount of bandwidth available on the interface.
- RSVP provides QoS guarantees for high priority traffic by reserving resources at the point of congestion, that is, the Frame Relay VC instead of the interface.
- RSVP provides support for point-to-point and multipoint interface configurations, thus enabling deployment of services such as VoIP in Frame Relay environments with QoS guarantees.

- RSVP, CBWFQ, and the **ip rtp priority** command do not oversubscribe the amount of bandwidth available on the interface or the VC even when they are running simultaneously. Prior to admitting a reservation, these features (and the **ip rtp priority** command) consult with an internal bandwidth manager to avoid oversubscription.
- IP QoS features can now be integrated seamlessly from IP into Frame Relay environments with RSVP providing admission control on a per-VC (DLCI) basis.

The RSVP Support for Frame Relay feature supports the following MIB and RFCs:

- RFC 2206, *RSVP Management Information Base using SMIv2*
- RFC 220, *Resource Reservation Protocol*
- RFC 2210, *RSVP with IETF Integrated Services*
- RFC 221, *Controlled-Load Network Element Service*
- RFC 2212, *Specification of Guaranteed Quality of Service*
- RFC 2215, *General Characterization Parameters for Integrated Service Network Elements*

For information on how to configure RVSP Support for Frame Relay, see the "Configuring RSVP Support for Frame Relay" module.

Restrictions

The following restrictions apply to the RSVP Support for Frame Relay feature:

- Interface-level Generic Traffic Shaping (GTS) is not supported.
- VC-level queueing and interface-level queueing on the same interface are not supported.
- Nonvoice RSVP flows are not supported.
- Multicast flows are not supported.

Prerequisites

The network must support the following Cisco IOS features before RSVP support for Frame Relay is enabled:

- RSVP
- WFQ on the VC
- LLQ
- Frame Relay Forum (FRF).12 on the interface

RSVP-ATM QoS Interworking

The RSVP-ATM QoS Interworking feature provides support for Controlled Load Service using RSVP over an ATM core network. This feature requires the ability to signal for establishment of switched virtual circuits (SVCs) across the ATM cloud in response to RSVP reservation request messages. To meet this requirement, RSVP over ATM supports mapping of RSVP sessions to ATM SVCs.

The RSVP-ATM QoS Interworking feature allows you to perform the following tasks:

- Configure an interface or subinterface to dynamically create SVCs in response to RSVP reservation request messages. To ensure defined QoS, these SVCs are established having QoS profiles consistent with the mapped RSVP flow specifications (flowspecs).
- Attach Distributed Weighted Random Early Detection (DWRED) group definitions to the Enhanced ATM port adapter (PA-A3) interface to support per-VC DWRED drop policy. Use of per-VC DWRED ensures that if packets must be dropped, then best-effort packets are dropped first and not those that conform to the appropriate QoS determined by the token bucket of RSVP.
- Configure the IP Precedence and ToS values to be used for packets that conform to or exceed QoS profiles. As part of its input processing, RSVP uses the values that you specify to set the ToS and IP Precedence bits on incoming packets. If per-VC DWRED is configured, it then uses the ToS and IP Precedence bit settings on the output interface of the same router in determining which packets to drop. Also, interfaces on downstream routers use these settings in processing packets.

This feature is supported on Cisco 7500 series routers with a VIP2-50 and Enhanced ATM port adapter (PA-A3). The hardware provides the traffic shaping required by the feature and satisfies the OC-3 rate performance requirement.

How It Works

Traditionally, RSVP has been coupled with WFQ. WFQ provides bandwidth guarantees to RSVP and gives RSVP visibility to all packets visible to it. This visibility allows RSVP to identify and mark packets pertinent to it.

The RSVP-ATM QoS Interworking feature allows you to decouple RSVP from WFQ, and instead associate it with ATM SVCs to handle reservation request messages (and provide bandwidth guarantees) and NetFlow to make packets visible to RSVP.

To configure an interface or subinterface to use the RSVP-ATM QoS Interworking feature, use the **ip rsvp svc-required** command. Then, whenever a new RSVP reservation is requested, the router software establishes a new ATM SVC to service the reservation.

To ensure correspondence between RSVP and ATM SVC values, the software algorithmically maps the rate and burst size parameters in the RSVP flowspec to the ATM sustained cell rate (SCR) and maximum burst size (MBS). For the peak cell rate (PCR), it uses the value you configure or it defaults to the line rate. RSVP-ATM QoS Interworking requires an Enhanced ATM port adapter (PA-A3) with OC-3 speed.

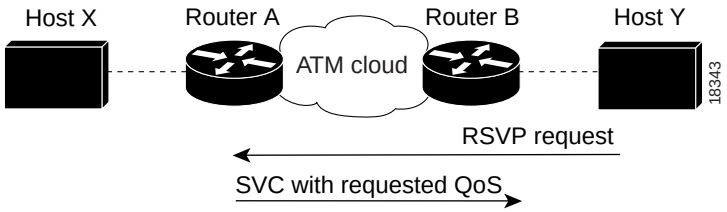
When a packet belonging to a reserved flow arrives on the interface or subinterface, the RSVP-ATM QoS Interworking software uses a token bucket to manage bandwidth guarantees. It measures actual traffic rates against the reservation flowspec to determine if the packet conforms to or exceeds the flowspec. Using values you configure for conformant or exceeding traffic, it sets the IP Precedence and ToS bits in the ToS byte of the header of the packet and delivers the packet to the appropriate virtual circuit (VC) for transmission. For the RSVP-ATM QoS Interworking feature, packets are shaped before they are sent on the ATM SVC. Shaping creates back pressure to the Versatile Interface Processor (VIP) when the offered load exceeds the rate.

The RSVP-ATM QoS Interworking software uses per-SVC DWRED to drop packets when shaping causes a queue to build up on the VIP. Use of per-SVC DWRED allows RSVP to deliver Controlled Load Service class, which requires that reserved packets experience performance equivalent to that of an unloaded network (which is one with very low loss and moderate delay). For a more detailed account of how the RSVP-ATM QoS Interworking feature works, see the following example scenario.

An Example Scenario

To understand the behavior of the RSVP-ATM QoS Interworking feature, consider the following example, which uses a Cisco 7500 router with VIP ingress and egress interfaces and RSVP ingress functionality implemented on the Route Switch Processor (RSP). Figure 4 illustrates this example; it shows a pair of routers that communicate over the ATM cloud. In this example, a single PVC is used for RSVP request messages and an ATM SVC is established to handle each new reservation request message.

Figure 4 Two Routers Connected over an ATM Core Network



Host X, which is upstream from Router A, is directly connected to Router A using FDDI. Host Y, which is downstream from Router B, is directly connected to Router B using FDDI. (In an alternative configuration, these host-router connections could use ATM VCs.)

For the RSVP-ATM QoS Interworking feature, reservations are needed primarily between routers across the ATM backbone network. To limit the number of locations where reservations are made, you can enable RSVP selectively only at subinterfaces corresponding to router-to-router connections across the ATM backbone network. Preventing reservations from being made between the host and the router both limits VC usage and reduces load on the router.

RSVP RESV messages flow from receiving host to sending host. In this example, Host Y is the sending host and Host X is the receiving host. (Host Y sends a RESV message to Host X.) Router B, which is at the edge of the ATM cloud, receives the RESV message and forwards it upstream to Router A across the PVC used for control messages. The example configuration shown in Figure 4 uses one PVC; as shown, it carries the RSVP request.

The ingress interface on Router A is configured for RSVP-ATM, which enables it to establish for each request an SVC to service any new RSVP RESV reservations made on the interface. When it receives a reservation request, the interface on Router A creates a new nonreal-time variable bit rate (nRTVBR) SVC with the appropriate QoS characteristics. The QoS characteristics used to establish the SVC result from algorithmic mapping of the flowspec in the RSVP RESV message to the appropriate set of ATM signalling parameters.

In this example, Controlled Load Service is used as the QoS class. The ATM PCR parameter is set to the line rate. If the **ip rsvp atm-peak-rate-limit** command is used on the interface to configure a rate limiter, the PCR is set to the peak rate limiter. The ATM SCR parameter is set to the RSVP flowspec rate and the ATM MBS is set to the RSVP flowspec burst size. Packets are shaped before they are sent on the ATM SVC. Shaping creates back pressure to the VIP when the offered load exceeds the rate.

When a new SVC is set up to handle a reservation request, another state is also set up including a classifier state that uses a source and destination addresses and port numbers of the packet to determine which, if any, reservation the packet belongs to. Also, a token bucket is set up to ensure that if a source sends more data than the data rate and MBS parameters of its flowspec specify, the excess traffic does not interfere with other reservations.

The following section describes more specifically, how data traverses the path.

When a data packet destined for Router B arrives at Router A, before they traverse the ATM cloud, the source and destination addresses and port numbers of the packet are checked against the RSVP filter specification (filterspec) to determine if the packet matches a reservation.

If the packet does not match a reservation, it is sent out the best-effort PVC to Router B. If a packet matches a reservation, it is further processed by RSVP. The packet is checked against the token bucket of the reservation to determine whether it conforms to or exceeds the token bucket parameters. (All packets matching a reservation are sent out on the SVC of the reservation to prevent misordering of packets.)

To introduce differentiation between flowspec-conformant and flowspec-exceeding packets, you can specify values for RSVP-ATM to use in setting the IP Precedence and ToS bits of the packets. To specify these values, you use the **ip rsvp precedence** and **ip rsvp tos** commands. When you set different precedence values for conformant and exceeding packets and use a preferential drop policy such as DWRED, RSVP-ATM ensures that flowspec-exceeding packets are dropped prior to flowspec-conformant packets when the VC is congested.

For information on how to configure the RSVP-ATM QoS Interworking feature, see the "Configuring RSVP-ATM QoS Interworking" module.

COPS for RSVP

Common Open Policy Service (COPS) is a protocol for communicating network traffic policy information to network devices. RSVP is a means for reserving network resources—primarily bandwidth—to guarantee that applications sending end-to-end across the Internet will perform at the desired speed and quality.

Combined, COPS with RSVP gives network managers centralized monitoring and control of RSVP, including the following abilities:

- Ensure adequate bandwidth and jitter and delay bounds for time-sensitive traffic such as voice transmission
- Ensure adequate bandwidth for multimedia applications such as video conferencing and distance learning
- Prevent bandwidth-hungry applications from delaying top priority flows or harming the performance of other applications customarily run over the same network

In so doing, COPS for RSVP supports the following crucial RSVP features:

- Admission control. The RSVP reservation is accepted or rejected based on *end-to-end* available network resources.
- Bandwidth guarantee. The RSVP reservation, if accepted, will guarantee that those reserved resources will continue to be available while the reservation is in place.
- Media-independent reservation. An end-to-end RSVP reservation can span arbitrary lower layer media types.
- Data classification. While a reservation is in place, data packets belonging to that RSVP flow are separated from other packets and forwarded as part of the reserved flow.
- Data policing. Data packets belonging to an RSVP flow that exceed the reserved bandwidth size are marked with a lower packet precedence.

Note In order to use the COPS for RSVP feature, your network must be running Cisco IOS 12.1(1)T or later releases. Moreover, a compatible policy server must be connected to the network, such as the Cisco *COPS QoS Policy Manager*.

Note The Cisco IOS 12.1(2)T release of COPS for RSVP does not support RSVP+.

COPS for RSVP functions on the following interfaces:

- Ethernet
- Fast Ethernet
- High-Speed Serial Interface (HSSI): V.35, EIA/TIA-232
- T1

The COPS for RSVP feature supports the following RFCs:

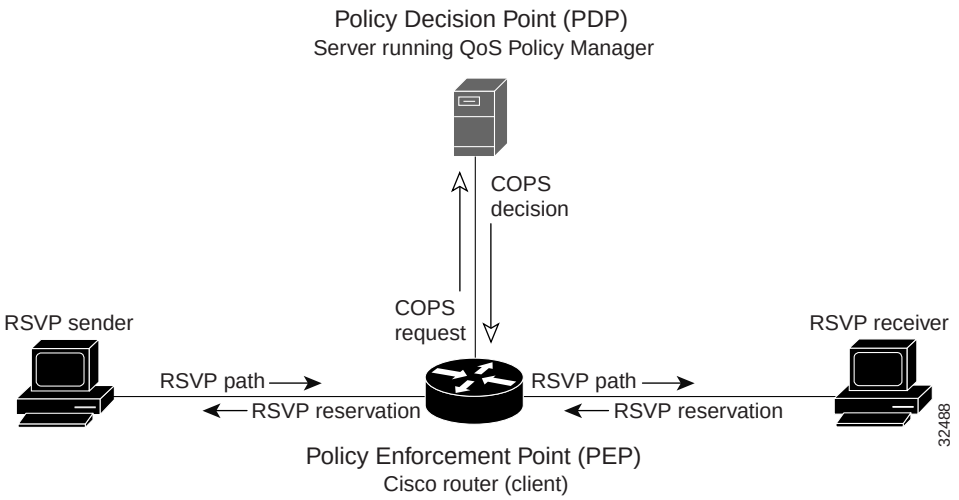
- RFC 2749, *COPS Usage for RSVP*
- RFC 2205, *Resource ReSerVation Protocol (RSVP)*
- RFC 2748, *The COPS (Common Open Policy Service) Protocol*

How It Works

This section provides a high-level overview of how the COPS for RSVP feature works on your network, and provides the general steps for configuring the COPS for RSVP feature.

Figure 5 is a sample arrangement of COPS with RSVP.

Figure 5 Sample Arrangement of COPS with RSVP



To configure a router to process all RSVP messages coming to it according to policies stored on a particular policy server (called the Policy Decision Point, or PDP), perform the following steps:

1. At the PDP server enter the policies using the Cisco COPS QoS Policy Manager or a compatible policy manager application.

2. Configure the router (through its command-line interface) to request decisions from the server regarding RSVP messages.

   After that configuration, network flows are processed by the router designated as the Policy Enforcement Point (PEP), as follows:

   a. When an RSVP signalling message arrives at the router, the router asks the PDP server how to process the message, either to accept, reject, forward, or install the message.

   b. The PDP server sends its decision to the router, which then processes the message as instructed.

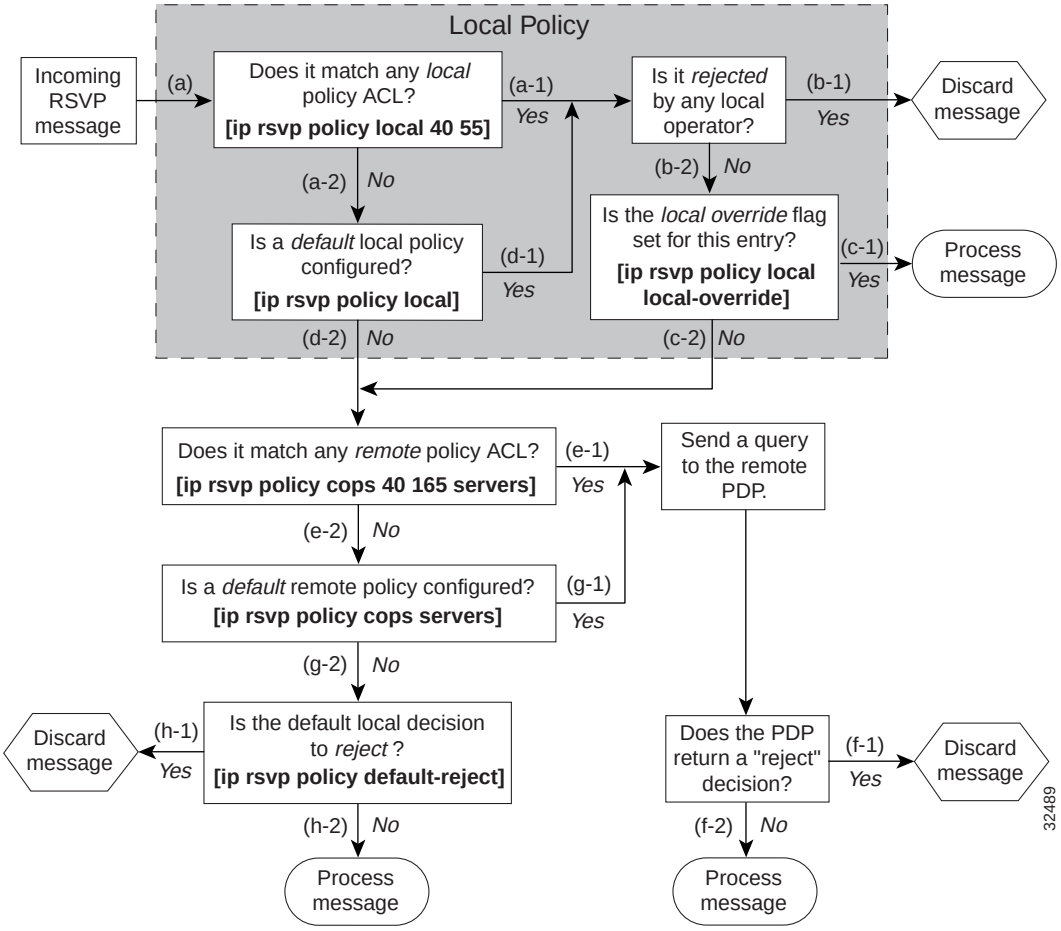
3. Alternatively, you may configure the router to make those decisions itself (“locally”) without it needing to consult first with the PDP server. (The local feature is not supported in this release but will be in a future release.)

A Detailed Look at COPS for RSVP Functioning

Figure 6 traces options available in policy management of RSVP message flows. For each option, an example of the router configuration command used for setting that option is given in brackets and boldface type.

The shaded area covers local policy operations; the remainder of the figure illustrates remote policy operation. (Configuring local policy will be available in a future release.)

Figure 6 Steps in Processing RSVP PATH and RESV Messages



The following information is keyed to the figure:

a. The router receives a PATH or RESV message and first tries to adjudicate it locally (that is, without referring to the policy server). If the router has been configured to adjudicate specific access control lists (ACLs) locally and the message matches one of those lists (a-1), the policy module of the router applies the operators with which it had been configured. Otherwise, policy processing continues (a-2).

b. For each message rejected by the operators, the router sends an error message to the sender and removes the PATH or RESV message from the database (b-1). If the message is not rejected, policy processing continues (b-2).

c. If the local override flag is set for this entry, the message is immediately accepted with the specified policy operators (c-1). Otherwise, policy processing continues (c-2).

d. If the message does not match any ACL configured for local policy (a-2), the router applies the default local policy (d-1). However, if no default local policy has been configured, the message is directed toward remote policy processing (d-2).

e. If the router has been configured with specific ACLs against specific policy servers (PDPs), and the message matches one of these ACLs, the router sends that message to the specific PDP for adjudication (e-1). Otherwise, policy processing continues (e-2).

f. If the PDP specifies a "reject" decision (f-1), the message is discarded and an error message is sent back to the sender, indicating this condition. If the PDP specifies an "accept" decision (f-2), the message is accepted and processed using normal RSVP processing rules.

g. If the message does not match any ACL configured for specific PDPs (e-2), the router applies the *default* PDP configuration. If a default COPS configuration has been entered, policy processing continues (g-1). Otherwise, the message is considered to be unmatched (g-2).

If the default policy decision for unmatched messages is to reject (h-1), the message is immediately discarded and an ERROR message is sent to thesender indicating this condition. Otherwise, the message is accepted and processed using normal RSVP processing rules (h-2).

Here are additional details about PDP-PEP communication and processing:

- Policy request timer. Whenever a request for adjudication (of any sort) is sent to a PDP, a 30-second timer associated with the PATH or RESV message is started. If the timer runs out before the PDP replies to the request, the PDP is assumed to be down and the request is given to the default policy (step g-2 in Figure 6).
- PDP tracking of PEP reservations. When the PDP specifies that a reservation can be installed, this reservation must then be installed on the router. Once bandwidth capacity has been allocated and the reservation installed, the policy module of the PEP sends a COMMIT message to the PDP. But if the reservation could not be installed because of insufficient resources, the reservation is folded back to the noninstalled state and a NO-COMMIT message is sent to the PDP. If the reservation was also new (no previous state), then a DELETE REQUEST message instead is sent to the PDP. In these ways, the PDP can keep track of reservations on the PEP.
- Resynchronization. If the PDP sends a SYNCHRONIZE-REQUEST message to the PEP, the policy module of the PEP scans its database for all paths and reservations that were previously adjudicated by this PDP, and resends requests for them. The previously adjudicated policy information is retained until a new decision is received. When all the PATH or RESV states have been reported to the PDP, a SYNCHRONIZE-COMPLETE message is sent by the policy module to the PDP. The PEP also sends queries concerning all flows that were locally adjudicated while the PDP was down.
- Readjudication:
  - So long as flows governed by the RSVP session continue to pass through the PEP router, the PDP can unilaterally decide to readjudicate any of the COPS decisions of that session. For example, the PDP might decide that a particular flow that was earlier granted acceptance now needs to be rejected (due perhaps to a sudden preemption or timeout). In such cases, the PDP sends a new decision message to the PEP, which then adjusts its behavior accordingly.
  - If the PEP router receives a RESV message in which an object has changed, the policy decision needs to be readjudicated. For example, if the sender wants to increase or decrease the bandwidth reservation, a new policy decision must be made. In such cases, the policy flags previously applied to this session are retained, and the session is readjudicated.
- Tear-downs. The policy module of the PEP is responsible for notifying the PDP whenever a reservation or path that was previously established through policy is torn down for any reason. The PEP notifies the PDP by sending the PDP a DELETE REQUEST message.
- Connection management:
  - If the connection to the PDP is closed (either because the PDP closed the connection, a TCP/IP error occurred, or the keepalives failed), the PEP issues a CLIENT-CLOSE message and then attempts to reconnect to the same PDP. If the PEP receives a CLIENT-CLOSE message containing a PDP redirect address, the PEP attempts to connect to the redirected PDP.
  - If either attempt fails, the PEP attempts to connect to the PDPs previously specified in the configuration **ip rsvp policy cops servers** command, obeying the sequence of servers given in that command, always starting with the first server in that list.

  - If the PEP reaches the end of the list of servers without connecting, it waits a certain time (called the "reconnect delay") before trying again to connect to the first server in the list. This reconnect delay is initially 30 seconds, and doubles each time the PEP reaches the end of the list without having connected, until the reconnect delay becomes its maximum of 30 minutes. As soon as a connection is made, the delay is reset to 30 seconds.

- Replacement objects—The matrix in Table 1 identifies objects that the PDP can replace within RSVP messages passing through the PEP. An x in the column indicates that the PDP can replace the particular object within RSVP messages.

Table 1 Matrix for Objects the PDP Can Replace Within RSVP Messages

Message Context	Objects				Items Affected
	Policy	TSpec	Flowspec	Errorspec	
Path In	x	x	•—	•—	• Installed PATH state. • All outbound PATH messages for this PATH.
Path Out	x	x	•—	•—	This refresh of the PATH (but not the installed PATH state).
Resv In	x	•—	x	•—	• Installed RESV state (incoming and traffic control installation). • All outbound RESV messages for this RESV.
Resv Alloc	•—	•—	x	•—	Installed resources for this session.
Resv Out	x	•—	x	•—	This particular refresh of the RESV message (but not the installed RESV state nor the traffic control allocation).
PathError In	x	•—	•—	x	The forwarded PATHERROR message.
PathError Out	x	•—	•—	x	The forwarded PATHERROR message.
ResvError In	x	•—	•—	x	All RESVERROR messages forwarded by this router.
ResvError Out	x	•—	•—	x	This particular forwarded RESVERROR message.

If an RSVP message whose object was replaced is later refreshed from upstream, the PEP keeps track of both the old and new versions of the object, and does not wrongly interpret the refresh as a change in the PATH or RESV state.

For information on how to configure COPS for RSVP, see the chapter "Configuring COPS for RSVP" in this book.

Subnetwork Bandwidth Manager

RSVP and its service class definitions are largely independent of the underlying network technologies. This independence requires that a user define the mapping of RSVP onto subnetwork technologies.

The Subnetwork Bandwidth Manager (SBM) feature answers this requirement for RSVP in relation to IEEE 802-based networks. SBM specifies a signalling method and protocol for LAN-based admission control for RSVP flows. SBM allows RSVP-enabled routers and Layer 2 and Layer 3 devices to support reservation of LAN resources for RSVP-enabled data flows. The SBM signalling method is similar to that of RSVP itself. SBM protocol entities have the following features:

- Reside in Layer 2 or Layer 3 devices.
- Can manage resources on a segment. A segment is a Layer 2 physical segment shared by one or more senders, such as a shared Ethernet or Token Ring wire.
- Can become candidates in a dynamic election process that designates one SBM as the segment manager. The elected candidate is called the Designated Subnetwork Bandwidth Manager (DSBM). The elected DSBM is responsible for exercising admission control over requests for resource reservations on a managed segment.

A managed segment includes those interconnected parts of a shared LAN that are not separated by DSBMs. The presence of a DSBM makes the segment a managed one. One or more SBMs may exist on a managed segment, but there can be only one DSBM on each managed segment.

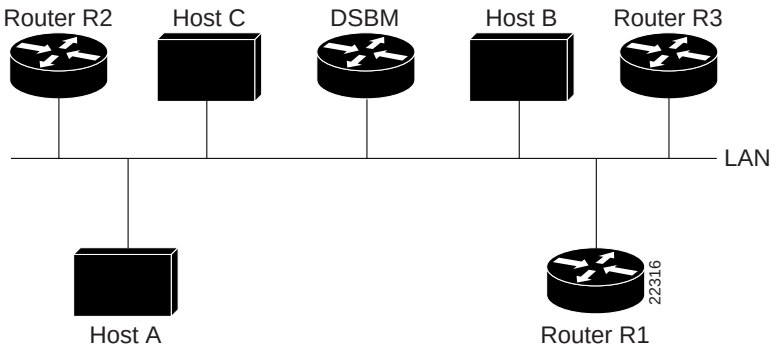
You can configure an interface on routers connected to the segment to participate in the DSBM election process. The contender configured with the highest priority becomes the DSBM for the managed segment.

If you do not configure a router as a DSBM candidate and RSVP is enabled, then the system interacts with the DSBM if a DSBM is present on the segment. In fact, if a DSBM, identifying itself as such, exists on the segment, the segment is considered a managed segment and all RSVP message forwarding will be based on the SBM message forwarding rules. This behavior exists to allow cases in which you might not want an RSVP-enabled interface on a router connected to a managed segment interface to become a DSBM, but you want it to interact with the DSBM if one is present managing the segment.

Note SBM is not supported currently on Token Ring LANs.

Figure 7 shows a managed segment in a Layer 2 domain that interconnects a set of hosts and routers.

Figure 7 DSBM Managed Segment



When a DSBM client sends or forwards an RSVP PATH message over an interface attached to a managed segment, it sends the PATH message to the DSBM of the segment instead of to the RSVP session destination address, as is done in conventional RSVP processing. As part of its message processing procedure, the DSBM builds and maintains a PATH state for the session and notes the previous Layer 2 or Layer 3 hop from which it received the PATH message. After processing the PATH message, the DSBM forwards it toward its destination address.

The DSBM receives the RSVP RESV message and processes it in a manner similar to how RSVP itself handles reservation request processing, basing the outcome on available bandwidth. The procedure is as follows:

- If it cannot grant the request because of lack of resources, the DSBM returns a RESVERROR message to the requester.
- If sufficient resources are available and the DSBM can grant the reservation request, it forwards the RESV message toward the previous hops using the local PATH state for the session.

For information on how to configure SBM, see the "Configuring Subnetwork Bandwidth Manager" module.

Configuring RSVP

Configuring RSVP

This chapter describes the tasks for configuring the Resource Reservation Protocol (RSVP) feature, which is an IP service that allows end systems or hosts on either side of a router network to establish a reserved-bandwidth path between them to predetermine and ensure QoS for their data transmission.

For a complete description of the RSVP commands in this module, see the *Cisco IOS Quality of Service Solutions Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

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

RSVP allows end systems to request QoS guarantees from the network. The need for network resource reservations differs for data traffic versus for real-time traffic, as follows:

- Data traffic seldom needs reserved bandwidth because internetworks provide datagram services for data traffic. This asynchronous packet switching may not need guarantees of service quality. End-to-end controls between data traffic senders and receivers help ensure adequate transmission of bursts of information.
- Real-time traffic (that is, voice or video information) experiences problems when operating over datagram services. Because real-time traffic sends an almost constant flow of information, the network "pipes" must be consistent. Some guarantee must be provided that service between real-time hosts will not vary. Routers operating on a first-in, first-out (FIFO) basis risk unrecoverable disruption of the real-time information that is being sent.

Data applications, with little need for resource guarantees, frequently demand relatively lower bandwidth than real-time traffic. The almost constant high bit-rate demands of a video conference application and the bursty low bit-rate demands of an interactive data application share available network resources.

RSVP prevents the demands of traffic such as large file transfers from impairing the bandwidth resources necessary for bursty data traffic. When RSVP is used, the routers sort and prioritize packets much like a statistical time-division multiplexer (TDM) would sort and prioritize several signal sources that share a single channel.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

RSVP mechanisms enable real-time traffic to reserve resources necessary for consistent latency. A video conferencing application can use settings in the router to propagate a request for a path with the required bandwidth and delay for video conferencing destinations. RSVP will check and repeat reservations at regular intervals. By this process, RSVP can adjust and alter the path between RSVP end systems to recover from route changes.

Real-time traffic (unlike data traffic) requires a guaranteed network consistency. Without consistent QoS, real-time traffic faces the following problems:

- Jitter. A slight time or phase movement in a transmission signal can introduce loss of synchronization or other errors.
- Insufficient bandwidth. Voice calls use a digital signal level 0 (DS-0 at 64 kbps), video conferencing uses T1/E1 (1.544 Mbps or 2.048 Mbps), and higher-fidelity video uses much more.
- Delay variations. If the wait time between when signal elements are sent and when they arrive varies, the real-time traffic will no longer be synchronized and transmission may fail.
- Information loss. When signal elements drop or arrive too late, lost audio causes distortions with noise or crackle sounds. The lost video causes image blurring, distortions, or blackouts.

RSVP works in conjunction with weighted fair queueing (WFQ) or Random Early Detection (RED). This conjunction of reservation setting with packet queueing uses two key concepts: end-to-end flows with RSVP and router-to-router conversations with WFQ:

- RSVP flow. This is a stream that operates “multidestination simplex,” because data travels across it in only one direction: from the origin to the targets. Flows travel from a set of senders to a set of receivers. The flows can be merged or left unmerged, and the method of merging them varies according to the attributes of the application using the flow.
- WFQ conversation. This is the traffic for a single transport layer session or network layer flow that crosses a given interface. This conversation is identified from the source and destination address, protocol type, port number, or other attributes in the relevant communications layer.

RSVP allows for hosts to send packets to a subset of all hosts (multicasting). RSVP assumes that resource reservation applies primarily to multicast applications (such as video conferencing). Although the primary target for RSVP is multimedia traffic, a clear interest exists for the reservation of bandwidth for unicast traffic (such as Network File System (NFS) and virtual private network management). A unicast transmission involves a host sending packets to a single host.

For more information about RSVP, see the “Signalling Overview” module.

RSVP cannot be configured with VIP-distributed Cisco Express Forwarding (dCEF).

RSVP Reservation Types

These are the two types of multicast flows:

- Distinct reservation. This constitutes a flow that originates from exactly one sender.
- Shared reservation. This constitutes a flow that originates from one or more senders.

RSVP describes these reservations as having certain algorithmic attributes.

Distinct Reservation

An example of a distinct reservation is a video application in which each sender emits a distinct data stream that requires admission and management in a queue. Such a flow, therefore, requires a separate reservation per sender on each transmission facility it crosses (such as Ethernet, a High-Level Data Link Control (HDLC) line, a Frame Relay data-link connection identifier (DLCI), or an ATM virtual channel). RSVP refers to this distinct reservation as explicit and installs it using a Fixed Filter style of reservation.

Use of RSVP for unicast applications is generally a degenerate case of a distinct flow.

Shared Reservation

An example of a shared reservation also is an audio application in which each sender emits a distinct data stream that requires admission and management in a queue. However, because of the nature of the application, a limited number of senders are sending data at any given time. Such a flow, therefore, does not require a separate reservation per sender. Instead, it uses a single reservation that can be applied to any sender within a set as needed.

RSVP installs a shared reservation using a Wild Card or Shared Explicit style of reservation, with the difference between the two determined by the scope of application (which is either wild or explicit):

- The Wild Card Filter reserves bandwidth and delay characteristics for any sender and is limited by the list of source addresses carried in the reservation message.
- The Shared Explicit style of reservation identifies the flows for specific network resources.

Planning for RSVP Configuration

You must plan carefully to successfully configure and use RSVP on your network. At a minimum, RSVP must reflect your assessment of bandwidth needs on router interfaces. Consider the following questions as you plan for RSVP configuration:

- How much bandwidth should RSVP allow per end-user application flow? You must understand the "feeds and speeds" of your applications. By default, the amount reservable by a single flow can be the entire reservable bandwidth. You can, however, limit individual reservations to smaller amounts using the single flow bandwidth parameter. The reserved bandwidth value may not exceed the interface reservable amount, and no one flow may reserve more than the amount specified.
- How much bandwidth is available for RSVP? By default, 75 percent of the bandwidth available on an interface is reservable. If you are using a tunnel interface, RSVP can make a reservation for the tunnel whose bandwidth is the sum of the bandwidths reserved within the tunnel.
- How much bandwidth must be excluded from RSVP so that it can fairly provide the timely service required by low-volume data conversations? End-to-end controls for data traffic assume that all sessions will behave so as to avoid congestion dynamically. Real-time demands do not follow this behavior. Determine the bandwidth to set aside so bursty data traffic will not be deprived as a side effect of the RSVP QoS configuration.

Note Before entering RSVP configuration commands, you must plan carefully.

RSVP Implementation Considerations

You should be aware of RSVP implementation considerations as you design your reservation system. RSVP does not model all data links likely to be present on the internetwork. RSVP models an interface as having a queueing system that completely determines the mix of traffic on the interface; bandwidth or delay characteristics are only deterministic to the extent that this model holds. Unfortunately, data links are often imperfectly modeled this way. Use the following guidelines:

- Serial line interfaces—PPP; HDLC; Link Access Procedure, Balanced (LAPB); High-Speed Serial Interface (HSSI); and similar serial line interfaces are well modeled by RSVP. The device can, therefore, make guarantees on these interfaces. Nonbroadcast multiaccess (NBMA) interfaces are also most in need of reservations.
- Multiaccess LANs—These data links are not modeled well by RSVP interfaces because the LAN itself represents a queueing system that is not under the control of the device making the guarantees. The device guarantees which load it will offer, but cannot guarantee the competing loads or timings of loads that neighboring LAN systems will offer. The network administrator can use admission controls to control how much traffic is placed on the LAN. The network administrator, however, should focus on the use of admission in network design in order to use RSVP effectively.

  The Subnetwork Bandwidth Manager (SBM) protocol is an enhancement to RSVP for LANs. One device on each segment is elected the Designated SBM (DSBM). The DSBM handles all reservations on the segment, which prevents multiple RSVP devices from granting reservations and overcommitting the shared LAN bandwidth. The DSBM can also inform hosts of how much traffic they are allowed to send without valid RSVP reservations.
- Public X.25 networks—It is not clear that rate or delay reservations can be usefully made on public X.25 networks.

You must use a specialized configuration on Frame Relay and ATM networks, as discussed in the next sections.

Frame Relay Internetwork Considerations

The following RSVP implementation considerations apply as you design your reservation system for a Frame Relay internetwork:

- Reservations are made for an interface or subinterface. If subinterfaces contain more than one data-link control (DLC), the bandwidth required and the bandwidth reserved may differ. Therefore, RSVP subinterfaces of Frame Relay interfaces must contain exactly one DLC to operate correctly.
- In addition, Frame Relay DLCs have committed information rates (CIR) and burst controls (Committed Burst and Excess Burst) that may not be reflected in the configuration and may differ markedly from the interface speed (either adding up to exceed it or being substantially smaller). Therefore, the **ip rsvp bandwidth** interface configuration command must be entered for both the interface and the subinterface. Both bandwidths are used as admission criteria.

For example, suppose that a Frame Relay interface runs at a T1 rate (1.544 Mbps) and supports several DLCs to remote offices served by 128-kbps and 56-kbps lines. You must configure the amount of the total interface (75 percent of which is 1.158 Mbps) and the amount of each receiving interface (75 percent of which would be 96 and 42 kbps, respectively) that may be reserved. Admission succeeds only if enough bandwidth is available on the DLC (the subinterface) and on the aggregate interface.

ATM Internetwork Considerations

The following RSVP implementation considerations apply as you design your reservation system for an ATM internetwork:

- When ATM is configured, it most likely uses a usable bit rate (UBR) or an available bit rate (ABR) virtual channel (VC) connecting individual routers. With these classes of service, the ATM network makes a "best effort" to meet the bit-rate requirements of the traffic and assumes that the end stations are responsible for information that does not get through the network.
- This ATM service can open separate channels for reserved traffic having the necessary characteristics. RSVP should open these VCs and adjust the cache to make effective use of the VC for this purpose.

Resource Reservation Protocol Configuration Task List

After you have planned your RSVP configuration, enter the Cisco IOS commands that implement your configuration plan. To configure RSVP, perform the tasks described in the following sections. The task in the first section is required; the tasks in the remaining sections are optional.

- Enabling RSVP (Required)
- Entering Senders in the RSVP Database (Optional)
- Entering Receivers in the RSVP Database (Optional)
- Specifying Multicast Destinations (Optional)
- Controlling Which RSVP Neighbor Can Offer a Reservation (Optional)
- Enabling RSVP to Attach to NetFlow (Optional)
- Setting the IP Precedence and ToS Values (Optional)
- Monitoring RSVP (Optional)

See the end of this chapter for the section "RSVP Configuration for a Multicast Session Example."

Enabling RSVP

By default, RSPV is disabled so that it is backward compatible with systems that do not implement RSVP. To enable RSVP for IP on an interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ip rsvp bandwidth [interface-kbps] [single-flow-kbps]`	Enables RSVP for IP on an interface.

This command starts RSVP and sets the bandwidth and single-flow limits. The default maximum bandwidth is up to 75 percent of the bandwidth available on the interface. By default, the amount reservable by a flow can be up to the entire reservable bandwidth.

On subinterfaces, this command applies the more restrictive of the available bandwidths of the physical interface and the subinterface. For example, a Frame Relay interface might have a T1 connector nominally capable of 1.536 Mbps, and 64-kbps subinterfaces on 128-kbps circuits (64-kbps CIR). RSVP bandwidth can be configured on the main interface up to 1200 kbps, and on each subinterface up to 100 kbps.

Reservations on individual circuits that do not exceed 100 kbps normally succeed. If, however, reservations have been made on other circuits adding up to 1.2 Mbps, and a reservation is made on a subinterface that itself has enough remaining bandwidth, the reservation request will still be refused because the physical interface lacks supporting bandwidth.

Entering Senders in the RSVP Database

You can configure the router to behave as though it is periodically receiving an RSVP PATH message from the sender or previous hop routes containing the indicated attributes. To enter senders in the RSVP database, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip rsvp sender` *session-ip-address sender-ip-address* [**tcp** | **udp** | *ip-protocol*] *session-dport sender-sport previous-hop-ip-address previous-hop-interface bandwidth burst-size*	Enters the senders in the RSVP database. Enables a router to behave like it is receiving and processing RSVP PATH messages.

The related **ip rsvp sender-host** command enables a router to simulate a host generating RSVP PATH messages. It is used mostly for debugging and testing purposes.

Entering Receivers in the RSVP Database

You can configure the router to behave as though it is continuously receiving an RSVP RESV message from the originator containing the indicated attributes. To enter receivers in the RSVP database, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip rsvp reservation` *session-ip-address sender-ip-address* [**tcp** | **udp** | *ip-protocol*] *session-dport sender-sport next-hop-ip-address next-hop-interface* {**ff** | **se** | **wf**} {**rate** | **load**} *bandwidth burst-size*	Enters the receivers in the RSVP database. Enables a router to behave like it is receiving and processing RSVP RESV messages.

The related **ip rsvp reservation-host** command enables a router to simulate a host generating RSVP RESV messages. It is used mostly for debugging and testing purposes.

Specifying Multicast Destinations

If RSVP neighbors are discovered to be using User Datagram Protocol (UDP) encapsulation, the router will automatically generate UDP-encapsulated messages for consumption by the neighbors.

However, in some cases, a host will not originate such a message until it has first heard from the router, which it can only do via UDP. You must instruct the router to generate UDP-encapsulated RSVP multicasts whenever it generates an IP-encapsulated multicast.

To specify multicast destinations that should receive UDP-encapsulated messages, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip rsvp udp-multicasts [multicast-address]`	Specifies multicast destinations that should receive UDP-encapsulated messages.

Controlling Which RSVP Neighbor Can Offer a Reservation

By default, any RSVP neighbor may offer a reservation request. To control which RSVP neighbors can offer a reservation request, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip rsvp neighbor access-list-number`	Limits which routers may offer reservations.

When this command is configured, only neighbors conforming to the access list are accepted. The access list is applied to the IP header.

Enabling RSVP to Attach to NetFlow

To enable RSVP to attach itself to NetFlow so that it can receive information about packets in order to update its token bucket and set IP precedence as required, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ip rsvp flow-assist`	Enables RSVP to attach itself to NetFlow.

This task is optional for the following reason: When the interface is configured with the **ip rsvp svc-required** command to use ATM switched virtual circuits (SVCs), RSVP automatically attaches itself to NetFlow to perform packet flow identification (in which case you need not perform this task). However, if you want to perform IP Precedence-type of service (ToS) bit setting in every packet without using ATM SVCs, then you must use the **ip rsvp flow-assist** command to instruct RSVP to attach itself to NetFlow.

Note If you use WFQ, then the ToS and IP Precedence bits will be set only on data packets that RSVP sees, due to congestion.

Setting the IP Precedence and ToS Values

To configure the IP Precedence and ToS values to be used to mark packets in an RSVP reserved path that either conform to or exceed the RSVP flow specification (flowspec), use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# ip rsvp precedence {conform` *precedence-value* `| exceed` *precedence-value*`}`	Sets the IP Precedence conform and exceed values.
Step 2	`Router(config-if)# ip rsvp tos {conform` *tos-value* `| exceed` *tos-value*`}`	Sets the ToS conform and exceed values.

Note You must configure the **ip rsvp flow-assist** command if you want to set IP Precedence or ToS values in every packet and you are not using ATM SVCs; that is, you have not configured the **ip rsvp svc-required** command.

The ToS byte in the IP header defines the three high-order bits as IP Precedence bits and the five low-order bits as ToS bits.

The router software checks the source and destination addresses and port numbers of a packet to determine if the packet matches an RSVP reservation. If a match exists, as part of its input processing, RSVP checks the packet for conformance to the flowspec of the reservation. During this process, RSVP determines if the packet conforms to or exceeds the flowspec, and it sets the IP header IP Precedence and ToS bits of the packet accordingly. These IP Precedence and ToS bit settings are used by per-VC Distributed Weighted Random Early Detection (DWRED) on the output interface, and they can be used by interfaces on downstream routers.

The combination of scheduling performed by the Enhanced ATM port adapter (PA-A3) and the per-SVC DWRED drop policy ensures that any packet that matches a reservation but exceeds the flowspec (that is, it does not conform to the token bucket for the reservation) is treated as if it were a best-effort packet. It is sent on the SVC for the reservation, but its IP precedence is marked to ensure that it does not interfere with conforming traffic.

To display the configured IP Precedence bit values and ToS bit values for an interface, use the **show ip rsvp** command.

Monitoring RSVP

To allow a user on a remote management station to monitor RSVP-related information, use the following command in global configuration mode:

Command	Purpose
`Router(config)# snmp-server enable traps rsvp`	Sends RSVP notifications.

After you configure the RSVP reservations that reflect your network resource policy, to verify the resulting RSVP operations, use the following commands in EXEC mode, as needed:

Command	Purpose
Router# **show ip rsvp interface** [*type number*]	Displays RSVP-related interface information.
Router# **show ip rsvp installed** [*type number*]	Displays RSVP-related filters and bandwidth information.
Router# **show ip rsvp neighbor** [*type number*]	Displays current RSVP neighbors.
Router# **show ip rsvp sender** [*type number*]	Displays RSVP sender information.
Router# **show ip rsvp request** [*type number*]	Displays RSVP request information.
Router# **show ip rsvp reservation** [*type number*]	Displays RSVP receiver information.

RSVP Configuration for a Multicast Session Example

This section describes configuration of RSVP on three Cisco 4500 routers for a multicast session.

For information on how to configure RSVP, see the section "Resource Reservation Protocol Configuration Task List" in this chapter.

The three routers form the router network between an RSVP sender application running on an upstream (end system) host and an RSVP receiver application running on a downstream (end system) host—neither host is shown in this example.

The router network includes three routers: Router A, Router B, and Router C. The example presumes that the upstream High-Speed Serial Interface (HSSI) interface 0 of Router A links to the upstream host. Router A and Router B are connected by the downstream Ethernet interface1 of Router A, which links to the upstream interface Ethernet 1 of Router B. Router B and Router C are connected by the downstream HSSI interface 0 of Router B, which links to the upstream HSSI interface 0 of Router C. The example presumes that the downstream Ethernet interface 2 of Router C links to the downstream host.

Typically, an RSVP-capable application running on an end system host on one side of a router network sends either unicast or multicast RSVP PATH (Set Up) messages to the destination end system or host on the other side of the router network with which it wishes to communicate. The initiating application is referred to as the sender; the target or destination application is called the receiver. In this example, the sender runs on the host upstream from Router A and the receiver runs on the host downstream from Router C. The router network delivers the RSVP PATH messages from the sender to the receiver. The receiver replies with RSVP RESV messages in an attempt to reserve across the network the requested resources that are required between itself and the sender. The RSVP RESV messages specify the parameters for the requisite QoS that the router network connecting the systems should attempt to offer.

This example does not show the host that would run the sender application and the host that would run the receiver application. Normally, the first router downstream from the sender in the router network—in this case, Router A—would receive the RSVP PATH message from the sender. Normally, the last router in the router network—that is, the next hop upstream from the host running the receiver application, in this case, Router C—would receive an RSVP RESV message from the receiver.

Because this example does not explicitly include the hosts on which the sender and receiver applications run, the routers have been configured to act as if they were receiving PATH messages from a sender and RESV messages from a receiver. The commands used for this purpose, allowing RSVP to be more fully illustrated in the example, are the **ip rsvp sender** command and the **ip rsvp reservation** command. On Router A, the following command has been issued:

```
ip rsvp sender 225.1.1.1 12.1.2.1 UDP 7001 7000 12.1.2.1 Hs0 20 1
```

This command causes the router to act as if it were receiving PATH messages destined to multicast address 225.1.1.1 from a source 12.1.2.1. The previous hop of the PATH message is 12.1.2.1, and the message was received on HSSI interface 0.

On Router C, the following command has been issued:

```
ip rsvp reservation 225.1.1.1 12.1.2.1 UDP 7001 7000 9.1.2.1 Et2 FF LOAD 8 1
```

This command causes the router to act as if it were receiving RESV messages for the session with multicast destination 225.1.1.1. The messages request a Fixed Filter reservation to source 12.1.2.1, and act as if they had arrived from a receiver on Ethernet interface 2 with address 9.1.2.1.

In the example, the RSVP PATH messages flow in one direction: downstream from the sender, which in this example is Router A. (If the host were to initiate the RSVP PATH message, the message would flow from the host to Router A.) Router A sends the message downstream to Router B, and Router B sends it downstream to Router C. (If the downstream host were the actual receiver, Router C would send the RSVP PATH message downstream to the receiver host.) Each router in the router network must process the RSVP PATH message and route it to the next downstream hop.

The RSVP RESV messages flow in one direction: upstream from the receiver (in this example, Router C), upstream from Router C to Router B, and upstream from Router B to Router A. If the downstream host were the receiver, the message would originate with the host, which would send it to Router C. If the upstream host were the sender, the final destination of the RSVP RESV message would be the upstream host. At each hop, the router receiving the RSVP RESV message must determine whether it can honor the reservation request.

The **ip rsvp bandwidth** command both enables RSVP on an interface and specifies the amount of bandwidth on the interface that can be reserved (and the amount of bandwidth that can be allocated to a single flow). To ensure QoS for the RSVP reservation, WFQ is configured on the interfaces enabled for the reservation.

If the router network is capable of offering the specified (QoS) level of service, then an end-to-end reserved path is established. If not, the reservation attempt is rejected and a RESV ERROR message is sent to the receiver. The ability of each router in the network to honor the requested level of service is verified, link by link, as the RSVP RESV messages are sent across the router network to the sender. However, the data itself for which the bandwidth is reserved travels one way only: from the sender to receiver across an established PATH. Therefore, the QoS is effective in only one direction. This is the common case for one-to-many multicast data flows.

After the three routers in the example are configured, the **show ip rsvp sender** and **show ip rsvp reservation** commands will make visible the PATH and RESV state.

Router A Configuration

On Router A, RSVP is enabled on Ethernet interface 1 with 10 kbps to be reserved for the data transmission. A weighted fair queue is reserved on this interface to ensure RSVP QoS. (On Router A, RSVP is also enabled on HSSI interface 0 with 1 kbps reserved, but this bandwidth is used simply for passing messages.)

```
!
version 12.0
service config
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service udp-small-servers
service tcp-small-servers
!
hostname routerA
!
ip subnet-zero
```

```
no ip domain-lookup
ip multicast-routing
ip dvmrp route-limit 20000
!
!
interface Ethernet0
 ip address 2.0.0.193 255.0.0.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 media-type 10BaseT
!
interface Ethernet1
 ip address 11.1.1.2 255.0.0.0
 no ip directed-broadcast
 ip pim dense-mode
 ip rsvp bandwidth 10 10
 fair-queue 64 256 1000
 media-type 10BaseT
!
interface Hssi0
 ip address 12.1.1.1 255.0.0.0
 no ip directed-broadcast
 ip pim dense-mode
 ip rsvp bandwidth 1 1
!
interface ATM0
 no ip address
 no ip directed-broadcast
 shutdown
!
router ospf 100
 network 11.0.0.0 0.255.255.255 area 10
 network 12.0.0.0 0.255.255.255 area 10
!
ip classless
ip rsvp sender 225.1.1.1 12.1.2.1 UDP 7001 7000 12.1.2.1 Hs0 20 1
!
line con 0
 exec-timeout 0 0
 length 0
 transport input none
line aux 0
line vty 0 4
 login
!
end
```

Router B Configuration

On Router B, RSVP is enabled on HSSI interface 0 with 20 kbps to be reserved for the data transmission. A weighted fair queue is reserved on this interface to ensure RSVP QoS. (On Router B, RSVP is also enabled on Ethernet interface 1 with 1 kbps reserved, but this bandwidth is used simply for passing messages.)

```
!
version 12.0
service config
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service udp-small-servers
service tcp-small-servers
!
```

```
hostname routerB
!
ip subnet-zero
no ip domain-lookup
ip multicast-routing
ip dvmrp route-limit 20000
clock calendar-valid
!
interface Ethernet0
 ip address 2.0.0.194 255.0.0.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 media-type 10BaseT
!
interface Ethernet1
 ip address 11.1.1.1 255.0.0.0
 no ip directed-broadcast
 ip pim dense-mode
 ip rsvp bandwidth 1 1
 media-type 10BaseT
!
interface Hssi0
 ip address 10.1.1.2 255.0.0.0
 no ip directed-broadcast
 ip pim dense-mode
 ip rsvp bandwidth 20 20
 fair-queue 64 256 1000
 hssi internal-clock
!
interface ATM0
 no ip address
 no ip directed-broadcast
 shutdown
!
router ospf 100
 network 10.0.0.0 0.255.255.255 area 10
 network 11.0.0.0 0.255.255.255 area 10
!
ip classless
!
line con 0
 exec-timeout 0 0
 length 0
 transport input none
line aux 0
line vty 0 4
 login
!
end
```

Router C Configuration

On Router C, RSVP is enabled on Ethernet interface 2 with 20 kbps to be reserved for the data transmission. A weighted fair queue is reserved on this interface to ensure RSVP QoS. (On Router C, RSVP is also enabled on HSSI interface 0 with 1 kbps reserved, but this bandwidth is used simply for passing messages.)

```
!
version 12.0
service config
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
```

```
service udp-small-servers
service tcp-small-servers
!
hostname routerC
!
ip subnet-zero
no ip domain-lookup
ip multicast-routing
ip dvmrp route-limit 20000
!
interface Ethernet0
 ip address 2.0.0.195 255.0.0.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 media-type 10BaseT
!
interface Ethernet1
 no ip address
 no ip directed-broadcast
 shutdown
 media-type 10BaseT
!
interface Ethernet2
 ip address 9.1.1.2 255.0.0.0
 no ip directed-broadcast
 ip pim dense-mode
 ip rsvp bandwidth 20 20
 fair-queue 64 256 1000
 media-type 10BaseT
!
interface Ethernet3
 no ip address
 no ip directed-broadcast
 shutdown
 media-type 10BaseT
!
interface Ethernet4
 no ip address
 no ip directed-broadcast
 shutdown
 media-type 10BaseT
!
interface Ethernet5
 no ip address
 no ip directed-broadcast
 shutdown
 media-type 10BaseT
!
interface Hssi0
 ip address 10.1.1.1 255.0.0.0
 no ip directed-broadcast
 ip pim dense-mode
 ip rsvp bandwidth 1 1
 hssi internal-clock
!
interface ATM0
 no ip address
 no ip directed-broadcast
 shutdown
!
router ospf 100
 network 9.0.0.0 0.255.255.255 area 10
 network 10.0.0.0 0.255.255.255 area 10
```

```
 network 11.0.0.0 0.255.255.255 area 10
!
ip classless
ip rsvp reservation 225.1.1.1 12.1.2.1 UDP 7001 7000 9.1.2.1 Et2 FF LOAD 8 1
!
line con 0
 exec-timeout 0 0
 length 0
 transport input none
line aux 0
line vty 0 4
 login
!
end
```

Control Plane DSCP Support for RSVP

Feature History

Release	Modification
Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.

This document describes the Cisco control plane differentiated services code point (DSCP) support for Resource Reservation Protocol (RSVP) feature. It identifies the supported platforms, provides configuration examples, and lists related IOS command line interface (CLI) commands.

This document includes the following major sections:

Feature Overview

Typically, networks operate on a best-effort delivery basis, which means that all traffic has equal priority and an equal chance of being delivered in a timely manner. When congestion occurs, all traffic has an equal chance of being dropped.

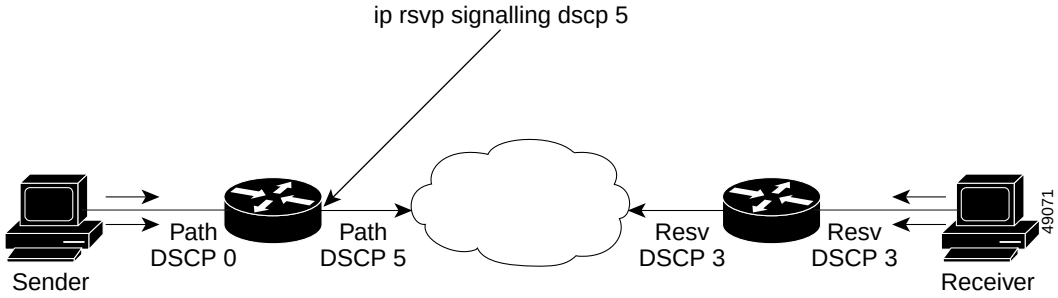
Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Before traffic can be handled according to its unique requirements, it must be identified or labeled. There are numerous classification techniques for doing this. These include Layer 3 schemes such as IP precedence or the differentiated services code point (DSCP), Layer 2 schemes such as 802.1P, and implicit characteristics of the data itself, such as the traffic type using the Real-Time Transport Protocol (RTP) and a defined port range.

The control plane DSCP support for RSVP feature allows you to set the priority value in the type of service (ToS) byte/differentiated services (DiffServ) field in the Internet Protocol (IP) header for RSVP messages. The IP header functions with resource providers such as weighted fair queueing (WFQ), so that voice frames have priority over data fragments and data frames. When packets arrive in a router's output queue, the voice packets are placed ahead of the data frames.

Figure 1 shows a path message originating from a sender with a DSCP value of 0 (the default) that is changed to 5 to give the message a higher priority and a reservation (resv) message originating from a receiver with a DSCP of 3.

Figure 1 Control Plane DSCP Support for RSVP



Raising the DSCP value reduces the possibility of packets being dropped, thereby improving call setup time in VoIP environments.

Benefits

Faster Call Setup Time

The control plane DSCP support for RSVP feature allows you to set the priority for RSVP messages. In a DiffServ QoS environment, higher priority packets get serviced before lower priority packets, thereby improving the call setup time for RSVP sessions.

Improved Message Delivery

During periods of congestion, routers drop lower priority traffic before they drop higher priority traffic. Since RSVP messages can now be marked with higher priority, the likelihood of these messages being dropped is significantly reduced.

Faster Recovery after Failure Conditions

When heavy congestion occurs, many packets are dropped. Network resources attempt to retransmit almost instantaneously resulting in further congestion. This leads to a considerable reduction in throughput.

Previously, RSVP messages were marked best effort and subject to being dropped by congestion avoidance mechanisms such as weighted random early detection (WRED). However, with the control plane DSCP support for RSVP feature, RSVP messages are likely to be dropped later, if at all, thereby providing faster recovery of RSVP reservations.

Restrictions

Control plane DSCP support for RSVP can be configured on interfaces and subinterfaces only. It affects all RSVP messages sent out the interface or that are on any logical circuit of the interface, including subinterfaces, permanent virtual circuits (PVCs), and switched virtual circuits (SVCs).

Related Features and Technologies

The control plane DSCP support for RSVP feature is related to QoS features, such as signalling, low latency queueing, and policing. (See the section on "Related Documents".)

Related Documents

The following documents provide additional information:

- "Quality of Service Overview" module
- *Cisco IOS Quality of Service Solutions Command Reference*

Supported Platforms

- Cisco 2600 series
- Cisco 3600 series (Cisco 3620, 3640, and 3660)
- Cisco 3810 multiservice access concentrator
- Cisco 7200 series
- Cisco 7500 route/switch processor (RSP) only
- Cisco 12000 series Gigabit Switch Router (GSR)

Supported Standards, MIBs, and RFCs

Standards

The control plane DSCP support for RSVP feature supports no new or modified standards.

MIBs

RFC 2206 (RSVP Management Information Base using SMIv2)

To obtain lists of MIBs supported by platform and Cisco IOS release and to download MIB modules, go to the Cisco MIB web site on Cisco.com at http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml.

RFCs

- RFC 2205 (Resource Reservation Protocol)

Prerequisites

The network must support the following Cisco IOS feature before control plane DSCP support for RSVP is enabled:

- Resource Reservation Protocol (RSVP)

Configuration Tasks

See the following sections for configuration tasks for the control plane DSCP support for RSVP feature. Each task in the list indicates whether the task is optional or required.

- Enabling RSVP on an Interface, page 4 (Required)
- Specifying the DSCP, page 4 (Required)

Enabling RSVP on an Interface

To enable RSVP on an interface, use the following command, beginning in interface configuration mode:

Command	Purpose
`Router(config-if)# ip rsvp bandwidth [interface-kbps] [single-flow-kbps]`	Enables RSVP on an interface.

Specifying the DSCP

To specify the DSCP, use the following command, beginning in interface configuration mode:

Command	Purpose
`Router(config-if)# ip rsvp signalling dscp [value]`	Specifies the DSCP to be used on all RSVP messages transmitted on an interface.

Verifying Control Plane DSCP Support for RSVP Configuration

To verify control plane DSCP support for RSVP configuration, enter the **show ip rsvp interface detail** command to display RSVP-related interface information.

In the following sample output from the **show ip rsvp interface detail** command, only the Se2/0 interface has DSCP configured. Interfaces that are not configured for DSCP do not show the DSCP value, which is 0 by default.

```
Router# show ip rsvp interface detail
Et1/1:
   Bandwidth:
     Curr allocated:0M bits/sec
     Max. allowed (total):7500K bits/sec
     Max. allowed (per flow):7500K bits/sec
   Neighbors:
     Using IP enacp:1.  Using UDP encaps:0
```

```
 Et1/2:
   Bandwidth:
     Curr allocated:0M bits/sec
     Max. allowed (total):7500K bits/sec
     Max. allowed (per flow):7500K bits/sec
   Neighbors:
     Using IP enacp:0.  Using UDP encaps:0

Se2/0:
   Bandwidth:
     Curr allocated:10K bits/sec
     Max. allowed (total):1536K bits/sec
     Max. allowed (per flow):1536K bits/sec
   Neighbors:
     Using IP enacp:1.  Using UDP encaps:0
   DSCP value used in Path/Resv msgs:0x6
   Burst Police Factor:300%
   RSVP:Data Packet Classification provided by: none
Router#
```

Monitoring and Maintaining Control Plane DSCP Support for RSVP

To monitor and maintain control plane DSCP support for RSVP, use the following command in EXEC mode:

Command	Purpose
`Router# show ip rsvp interface detail`	Displays RSVP-related information about interfaces.

Configuration Examples

This section provides a configuration example for the control plane DSCP support for RSVP feature.

```
 Router(config-if)# ip rsvp sig ?
 dscp  DSCP for RSVP signalling messages

 Router(config-if)# ip rsvp sig dscp ?
 <0-63>  DSCP value

Router(config-if)# ip rsvp sig dscp 48

Router# show run int e3/0
interface Ethernet3/0
ip address 50.50.50.1 255.255.255.0
fair-queue 64 256 235
ip rsvp signalling dscp 48
ip rsvp bandwidth 7500 7500
```

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Quality of Service Solutions Command Reference* at http://www.cisco.com/en/US/docs/ios/qos/command/reference/qos_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

- **ip rsvp signalling dscp**
- **show ip rsvp interface**

Glossary

CBWFQ—Class-based weighted fair queueing. A queueing mechanism that extends the standard WFQ functionality to provide support for user-defined traffic classes.

class-based weighted fair queueing—See CBWFQ.

differentiated services—See DiffServ.

differentiated services code point—See DSCP.

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

DSCP—Differentiated services code point. The six most significant bits of the 1-byte IP type of service (ToS) field. The per-hop behavior represented by a particular DSCP value is configurable. DSCP values range between 0 and 63.

IP precedence—The three most significant bits of the 1-byte type of service (ToS) field. IP precedence values range between zero for low priority and seven for high priority.

latency—The delay between the time a device receives a packet and the time that packet is forwarded out the destination port.

marking—The process of setting a Layer 3 DSCP value in a packet.

QoS—Quality of service. A measure of performance for a transmission system that reflects its transmission quality and service availability.

quality of service—See QoS.

Resource Reservation Protocol—See RSVP.

RSVP—Resource Reservation Protocol. A protocol for reserving network resources to provide quality of service guarantees to application flows.

ToS—Type of service. An 8-bit value in the IP header field.

type of service—See ToS.

Voice over IP—See VoIP.

VoIP—Voice over IP. The ability to carry normal telephony-style voice over an IP-based internet maintaining telephone-like functionality, reliability, and voice quality.

weighted fair queueing—See WFQ.

weighted random early detection—See WRED.

WFQ—Weighted fair queueing. A queue management algorithm that provides a certain fraction of link bandwidth to each of several queues, based on relative bandwidth applied to each of the queues.

WRED—Weighted random early detection. A congestion avoidance mechanism that slows traffic by randomly dropping packets when there is congestion.

Configuring RSVP Support for Frame Relay

This chapter describes the tasks for configuring the RSVP Support for Frame Relay feature.

For complete conceptual information, see the "Signalling Overview" module.

For a complete description of the RSVP Support for Frame Relay commands in this chapter, see the *Cisco IOS Quality of Service Solutions Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

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

RSVP Support for Frame Relay Configuration Task List

To configure Resource Reservation Protocol (RSVP) support for Frame Relay, perform the tasks described in the following sections. Each task is identified as either optional or required.

- Enabling Frame Relay Encapsulation on an Interface (Required)
- Configuring a Virtual Circuit (Required)
- Enabling Frame Relay Traffic Shaping on an Interface (Required)
- Enabling Enhanced Local Management Interface (Optional)
- Enabling RSVP on an Interface (Required)
- Specifying a Traffic Shaping Map Class for an Interface (Required)
- Defining a Map Class with WFQ and Traffic Shaping Parameters (Required)
- Specifying the CIR (Required)
- Specifying the Minimum CIR (Optional)
- Enabling WFQ (Required)
- Enabling FRF.12 (Required)
- Configuring a Path (Optional)
- Configuring a Reservation (Optional)
- Verifying RSVP Support for Frame Relay (Optional)

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

- Monitoring and Maintaining RSVP Support for Frame Relay (Optional)

See the end of this chapter for the section "RSVP Support for Frame Relay Configuration Examples."

Enabling Frame Relay Encapsulation on an Interface

To enable Frame Relay encapsulation on an interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface s3/0`	Enables an interface (for example, serial interface 3/0) and enters configuration interface mode.
Step 2	`Router(config-if)# encapsulation frame-relay [cisco | ietf]`	Enables Frame Relay and specifies the encapsulation method.

Configuring a Virtual Circuit

To configure a virtual circuit (VC), use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# frame-relay interface-dlci dlci`	Assigns a data-link connection identifier (DLCI) to a specified Frame Relay subinterface on a router or access server.

Enabling Frame Relay Traffic Shaping on an Interface

To enable Frame Relay Traffic Shaping (FRTS) on an interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# frame-relay traffic-shaping`	Enables traffic shaping and per-VC queueing for all permanent virtual circuits (PVCs) and switched virtual circuits (SVCs) on a Frame Relay interface.

Enabling Enhanced Local Management Interface

To enable enhanced Local Management Interface (LMI), use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# frame-relay lmi-type`	Selects the LMI type.

Enabling RSVP on an Interface

To enable RSVP on an interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ip rsvp bandwidth`	Enables RSVP on an interface.

Specifying a Traffic Shaping Map Class for an Interface

To specify a traffic shaping map class for an interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# frame-relay class` *name*	Associates a map class with an interface or subinterface.

Defining a Map Class with WFQ and Traffic Shaping Parameters

To define a map class with weighted fair queueing (WFQ) and traffic shaping parameters, use the following command in global configuration mode:

Command	Purpose
`Router(config)# map-class frame-relay` *map-class-name*	Defines parameters for a specified class.

Specifying the CIR

To specify the committed information rate (CIR), use the following command in map-class configuration mode:

Command	Purpose
`Router(config-map-class)# frame-relay cir {in | out}` *bps*	Specifies the maximum incoming or outgoing CIR for a Frame Relay VC.

Specifying the Minimum CIR

To specify the minimum acceptable incoming or outgoing CIR (minCIR) for a Frame Relay VC, use the following command in map-class configuration mode:

Command	Purpose
`Router(config-map-class)# frame-relay mincir {in | out} bps`	Specifies the minimum acceptable incoming or outgoing CIR for a Frame Relay VC. **Note** If the minCIR is not configured, then the admission control value is the CIR/2.

Enabling WFQ

To enable WQF, use the following command in map-class configuration mode:

Command	Purpose
`Router(config-map-class)# frame-relay fair-queue`	Enables WFQ on a PVC.

Enabling FRF.12

To enable FRF.12, use the following command in map-class configuration mode:

Command	Purpose
`Router(config-map-class)# frame-relay fragment fragment-size`	Enables Frame Relay fragmentation on a PVC.

Configuring a Path

To configure a path, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip rsvp sender`	Specifies the RSVP path parameters, including the destination and source addresses, the protocol, the destination and source ports, the previous hop address, the average bit rate, and the burst size.

Configuring a Reservation

To configure a reservation, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip rsvp reservation`	Specifies the RSVP reservation parameters, including the destination and source addresses, the protocol, the destination and source ports, the next hop address, the next hop interface, the reservation style, the service type, the average bit rate, and the burst size.

Verifying RSVP Support for Frame Relay

The following sections contain the procedures for verifying RSVP support for Frame Relay in either a multipoint configuration or a point-to-point configuration.

Multipoint Configuration

To verify RSVP support for Frame Relay in a multipoint configuration, perform the following steps:

Step 1 Enter the **show ip rsvp installed** command to display information about interfaces and their admitted reservations. The output in the following example shows that serial subinterface 3/0.1 has two reservations:

```
Router# show ip rsvp installed

RSVP:Serial3/0
BPS    To              From            Protoc DPort  Sport  Weight Conversation
RSVP:Serial3/0.1
BPS    To              From            Protoc DPort  Sport  Weight Conversation
40K    145.20.22.212   145.10.10.211   UDP    10     10     0      24
50K    145.20.21.212   145.10.10.211   UDP    10     10     6      25
```

Note Weight 0 is assigned to voice-like flows, which proceed to the priority queue.

Step 2 Enter the **show ip rsvp installed detail** command to display additional information about interfaces, subinterfaces, DLCI PVCs, and their current reservations.

Note In the following output, the first flow gets a reserved queue with a weight > 0, and the second flow gets the priority queue with a weight = 0.

```
Router# show ip rsvp installed detail

RSVP:Serial3/0 has the following installed reservations
RSVP:Serial3/0.1 has the following installed reservations
RSVP Reservation. Destination is 145.20.21.212, Source is 145.10.10.211,
  Protocol is UDP, Destination port is 10, Source port is 10
  Reserved bandwidth:50K bits/sec, Maximum burst:1K bytes, Peak rate:50K bits/sec
QoS provider for this flow:
    WFQ on FR PVC dlci 101 on Se3/0: RESERVED queue 25.  Weight:6
  Data given reserved service:0 packets (0M bytes)
  Data given best-effort service:0 packets (0 bytes)
  Reserved traffic classified for 68 seconds
  Long-term average bitrate (bits/sec):0M reserved, 0M best-effort
RSVP Reservation. Destination is 145.20.22.212, Source is 145.10.10.211,
  Protocol is UDP, Destination port is 10, Source port is 10
```

```
  Reserved bandwidth:40K bits/sec, Maximum burst:1K bytes, Peak rate:40K bits/sec
QoS provider for this flow:
    WFQ on FR PVC dlci 101 on Se3/0: PRIORITY queue 24.  Weight:0
  Data given reserved service:0 packets (0M bytes)
  Data given best-effort service:0 packets (0 bytes)
  Reserved traffic classified for 707 seconds
  Long-term average bitrate (bits/sec):0M reserved, 0M best-effort
```

Point-to-Point Configuration

To verify RSVP support for Frame Relay in a point-to-point configuration, perform the following steps:

Step 1 Enter the **show ip rsvp installed** command to display information about interfaces and their admitted reservations. The output in the following example shows that serial subinterface 3/0.1 has one reservation, and serial subinterface 3/0.2 has one reservation.

```
Router# show ip rsvp installed

RSVP:Serial3/0
BPS    To              From            Protoc DPort  Sport
RSVP:Serial3/0.1
BPS    To              From            Protoc DPort  Sport
50K    145.20.20.212   145.10.10.211   UDP    10     10

RSVP:Serial3/0.2
BPS    To              From            Protoc DPort  Sport
10K    145.20.21.212   145.10.10.211   UDP    11     11
```

Note Weight 0 is assigned to voice-like flows, which proceed to the priority queue.

Step 2 Enter the **show ip rsvp installed detail** command to display additional information about interfaces, subinterfaces, DLCI PVCs, and their current reservations.

Note In the following output, the first flow with a weight > 0 gets a reserved queue and the second flow with a weight = 0 gets the priority queue.

```
Router# show ip rsvp installed detail

RSVP:Serial3/0 has the following installed reservations
RSVP:Serial3/0.1 has the following installed reservations
RSVP Reservation. Destination is 145.20.20.212, Source is 145.10.10.211,
  Protocol is UDP, Destination port is 10, Source port is 10
  Reserved bandwidth:50K bits/sec, Maximum burst:1K bytes, Peak rate:50K bits/sec
QoS provider for this flow:
    WFQ on FR PVC dlci 101 on Se3/0: RESERVED queue 25.  Weight:6
  Data given reserved service:415 packets (509620 bytes)
  Data given best-effort service:0 packets (0 bytes)
  Reserved traffic classified for 862 seconds
  Long-term average bitrate (bits/sec):4724 reserved, 0M best-effort
RSVP Reservation. Destination is 145.20.20.212, Source is 145.10.10.211,
  Protocol is UDP, Destination port is 11, Source port is 11
  Reserved bandwidth:10K bits/sec, Maximum burst:1K bytes, Peak rate:10K bits/sec
QoS provider for this flow:
    WFQ on FR PVC dlci 101 on Se3/0: PRIORITY queue 24.  Weight:0
  Data given reserved service:85 packets (104380 bytes)
  Data given best-effort service:0 packets (0 bytes)
```

```
  Reserved traffic classified for 875 seconds
  Long-term average bitrate (bits/sec):954 reserved, 0M best-effort
RSVP:Serial3/0.2 has the following installedreservations

RSVP Reservation. Destination is 145.20.21.212, Source is 145.10.10.211,

  Protocol is UDP, Destination port is 11, Source port is 11
  Reserved bandwidth:10K bits/sec, Maximum burst:1K bytes, Peak rate:10Kbits/sec
QoS provider for this flow:
    WFQ on FR PVC dlci 101 on Se3/0:PRIORITY queue 24.  Weight:0
  Data given reserved service:85 packets (104380 bytes)
  Data given best-effort service:0 packets (0 bytes)
  Reserved traffic classified for 875 seconds
  Long-term average bitrate (bits/sec):954 reserved, 0M best-effort
```

Monitoring and Maintaining RSVP Support for Frame Relay

To monitor and maintain RSVP support for Frame Relay, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router# show ip rsvp installed`	Displays information about interfaces and their admitted reservations.
`Router# show ip rsvp installed detail`	Displays additional information about interfaces, DLCIs, and their admitted reservations.
`Router# show queueing`	Displays all or selected configured queueing strategies.

RSVP Support for Frame Relay Configuration Examples

The following sections provide RSVP support for Frame Relay configuration examples:

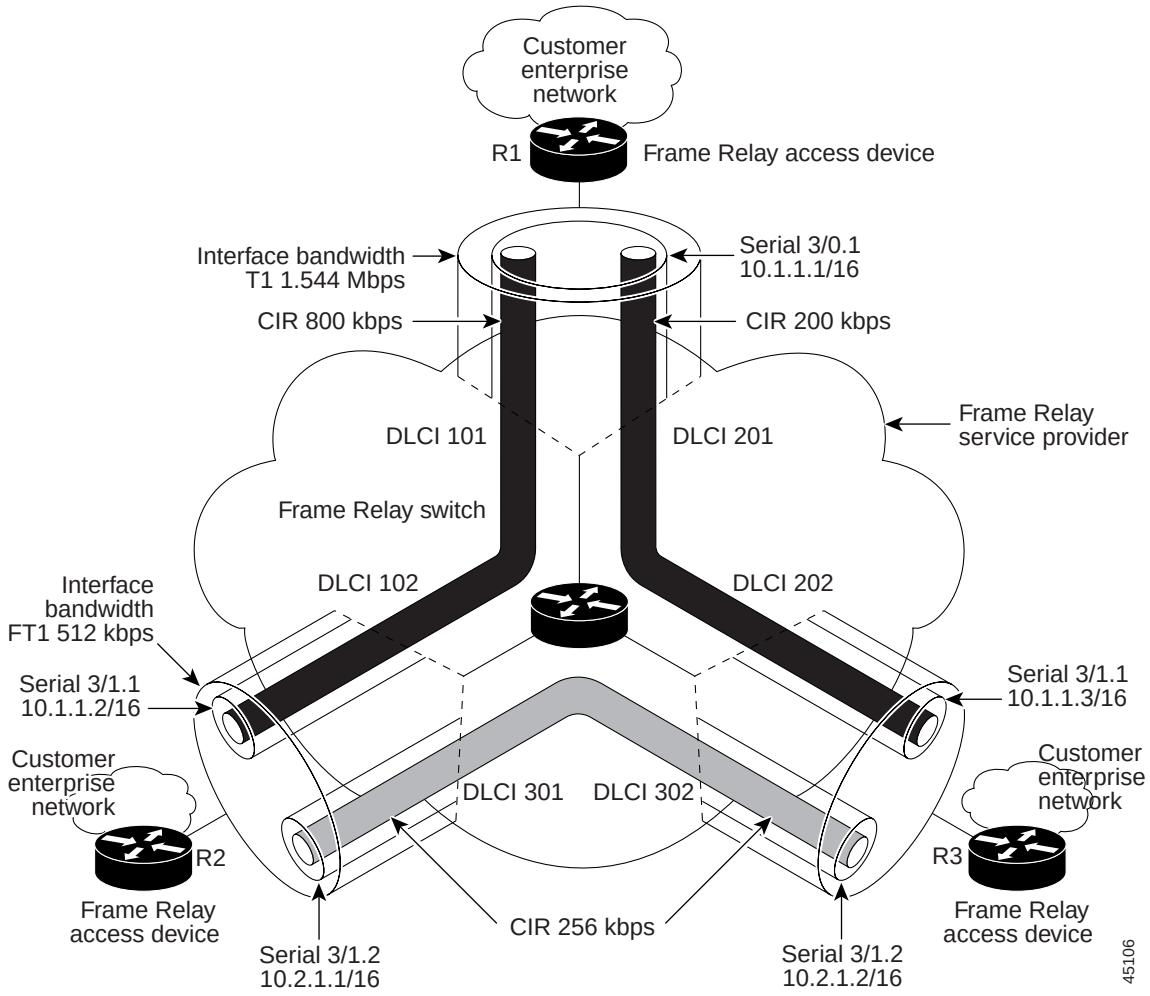
- Multipoint Configuration Example
- Point-to-Point Configuration Example

For information on how to configure the RSVP Support for Frame Relay feature, see the section "RSVP Support for Frame Relay Configuration Task List" in this chapter.

Multipoint Configuration Example

Figure 1 shows a multipoint interface configuration commonly used in Frame Relay environments in which multiple PVCs are configured on the same subinterface at router R1.

Figure 1 Multipoint Interface Configuration



RSVP performs admission control based on the minCIR of DLCI 101 and DLCI 201. The congestion point is not the 10.1.1.1/16 subinterface, but the CIR of DLCI 101 and DLCI 201.

The following example is a sample output for serial interface 3/0:

```
interface Serial3/0
 no ip address
 encapsulation frame-relay
 max-reserved-bandwidth 20
 no fair-queue
 frame-relay traffic-shaping
 frame-relay lmi-type cisco
 ip rsvp bandwidth 350 350
!
interface Serial3/0.1 multipoint
 ip address 10.1.1.1 255.255.0.0
 frame-relay interface-dlci 101
  class fr-voip
 frame-relay interface-dlci 201
  class fast-vcs
 ip rsvp bandwidth 350 350

ip rsvp pq-profile 6000 2000 ignore-peak-value
```

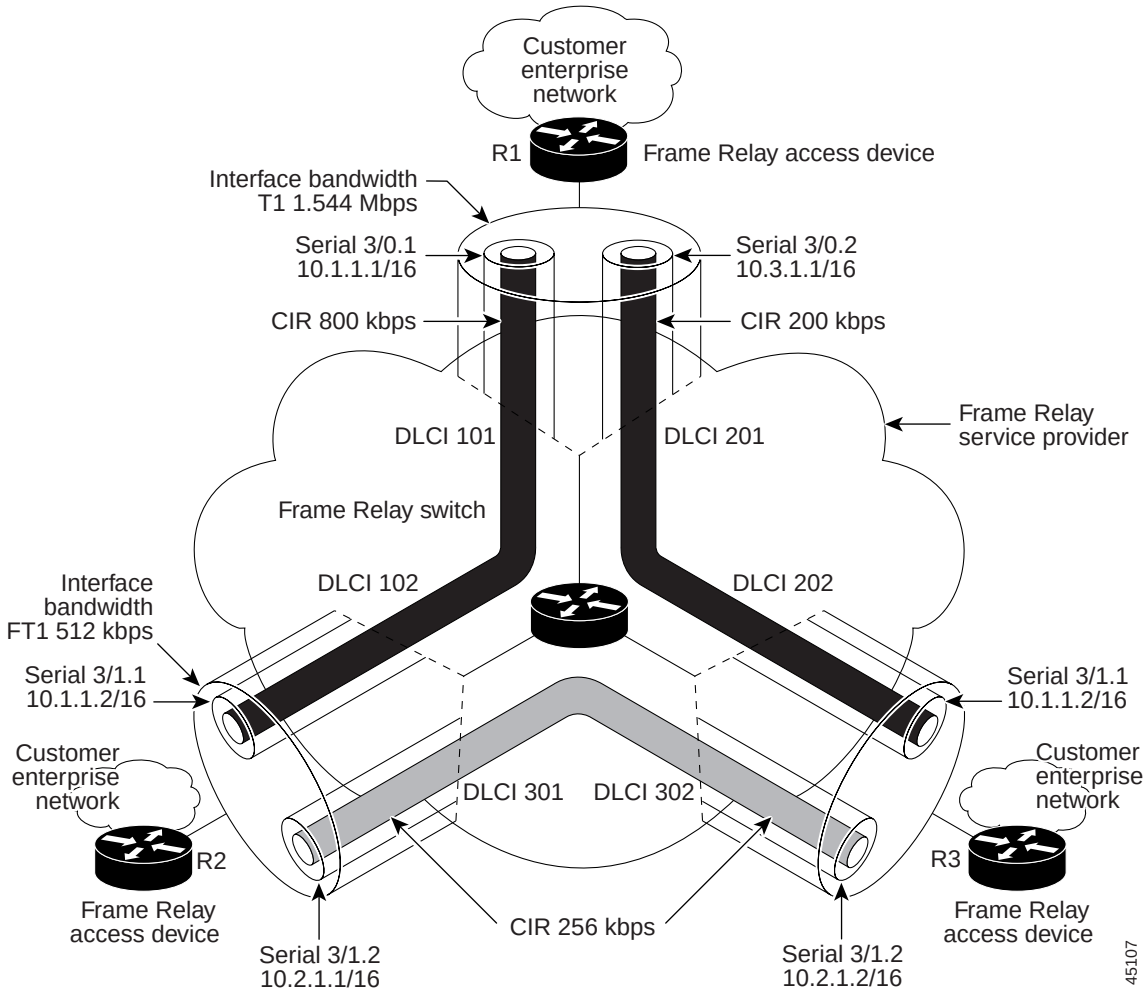
```
!
!
map-class frame-relay fr-voip
 frame-relay cir 800000
 frame-relay bc 8000
 frame-relay mincir 128000
 frame-relay fragment 280
 no frame-relay adaptive-shaping
 frame-relay fair-queue
!
map-class frame-relay fast-vcs
 frame-relay cir 200000
 frame-relay bc 2000
 frame-relay mincir 60000
 frame-relay fragment 280
 no frame-relay adaptive-shaping
 frame-relay fair-queue
!
```

Note When FRTS is enabled, the Frame Relay Committed Burst (Bc) value (in bits) should be configured to a maximum of 1/100th of the CIR value (in bits per second). This configuration ensures that the FRTS token bucket interval (Bc/CIR) does not exceed 10 Ms, and that voice packets are serviced promptly.

Point-to-Point Configuration Example

Figure 2 shows a point-to-point interface configuration commonly used in Frame Relay environments in which one PVC per subinterface is configured at router R1.

Figure 2 Sample Point-to-Point Interface Configuration



Notice that the router interface bandwidth for R1 is T1 (1.544 Mbps), whereas the CIR value of DLCI 201 toward R3 is 256 kbps. For traffic flows from R1 to R3 over DLCI 201, the congestion point is the CIR for DLCI 201. As a result, RSVP performs admission control based on the minCIR and reserves resources, including queues and bandwidth, on the WFQ system that runs on each DLCI.

The following example is sample output for serial interface 3/0:

```
interface Serial3/0
 no ip address
 encapsulation frame-relay
 max-reserved-bandwidth 20
 no fair-queue
 frame-relay traffic-shaping
 frame-relay lmi-type cisco
 ip rsvp bandwidth 500 500
!
interface Serial3/0.1 point-to-point
 ip address 10.1.1.1 255.255.0.0
 frame-relay interface-dlci 101
  class fr-voip
 ip rsvp bandwidth 350 350
!
interface Serial3/0.2 point-to-point
```

```
 ip address 10.3.1.1 255.255.0.0
 frame-relay interface-dlci 201
  class fast-vcs
 ip rsvp bandwidth 150 150

ip rsvp pq-profile 6000 2000 ignore-peak-value
!
!
map-class frame-relay fr-voip
 frame-relay cir 800000
 frame-relay bc 8000
 frame-relay mincir 128000
 frame-relay fragment 280
 no frame-relay adaptive-shaping
 frame-relay fair-queue
```

Note When FRTS is enabled, the Frame Relay Committed Burst (Bc) value (in bits) should be configured to a maximum of 1/100th of the CIR value (in bits per second). This configuration ensures that the FRTS token bucket interval (Bc/CIR) does not exceed 10 Ms, and that voice packets are serviced promptly.

RSVP Scalability Enhancements

This document describes the Cisco Resource Reservation Protocol (RSVP) scalability enhancements. It identifies the supported platforms, provides configuration examples, and lists related IOS command line interface (CLI) commands.

This document includes the following major sections:

Feature Overview

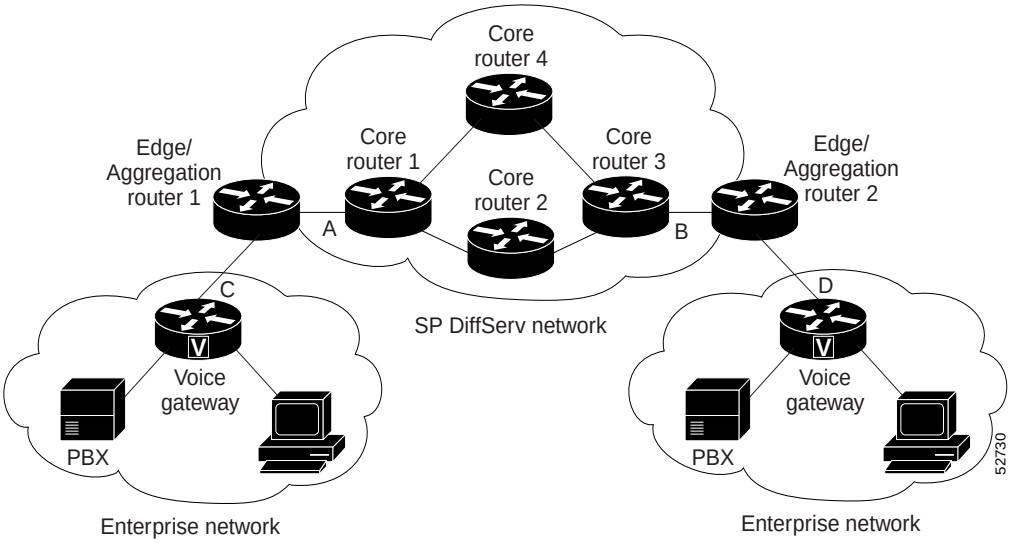
RSVP typically performs admission control, classification, policing, and scheduling of data packets on a per-flow basis and keeps a database of information for each flow. RSVP scalability enhancements let you select a resource provider (formerly called a quality of service (QoS) provider) and disable data packet classification so that RSVP performs admission control only. This facilitates integration with service provider (differentiated services (DiffServ)) networks and enables scalability across enterprise networks.

Class-based weighted fair queueing (CBWFQ) provides the classification, policing, and scheduling functions. CBWFQ puts packets into classes based on the differentiated services code point (DSCP) value in the packet's Internet Protocol (IP) header, thereby eliminating the need for per-flow state and per-flow processing.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Figure 1 shows two enterprise networks interconnected through a service provider (SP) network. The SP network has an IP backbone configured as a DiffServ network. Each enterprise network has a voice gateway connected to an SP edge/aggregation router via a wide area network (WAN) link. The enterprise networks are connected to a private branch exchange (PBX).

Figure 1 RSVP/DiffServ Integration Topology



The voice gateways are running classic RSVP, which means RSVP is keeping a state per flow and also classifying, marking, and scheduling packets on a per flow basis. The edge/aggregation routers are running classic RSVP on the interfaces (labeled C and D) connected to the voice gateways and running RSVP for admission control only on the interfaces connected to core routers 1 and 3. The core routers in the DiffServ network are not running RSVP, but are forwarding the RSVP messages to the next hop. The core routers inside the DiffServ network implement a specific per hop behavior (PHB) for a collection of flows that have the same DSCP value.

The voice gateways identify voice data packets and set the appropriate DSCP in their IP headers such that the packets are classified into the priority class in the edge/aggregation routers and in core routers 1, 2, 3 or 1, 4, 3.

The interfaces or the edge/aggregation routers (labeled A and B) connected to core routers 1 and 3 are running RSVP, but are doing admission control only per flow against the RSVP bandwidth pool configured on the DiffServ interfaces of the edge/aggregation routers. CBWFQ is performing the classification, policing, and scheduling functions.

Benefits

Enhanced Scalability

RSVP scalability enhancements handle similar flows on a per-class basis instead of a per-flow basis. Since fewer resources are required to maintain per-class QoS guarantees, faster processing results, thereby enhancing scalability.

Improved Router Performance

RSVP scalability enhancements improve router performance by reducing the cost for data packet classification and scheduling, which decrease central processing unit (CPU) resource consumption. The saved resources can then be used for other network management functions.

Restrictions

- Sources should not send marked packets without an installed reservation.
- Sources should not send marked packets that exceed the reserved bandwidth.
- Sources should not send marked packets to a destination other than the reserved path.

Related Features and Technologies

The RSVP scalability enhancements are related to QoS features such as signalling, classification, and congestion management. (See the section on "Related Documents".)

Related Documents

The following documents provide additional information:

- "Quality of Service Overview" module
- *Cisco IOS Quality of Service Solutions Command Reference*

Supported Platforms

- Cisco 2600 series
- Cisco 3600 series (Cisco 3620, 3640, and 3660)
- Cisco 3810 multiservice access concentrator
- Cisco 7200 series
- Cisco 7500 route/switch processor (RSP) only

Supported Standards, MIBs, and RFCs

Standards

RSVP scalability enhancements support no new or modified standards.

MIBs

RFC 2206 (RSVP Management Information Base using SMIv2)

To obtain lists of MIBs supported by platform and Cisco IOS release and to download MIB modules, go to the Cisco MIB web site on Cisco.com at http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml.

RFCs

- RFC 2205 (Resource Reservation Protocol)

Prerequisites

The network must support the following Cisco IOS features before the RSVP scalability enhancements are enabled:

- Resource Reservation Protocol (RSVP)
- Class-based weighted fair queueing (CBWFQ)

Configuration Tasks

See the following sections for configuration tasks for the RSVP scalability enhancements. Each task in the list indicates whether the task is optional or required.

- Enabling RSVP on an Interface (Required)
- Setting the Resource Provider (Required)
- Disabling Data Packet Classification (Required)
- Configuring Class and Policy Maps (Required)
- Attaching a Policy Map to an Interface (Required)

Enabling RSVP on an Interface

To enable RSVP on an interface, use the following command, beginning in interface configuration mode:

Command	Purpose
`Router(config-if)# ip rsvp bandwidth [interface-kbps] [single-flow-kbps]`	Enables RSVP on an interface.

Note The bandwidth that you configure on the interface must match the bandwidth that you configure for the CBWFQ priority queue. See the section on "Configuration Examples".

Setting the Resource Provider

Note Resource provider was formerly called QoS provider.

To set the resource provider, use the following command, beginning in interface configuration mode:

Command	Purpose
`Router(config-if)# ip rsvp resource-provider none`	Sets the resource provider to none.

Note Setting the resource provider to none instructs RSVP to*not* associate any resources, such as WFQ queues or bandwidth, with a reservation.

Disabling Data Packet Classification

To turn off (disable) data packet classification, use the following command, beginning in interface configuration mode:

Command	Purpose
`Router(config-if)# ip rsvp data-packet classification none`	Disables data packet classification.

Note Disabling data packet classification instructs RSVP *not* to process every packet, but to perform admission control only.

Configuring Class and Policy Maps

To configure class and policy maps, use the following commands, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# class-map class-map-name`	Specifies the name of the class forwhich you want to create or modify class map match criteria.
Step 2	`Router(config)# policy-map policy-map-name`	Specifies the name ofthe policy map to becreated, added to, or modified before you can configure policies for classes whose match criteria are defined in a class map.

Attaching a Policy Map to an Interface

To attach a policy map to an interface, use the following command, beginning in interface configuration mode:

Command	Purpose
`Router(config-if)# service-policy {input | output} policy-map-name`	Attaches a single policy map to one or more interfaces to specify the service policy for those interfaces.

Note If at the time you configure the RSVP scalability enhancements, there are existing reservations that use classic RSVP, no additional marking, classification, or scheduling is provided for these flows. You can also delete these reservations after you configure the RSVP scalability enhancements.

Verifying RSVP Scalability Enhancements Configuration

To verify RSVP scalability enhancements, use this procedure:

Step 1 Enter the **show ip rsvp interface detail** command to display information about interfaces, subinterfaces, resource providers, and data packet classification. The output in the following example shows that the ATM 6/0 interface has resource provider none configured and data packet classification is turned off:

```
Router# show ip rsvp interface detail
 AT6/0:
   Bandwidth:
     Curr allocated: 190K bits/sec
     Max. allowed (total): 112320K bits/sec
     Max. allowed (per flow): 112320K bits/sec
   Neighbors:
     Using IP encap: 1.  Using UDP encaps: 0
   DSCP value used in Path/Resv msgs: 0x30
   RSVP Data Packet Classification is OFF
   RSVP resource provider is: none
```

Note The last two lines in the preceding output verify that the RSVP scalability enhancements (disabled data packet classification and resource provider none) are present.

Step 2 Enter the **show ip rsvp installed detail** command to display information about interfaces, subinterfaces, their admitted reservations, bandwidth, resource providers, and data packet classification.

```
Router# show ip rsvp installed detail
RSVP: Ethernet3/3 has no installed reservations

RSVP: ATM6/0 has the following installed reservations
RSVP Reservation. Destination is 145.20.20.212, Source is 145.10.10.211,
  Protocol is UDP, Destination port is 14, Source port is 14
  Reserved bandwidth: 50K bits/sec, Maximum burst: 1K bytes, Peak rate: 50K bits/sec
  Min Policed Unit: 0 bytes, Max Pkt Size: 1514 bytes
  Resource provider for this flow: None
  Conversation supports 1 reservations
  Data given reserved service: 0 packets (0 bytes)
  Data given best-effort service: 0 packets (0 bytes)
  Reserved traffic classified for 54 seconds
  Long-term average bitrate (bits/sec): 0M reserved, 0M best-effort
RSVP Reservation. Destination is 145.20.20.212, Source is 145.10.10.211,
  Protocol is UDP, Destination port is 10, Source port is 10
  Reserved bandwidth: 20K bits/sec, Maximum burst: 1K bytes, Peak rate: 20K bits/sec
  Min Policed Unit: 0 bytes, Max Pkt Size: 1514 bytes
  Resource provider for this flow: None
  Conversation supports 1 reservations
  Data given reserved service: 0 packets (0 bytes)
  Data given best-effort service: 0 packets (0 bytes)
  Reserved traffic classified for 80 seconds
  Long-term average bitrate (bits/sec): 0M reserved, 0M best-effort
```

Step 3 Wait for a while, then enter the **show ip rsvp installed detail** command again. In the following output, notice there is no increment in the number of packets classified:

```
Router# show ip rsvp installed detail
RSVP: Ethernet3/3 has no installed reservations

RSVP: ATM6/0 has the following installed reservations
RSVP Reservation. Destination is 145.20.20.212, Source is 145.10.10.211,
  Protocol is UDP, Destination port is 14, Source port is 14
  Reserved bandwidth: 50K bits/sec, Maximum burst: 1K bytes, Peak rate: 50K bits/sec
  Min Policed Unit: 0 bytes, Max Pkt Size: 1514 bytes
  Resource provider for this flow: None
  Conversation supports 1 reservations
  Data given reserved service: 0 packets (0 bytes)
  Data given best-effort service: 0 packets (0 bytes)
  Reserved traffic classified for 60 seconds
  Long-term average bitrate (bits/sec): 0 reserved, OM best-effort
RSVP Reservation. Destination is 145.20.20.212, Source is 145.10.10.211,
  Protocol is UDP, Destination port is 10, Source port is 10
  Reserved bandwidth: 20K bits/sec, Maximum burst: 1K bytes, Peak rate: 20K bits/sec
  Min Policed Unit: 0 bytes, Max Pkt Size: 1514 bytes
  Resource provider for this flow: None
  Conversation supports 1 reservations
  Data given reserved service: 0 packets (0 bytes)
  Data given best-effort service: 0 packets (0 bytes)
  Reserved traffic classified for 86 seconds
  Long-term average bitrate (bits/sec): OM reserved, 0M best-effort
```

Monitoring and Maintaining RSVP Scalability Enhancements

To monitor and maintain RSVP scalability enhancements, use the following commands in EXEC mode:

Command	Purpose
`Router# show ip rsvp installed`	Displays information about interfaces and their admitted reservations.
`Router# show ip rsvp installed detail`	Displays additional information about interfaces and their admitted reservations.
`Router# show ip rsvp interface`	Displays RSVP-related interface information.
`Router# show ip rsvp interface detail`	Displays additional RSVP-related interface information.
`Router# show queueing [custom | fair | priority | random-detect [interface serial-number]]`	Displays all or selected configured queueing strategies and available bandwidth for RSVP reservations.

Configuration Examples

This section provides the following configuration examples:

- Configuring CBWFQ to Accommodate Reserved Traffic: Example
- Configuring the Resource Provider as None with Data Classification Turned Off: Example

Configuring CBWFQ to Accommodate Reserved Traffic: Example

The following output shows a class map and a policy map being configured for voice:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# class-map match-all voice
Router(config-cmap)# match access-group 100
Router(config-cmap)# exit
Router(config)# policy-map wfq-voip
Router(config-pmap)# class voice
Router(config-pmap-c)# priority 24
Router(config-pmap-c)# end
Router#
```

Note The bandwidth that you configured for the CBWFQ priority queue (24 kbps) must match the bandwidth that you configured for the interface. See the section "Enabling RSVP on an Interface".

The following output shows an access list being configured:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# access-list 100 permit udp any any range 16384 32500
```

The following output shows a class being applied to the outgoing interface:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# int atm6/0
Router(config-if)# service-policy output wfq-voip
```

The following output shows bandwidth being configured on an interface:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# int atm6/0
Router(config-if)# ip rsvp bandwidth 24
```

Note The bandwidth that you configure for the interface (24 kbps) must match the bandwidth that you configured for the CBWFQ priority queue.

Configuring the Resource Provider as None with Data Classification Turned Off: Example

The **show run** command displays the current configuration in the router:

```
Router# show run int atm6/0
 class-map match-all voice
  match access-group 100
!
policy-map wfq-voip
  class voice
    priority 24
  class class-default
   fair-queue
!
interface ATM6/0
 ip address 20.20.22.1 255.255.255.0
 no ip redirects
 no ip proxy-arp
 no ip route-cache cef
 atm uni-version 4.0
 atm pvc 1 0 5 qsaal
 atm pvc 2 0 16 ilmi
 atm esi-address 111111111181.00
 no atm auto-configuration
 no atm ilmi-keepalive
 pvc blue 200/100
  abr 700 600
  inarp 1
  broadcast
  encapsulation aal5snap
  service-policy output wfq-voip
 !
 ip rsvp bandwidth 24 24
 ip rsvp signalling dscp 48
access-list 100 permit udp any any range 16384 32500
```

Here is output from the **show ip rsvp interface detail** command before resource provider none is configured and data-packet classification is turned off:

```
Router# show ip rsvp interface detail
 AT6/0:
```

```
   Bandwidth:
     Curr allocated: 190K bits/sec
     Max. allowed (total): 112320K bits/sec
     Max. allowed (per flow): 112320K bits/sec
   Neighbors:
     Using IP encap: 1.  Using UDP encaps: 0
   DSCP value used in Path/Resv msgs: 0x30
```

Here is output from the **show queueing** command before resource provider none is configured and data packet classification is turned off:

```
Router# show queueing int atm6/0
  Interface ATM6/0 VC 200/100
  Queueing strategy: weighted fair
  Output queue: 63/512/64/3950945 (size/max total/threshold/drops)
     Conversations  2/5/64 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
     Available Bandwidth 450 kilobits/sec
```

Note New reservations do not reduce the available bandwidth (450 kilobits/sec shown above). Instead RSVP performs admission control only using the bandwidth limit configured in the **ip rsvp bandwidth** command. The bandwidth configured in this command should match the bandwidth configured in the CBWFQ class that you set up to handle the reserved traffic.

The following output shows resource provider none being configured:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# int atm6/0
Router(config-if)# ip rsvp resource-provider none
Router(config-if)# end
Router#
```

The following output shows data packet classification being turned off:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# int atm6/0
Router(config-if)# ip rsvp data-packet classification none
Router(config-if)# end
Router#
```

Here is output from the **show ip rsvp interface detail** command after resource provider none has been configured and data packet classification has been turned off:

```
Router# show ip rsvp interface detail
 AT6/0:
   Bandwidth:
     Curr allocated: 190K bits/sec
     Max. allowed (total): 112320K bits/sec
     Max. allowed (per flow): 112320K bits/sec
   Neighbors:
     Using IP encap: 1.  Using UDP encaps: 0
   DSCP value used in Path/Resv msgs: 0x30
   RSVP Data Packet Classification is OFF
   RSVP resource provider is: none
```

The following output from the **show ip rsvp installed detail** command verifies that resource provider none is configured and data packet classification is turned off:

```
Router# show ip rsvp installed detail
RSVP: ATM6/0 has the following installed reservations
RSVP Reservation. Destination is 145.20.20.212, Source is 145.10.10.211,
  Protocol is UDP, Destination port is 14, Source port is 14
  Reserved bandwidth: 50K bits/sec, Maximum burst: 1K bytes, Peak rate: 50K bits/sec
  Min Policed Unit: 0 bytes, Max Pkt Size: 1514 bytes
  Resource provider for this flow: None
  Conversation supports 1 reservations
  Data given reserved service: 3192 packets (1557696 bytes)
  Data given best-effort service: 42 packets (20496 bytes)
  Reserved traffic classified for 271 seconds
  Long-term average bitrate (bits/sec): 45880 reserved, 603 best-effort
RSVP Reservation. Destination is 145.20.20.212, Source is 145.10.10.211,
  Protocol is UDP, Destination port is 10, Source port is 10
  Reserved bandwidth: 20K bits/sec, Maximum burst: 1K bytes, Peak rate: 20K bits/sec
  Min Policed Unit: 0 bytes, Max Pkt Size: 1514 bytes
  Resource provider for this flow: None
  Conversation supports 1 reservations
  Data given reserved service: 1348 packets (657824 bytes)
  Data given best-effort service: 0 packets (0 bytes)
  Reserved traffic classified for 296 seconds
  Long-term average bitrate (bits/sec): 17755 reserved, 0M best-effort
```

The following output shows no increments in packet counts after the source sends data packets that match the reservation:

```
Router# show ip rsvp installed detail
RSVP: Ethernet3/3 has no installed reservations

RSVP: ATM6/0 has the following installed reservations
RSVP Reservation. Destination is 145.20.20.212, Source is 145.10.10.211,
  Protocol is UDP, Destination port is 14, Source port is 14
  Reserved bandwidth: 50K bits/sec, Maximum burst: 1K bytes, Peak rate: 50K bits/sec
  Min Policed Unit: 0 bytes, Max Pkt Size: 1514 bytes
  Resource provider for this flow: None
  Conversation supports 1 reservations
  Data given reserved service: 3192 packets (1557696 bytes)
  Data given best-effort service: 42 packets (20496 bytes)
  Reserved traffic classified for 282 seconds
  Long-term average bitrate (bits/sec): 44051 reserved, 579 best-effort
RSVP Reservation. Destination is 145.20.20.212, Source is 145.10.10.211,
  Protocol is UDP, Destination port is 10, Source port is 10
  Reserved bandwidth: 20K bits/sec, Maximum burst: 1K bytes, Peak rate: 20K bits/sec
  Min Policed Unit: 0 bytes, Max Pkt Size: 1514 bytes
  Resource provider for this flow: None
  Conversation supports 1 reservations
  Data given reserved service: 1348 packets (657824 bytes)
  Data given best-effort service: 0 packets (0 bytes)
  Reserved traffic classified for 307 seconds
  Long-term average bitrate (bits/sec): 17121 reserved, 0M best-effort
```

The following output shows that data packet classification is enabled again:

```
Router# configure terminal
Router(config)# int atm6/0
Router(config-if) no ip rsvp data-packet classification
Router(config-if)# end
```

The following output verifies that data packet classification is occurring:

```
Router# show ip rsvp installed detail
Enter configuration commands, one per line.  End with CNTL/Z.
RSVP: ATM6/0 has the following installed reservations
RSVP Reservation. Destination is 145.20.20.212, Source is 145.10.10.211,
  Protocol is UDP, Destination port is 14, Source port is 14
  Reserved bandwidth: 50K bits/sec, Maximum burst: 1K bytes, Peak rate: 50K bits/sec
  Min Policed Unit: 0 bytes, Max Pkt Size: 1514 bytes
  Resource provider for this flow: None
  Conversation supports 1 reservations
  Data given reserved service: 3683 packets (1797304 bytes)
  Data given best-effort service: 47 packets (22936 bytes)
  Reserved traffic classified for 340 seconds
  Long-term average bitrate (bits/sec): 42201 reserved, 538 best-effort
RSVP Reservation. Destination is 145.20.20.212, Source is 145.10.10.211,
  Protocol is UDP, Destination port is 10, Source port is 10
  Reserved bandwidth: 20K bits/sec, Maximum burst: 1K bytes, Peak rate: 20K bits/sec
  Min Policed Unit: 0 bytes, Max Pkt Size: 1514 bytes
  Resource provider for this flow: None
  Conversation supports 1 reservations
  Data given reserved service: 1556 packets (759328 bytes)
  Data given best-effort service: 0 packets (0 bytes)
  Reserved traffic classified for 364 seconds
  Long-term average bitrate (bits/sec): 16643 reserved, 0M best-effort
```

Here is output from the **show run** command after you have performed all the previous configuration tasks:

```
Router# show run int atm6/0
 class-map match-all voice
  match access-group 100
!
policy-map wfq-voip
  class voice
    priority 24
  class class-default
   fair-queue
!
interface ATM6/0
 ip address 20.20.22.1 255.255.255.0
 no ip redirects
 no ip proxy-arp
 no ip route-cache cef
 atm uni-version 4.0
 atm pvc 1 0 5 qsaal
 atm pvc 2 0 16 ilmi
 atm esi-address 111111111181.00
 no atm auto-configuration
 no atm ilmi-keepalive
 pvc blue 200/100
  abr 700 600
  inarp 1
  broadcast
  encapsulation aal5snap
  service-policy output wfq-voip
 !
 ip rsvp bandwidth 24 24
 ip rsvp signalling dscp 48
 ip rsvp data-packet classification none
 ip rsvp resource-provider none

access-list 100 permit udp any any range 16384 32500
```

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Quality of Service Solutions Command Reference* at http://www.cisco.com/en/US/docs/ios/qos/command/reference/qos_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

- **debug ip rsvp traffic-control**
- **debug ip rsvp wfq**
- **ip rsvp data-packet classification none**
- **ip rsvp resource-provider**
- **show ip rsvp installed**
- **show ip rsvp interface**
- **show queueing**

Note You can use **debug ip rsvp traffic-control** and **debug ip rsvp wfq** simultaneously. Use the **show debug** command to see which debugging commands are enabled.

Glossary

admission control—The process in which an RSVP reservation is accepted or rejected based on end-to-end available network resources.

aggregate—A collection of packets with the same DSCP.

bandwidth—The difference between the highest and lowest frequencies available for network signals. This term also describes the rated throughput capacity of a given network medium or protocol.

CBWFQ—Class-based weighted fair queueing. A queueing mechanism that extends the standard WFQ functionality to provide support for user-defined traffic classes.

Class-based weighted fair queueing—See CBWFQ.

differentiated services—See DiffServ.

differentiated services code point—See DSCP.

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

DSCP—Differentiated services code point. The six most significant bits of the 1-byte IP type of service (ToS) field. The per-hop behavior represented by a particular DSCP value is configurable. DSCP values range between 0 and 63.

enterprise network—A large and diverse network connecting most major points in a company or other organization.

flow—A stream of data traveling between two endpoints across a network (for example, from one LAN station to another). Multiple flows can be transmitted on a single circuit.

packet—A logical grouping of information that includes a header containing control information and (usually) user data. Packets most often refer to network layer units of data.

PBX—Private branch exchange. A digital or analog telephone switchboard located on the subscriber premises and used to connect private and public telephone networks.

PHB—Per hop behavior. A DiffServ concept that specifies how specifically marked packets are to be treated by each DiffServ router.

QoS—Quality of service. A measure of performance for a transmission system that reflects its transmission quality and service availability.

quality of service—See QoS.

Resource Reservation Protocol—See RSVP.

RSVP—Resource Reservation Protocol. A protocol for reserving network resources to provide quality of service guarantees to application flows.

Voice over IP—See VoIP.

VoIP—Voice over IP. The ability to carry normal telephony-style voice over an IP-based internet maintaining telephone-like functionality, reliability, and voice quality.

Weighted Fair Queueing—See WFQ.

WFQ—Weighted fair queueing. A queue management algorithm that provides a certain fraction of link bandwidth to each of several queues, based on relative bandwidth applied to each of the queues.

RSVP Refresh Reduction and Reliable Messaging

First Published: November 25, 2002
Last Updated: February 27, 2009

The RSVP Refresh Reduction and Reliable Messaging feature includes refresh reduction, which improves the scalability, latency, and reliability of Resource Reservation Protocol (RSVP) signaling to enhance network performance and message delivery.

History for the RSVP Refresh Reduction and Reliable Messaging Feature

Release	Modification
12.2(13)T	This feature was introduced.
12.0(24)S	This feature was integrated into Cisco IOS Release 12.0(24)S.
12.2(14)S	This feature was integrated into Cisco IOS Release 12.2(14)S.
12.0(26)S	Two commands, **ip rsvp signalling refresh misses** and **ip rsvp signalling refresh interval**, were added into Cisco IOS Release 12.0(26)S.
12.0(29)S	The *burst* and *max-size* argument defaults for the **ip rsvp signalling rate-limit** command were increased to 8 messages and 2000 bytes, respectively.
12.2(28)SB	This feature was integrated into Cisco IOS Release 12.2(28)SB.
12.2(18)SXF5	This feature was integrated into Cisco IOS Release 12.2(18)SXF5.
12.2(33)SRB	This feature was integrated into Cisco IOS Release 12.2(33)SRB.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Contents

Prerequisites for RSVP Refresh Reduction and Reliable Messaging

RSVP must be configured on two or more routers within the network before you can use the RSVP Refresh Reduction and Reliable Messaging feature.

Restrictions for RSVP Refresh Reduction and Reliable Messaging

Multicast flows are not supported for the reliable messages and summary refresh features.

Information About RSVP Refresh Reduction and Reliable Messaging

To configure the RSVP Refresh Reduction and Reliable Messaging feature, you should understand the following concepts:

Feature Design of RSVP Refresh Reduction and Reliable Messaging

RSVP is a network-control, soft-state protocol that enables Internet applications to obtain special qualities of service (QoS) for their data flows. As a soft-state protocol, RSVP requires that state be periodically refreshed. If refresh messages are not transmitted during a specified interval, RSVP state automatically times out and is deleted.

In a network that uses RSVP signaling, reliability and latency problems occur when an RSVP message is lost in transmission. A lost RSVP setup message can cause a delayed or failed reservation; a lost RSVP refresh message can cause a delay in the modification of a reservation or in a reservation timeout. Intolerant applications can fail as a result.

Reliability problems can also occur when there is excessive RSVP refresh message traffic caused by a large number of reservations in the network. Using summary refresh messages can improve reliability by significantly reducing the amount of RSVP refresh traffic.

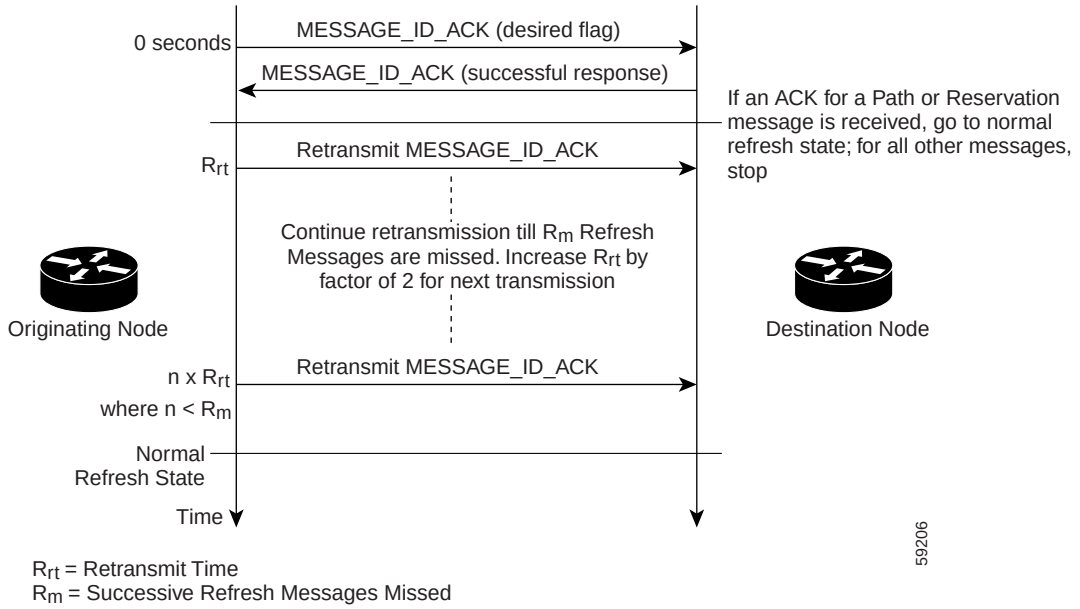
Note RSVP packets consist of headers that identify the types of messages, and object fields that contain attributes and properties describing how to interpret and act on the content.

Types of Messages in RSVP Refresh Reduction and Reliable Messaging

The RSVP Refresh Reduction and Reliable Messaging feature (Figure 1) includes refresh reduction, which improves the scalability, latency, and reliability of RSVP signaling by introducing the following extensions:

- Reliable messages (MESSAGE_ID, MESSAGE_ID_ACK objects, and ACK messages)
- Bundle messages (reception and processing only)
- Summary refresh messages (MESSAGE_ID_LIST and MESSAGE_ID_NACK objects)

Figure 1 RSVP Refresh Reduction and Reliable Messaging



Reliable Messages

The reliable messages extension supports dependable message delivery among neighboring routers by implementing an acknowledgment mechanism that consists of a MESSAGE_ID object and a MESSAGE_ID_ACK object. The acknowledgments can be transmitted in an ACK message or piggybacked in other RSVP messages.

Each RSVP message contains one MESSAGE_ID object. If the ACK_Desired flag field is set within the MESSAGE_ID object, the receiver transmits a MESSAGE_ID_ACK object to the sender to confirm delivery.

Bundle Messages

A bundle message consists of several standard RSVP messages that are grouped into a single RSVP message.

A bundle message must contain at least one submessage. A submessage can be any RSVP message type other than another bundle message. Submessage types include Path, PathErr, Resv, ResvTear, ResvErr, ResvConf, and ACK.

Bundle messages are addressed directly to the RSVP neighbor. The bundle header immediately follows the IP header, and there is no intermediate transport header.

When a router receives a bundle message that is not addressed to one of its local IP addresses, it forwards the message.

Note Bundle messages can be received, but not sent.

Summary Refresh Messages

A summary refresh message supports the refreshing of RSVP state without the transmission of conventional Path and Resv messages. Therefore, the amount of information that must be transmitted and processed to maintain RSVP state synchronization is greatly reduced.

A summary refresh message carries a set of MESSAGE_ID objects that identify the Path and Resv states that should be refreshed. When an RSVP node receives a summary refresh message, the node matches each received MESSAGE_ID object with the locally installed Path or Resv state. If the MESSAGE_ID objects match the local state, the state is updated as if a standard RSVP refresh message were received. However, if a MESSAGE_ID object does not match the receiver's local state, the receiver notifies the sender of the summary refresh message by transmitting a MESSAGE_ID_NACK object.

When a summary refresh message is used to refresh the state of an RSVP session, the transmission of conventional refresh messages is suppressed. The summary refresh extension cannot be used for a Path or Resv message that contains changes to a previously advertised state. Also, only a state that was previously advertised in Path or Resv messages containing MESSAGE_ID objects can be refreshed by using a summary refresh message.

Benefits of RSVP Refresh Reduction and Reliable Messaging

Enhanced Network Performance

Refresh reduction reduces the volume of steady-state network traffic generated, the amount of CPU resources used, and the response time, thereby enhancing network performance.

Improved Message Delivery

The MESSAGE_ID and the MESSAGE_ID_ACK objects ensure the reliable delivery of messages and support rapid state refresh when a network problem occurs. For example, MESSAGE_ID_ACK objects are used to detect link transmission losses.

How to Configure RSVP Refresh Reduction and Reliable Messaging

This section contains the following procedures:

- Enabling RSVP on an Interface, page 5 (required)
- Enabling RSVP Refresh Reduction, page 6 (required)
- Verifying RSVP Refresh Reduction and Reliable Messaging, page 7 (optional)

Enabling RSVP on an Interface

Perform the following task to enable RSVP on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip rsvp bandwidth** [*interface-kbps* [*sub-pool*]]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`interface` *type number* **Example:** `Router(config)# interface Ethernet1`	Enters interface configuration mode. • The *type* and *number* arguments identify the interface to be configured.

	Command or Action	Purpose
Step 4	**ip rsvp bandwidth** [*interface-kbps* [*sub-pool*]] **Example:** `Router(config-if)# ip rsvp bandwidth 7500 7500`	Enables RSVP on an interface. • The optional *interface-kbps* and *sub-pool* arguments specify the amount of bandwidth that can be allocated by RSVP flows or to a single flow, respectively. Values are from 1 to 10000000, and from 0 to 10000000, respectively.
Step 5	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Enabling RSVP Refresh Reduction

Perform the following task to enable RSVP refresh reduction.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp signalling refresh reduction**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp signalling refresh reduction** **Example:** `Router(config)# ip rsvp signalling refresh reduction`	Enables refresh reduction.
Step 4	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.

Verifying RSVP Refresh Reduction and Reliable Messaging

Perform the following task to verify that the RSVP Refresh Reduction and Reliable Messaging feature is functioning.

SUMMARY STEPS

1. **enable**
2. **clear ip rsvp counters [confirm]**
3. **show ip rsvp**
4. **show ip rsvp counters [interface** *interface-unit* **| summary | neighbor]**
5. **show ip rsvp interface [***interface-type interface-number***] [detail**]
6. **show ip rsvp neighbor** [**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ip rsvp counters** [**confirm**] **Example:** `Router# clear ip rsvp counters`	(Optional) Clears (sets to zero) all IP RSVP counters that are being maintained by the router.
Step 3	**show ip rsvp** **Example:** `Router# show ip rsvp`	(Optional) Displays RSVP rate-limiting, refresh-reduction, and neighbor information.
Step 4	**show ip rsvp counters [interface** *interface-unit* **| summary | neighbor]** **Example:** `Router# show ip rsvp counters summary`	(Optional) Displays the number of RSVP messages that were sent and received on each interface. • The optional **summary** keyword displays the cumulative number of RSVP messages sent and received by the router over all interfaces.
Step 5	**show ip rsvp interface [***interface-type interface-number***] [detail]** **Example:** `Router# show ip rsvp interface detail`	(Optional) Displays information about interfaces on which RSVP is enabled including the current allocation budget and maximum available bandwidth. • The optional **detail** keyword displays the bandwidth and signaling parameters.
Step 6	**show ip rsvp neighbor [detail]** **Example:** `Router# show ip rsvp neighbor detail`	(Optional) Displays RSVP-neighbor information including IP addresses. • The optional **detail** keyword displays the current RSVP neighbors and identifies if the neighbor is using IP, User Datagram Protocol (UDP), or RSVP encapsulation for a specified interface or all interfaces.

Configuration Examples for RSVP Refresh Reduction and Reliable Messaging

This section provides the following configuration example:

RSVP Refresh Reduction and Reliable Messaging: Example

In the following example, RSVP refresh reduction is enabled:

```
Router# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface Ethernet1
Router(config-if)# ip rsvp bandwidth 7500 7500
Router(config-if)# exit
Router(config)# ip rsvp signalling refresh reduction
Router(config)# end
```

The following example verifies that RSVP refresh reduction is enabled:

```
Router# show running-config

Building configuration...
Current configuration : 1503 bytes

!
version 12.2
no service single-slot-reload-enable
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
!
hostname Router
!
no logging buffered
logging rate-limit console 10 except errors
!
ip subnet-zero
ip cef
!
ip multicast-routing
no ip dhcp-client network-discovery
lcp max-session-starts 0
mpls traffic-eng tunnels
!
!
interface Loopback0
 ip address 192.168.1.1 255.255.255.0
 ip rsvp bandwidth 1705033 1705033
!
interface Tunnel777
 no ip address
 shutdown
!
```

```
interface Ethernet0
 ip address 192.168.0.195 255.0.0.0
 no ip mroute-cache
 media-type 10BaseT
!
interface Ethernet1
 ip address 192.168.5.2 255.255.255.0
 no ip redirects
 no ip proxy-arp
 ip pim dense-mode
 no ip mroute-cache
 media-type 10BaseT
 ip rsvp bandwidth 7500 7500
!
interface Ethernet2
 ip address 192.168.1.2 255.255.255.0
 no ip redirects
 no ip proxy-arp
 ip pim dense-mode
 no ip mroute-cache
 media-type 10BaseT
 mpls traffic-eng tunnels
 ip rsvp bandwidth 7500 7500
!
interface Ethernet3
 ip address 192.168.2.2 255.255.255.0
 ip pim dense-mode
 media-type 10BaseT
 mpls traffic-eng tunnels
!
!
router eigrp 17
 network 192.168.0.0
 network 192.168.5.0
 network 192.168.12.0
 network 192.168.30.0
 auto-summary
 no eigrp log-neighbor-changes
!
ip classless
no ip http server
ip rsvp signalling refresh reduction
!
!
!
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 login
 transport input pad v120 telnet rlogin udptn
!
end
```

Additional References

The following sections provide references related to the RSVP Refresh Reduction and Reliable Messaging feature.

Related Documents

Related Topic	Document Title
RSVP commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
QoS features including signaling, classification, and congestion management	"Quality of Service Overview" module

Standards

Standard	Title
None	—

MIBs

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

RFCs

RFC	Title
RFC 2205	*Resource Reservation Protocol*
RFC 2206	*RSVP Management Information Base Using SMIv2*
RFC 2209	*RSVP—Version 1 Message Processing Rules*
RFC 2210	*The Use of RSVP with IETF Integrated Services*
RFC 2211/2212	*Specification of the Controlled-Load Network Element Service*
RFC 2702	*Requirements for Traffic Engineering over MPLS*
RFC 2749	*Common Open Policy Service (COPS) Usage for RSVP*
RFC 2750	*RSVP Extensions for Policy Control*
RFC 2814	*SBM Subnet Bandwidth Manager: A Protocol for RSVP-based Admission Control over IEEE 802-style Networks*

RFC	Title
RFC 2961	*RSVP Refresh Overhead Reduction Extensions*
RFC 2996	*Format of the RSVP DCLASS Object*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Quality of Service Solutions Command Reference* at http://www.cisco.com/en/US/docs/ios/qos/command/reference/qos_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

- **ip rsvp signalling rate-limit**
- **show ip rsvp signalling rate-limit**

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

RSVP Message Authentication

First Published: March 17, 2003
Last Updated: August 6, 2007

The Resource Reservation Protocol (RSVP) Message Authentication feature provides a secure method to control quality of service (QoS) access to a network.

History for the RSVP Message Authentication Feature

Release	Modification
12.2(15)T	This feature was introduced.
12.0(26)S	Restrictions were added for interfaces that use Fast Reroute (FRR) node or link protection and for RSVP hellos for FRR for packet over SONET (POS) interfaces.
12.0(29)S	Support was added for per-neighbor keys.
12.2(33)SRA	This feature was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This feature was integrated into Cisco IOS Release 12.2(33)SXH.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for RSVP Message Authentication

Ensure that RSVP is configured on one or more interfaces on at least two neighboring routers that share a link within the network.

Restrictions for RSVP Message Authentication

- The RSVP Message Authentication feature is only for authenticating RSVP neighbors.
- The RSVP Message Authentication feature cannot discriminate between various QoS applications or users, of which many may exist on an authenticated RSVP neighbor.
- Different send and accept lifetimes for the same key in a specific key chain are not supported; all RSVP key types are bidirectional.
- Authentication for graceful restart hello messages is supported for per-neighbor and per-access control list (ACL) keys, but not for per-interface keys.
- You cannot use the **ip rsvp authentication key** and the **ip rsvp authentication key-chain** commands on the same router interface.
- For a Multiprotocol Label Switching/Traffic Engineering (MPLS/TE) configuration, use per-neighbor keys with physical addresses and router IDs.

Information About RSVP Message Authentication

To configure RSVP Message Authentication, you need to understand the following concepts:

Feature Design of RSVP Message Authentication

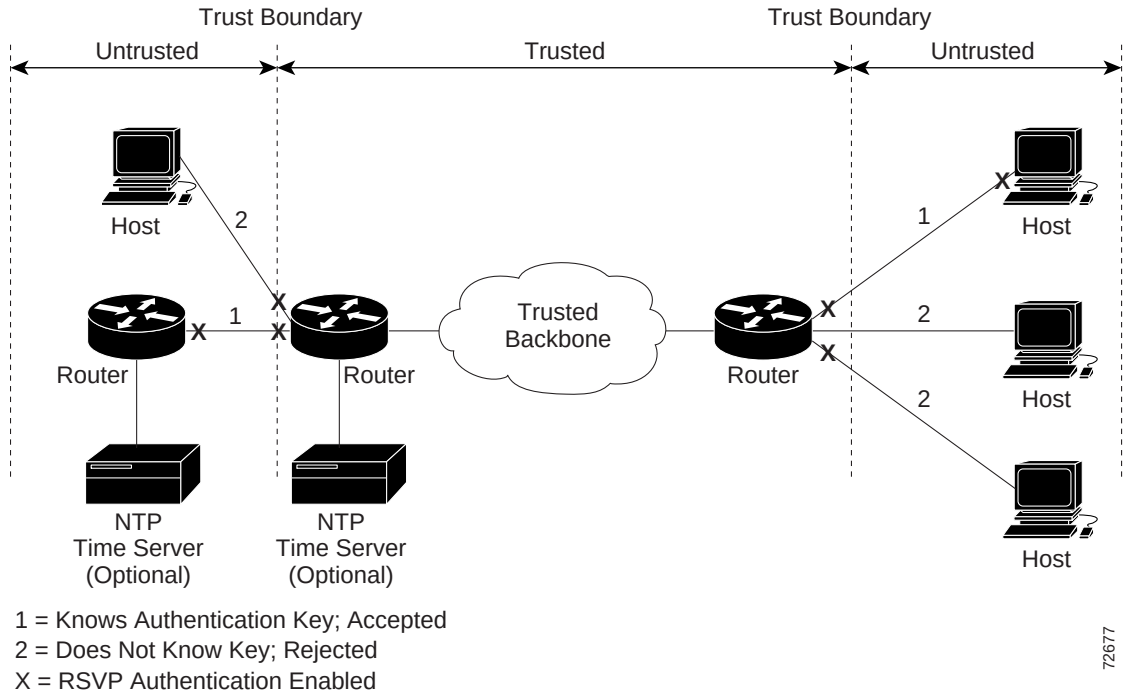
Network administrators need the ability to establish a security domain to control the set of systems that initiate RSVP requests.

The RSVP Message Authentication feature permits neighbors in an RSVP network to use a secure hash to sign all RSVP signaling messages digitally, thus allowing the receiver of an RSVP message to verify the sender of the message without relying solely on the sender's IP address as is done by issuing the **ip rsvp neighbor** command with an ACL.

The signature is accomplished on a per-RSVP-hop basis with an RSVP integrity object in the RSVP message as defined in RFC 2747. This method provides protection against forgery or message modification. However, the receiver must know the security key used by the sender in order to validate the digital signature in the received RSVP message.

Network administrators manually configure a common key for each RSVP neighbor interface on the shared network. A sample configuration is shown in Figure 1.

Figure 1 RSVP Message Authentication Configuration



Global Authentication and Parameter Inheritance

You can configure global defaults for all authentication parameters including key, type, window size, lifetime, and challenge. These defaults are inherited when you enable authentication for each neighbor or interface. However, you can also configure these parameters individually on a per-neighbor or per-interface basis in which case the inherited global defaults are ignored.

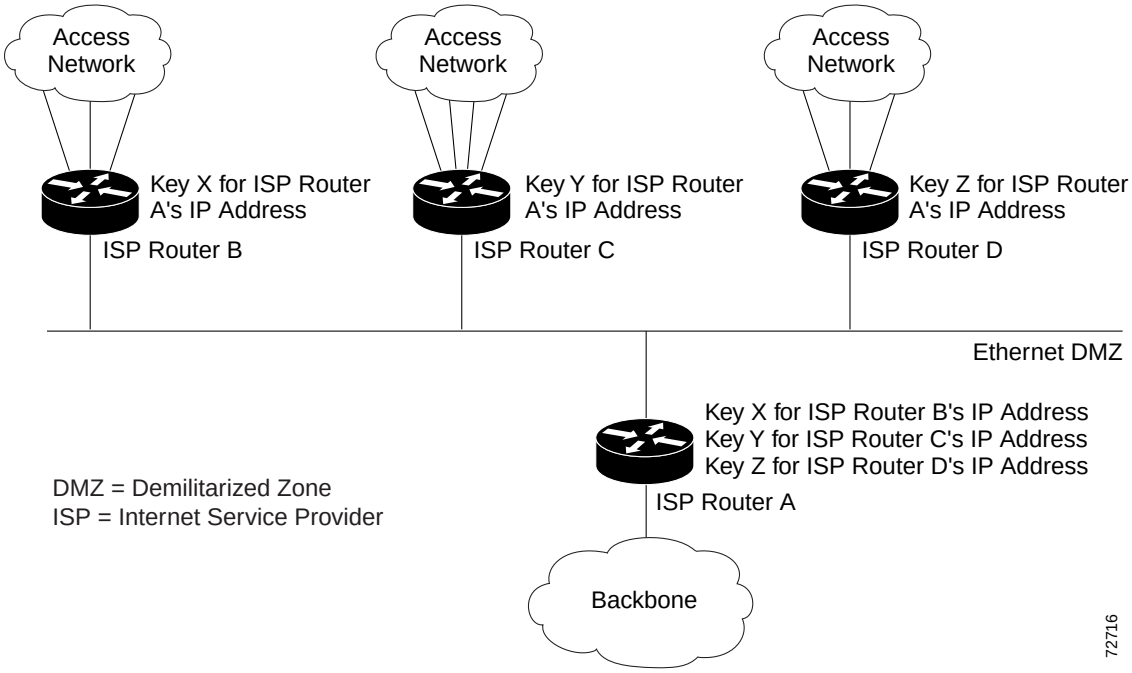
Using global authentication and parameter inheritance can simplify configuration because you can enable or disable authentication without having to change each per-neighbor or per-interface attribute. You can activate authentication for all neighbors by using two commands, one to define a global default key and one to enable authentication globally. However, using the same key for all neighbors does not provide the best network security.

Note RSVP uses the following rules when choosing which authentication parameter to use when that parameter is configured at multiple levels (per-interface, per-neighbor, or global). RSVP goes from the most specific to the least specific; that is, per-neighbor, per-interface, and then global. The rules are slightly different when searching the configuration for the right key to authenticate an RSVP message—per-neighbor, per-ACL, per-interface, and then global.

Per-Neighbor Keys

In Figure 2, to enable authentication between Internet service provider (ISP) Routers A and B, A and C, and A and D, the ISPs must share a common key. However, sharing a common key also enables authentication between ISP Routers B and C, C and D, and B and D. You may not want authentication among all the ISPs because they might be different companies with unique security domains Figure 2.

Figure 2 RSVP Message Authentication in an Ethernet Configuration



On ISP Router A, you create a different key for ISP Routers B, C, and D and assign them to their respective IP addresses using RSVP commands. On the other routers, create a key to communicate with ISP Router A's IP address.

Key Chains

For each RSVP neighbor, you can configure a list of keys with specific IDs that are unique and have different lifetimes so that keys can be changed at predetermined intervals automatically without any disruption of service. Automatic key rotation enhances network security by minimizing the problems that could result if an untrusted source obtained, deduced, or guessed the current key.

Note If you use overlapping time windows for your key lifetimes, RSVP asks the Cisco IOS software key manager component for the next live key starting at time T. The key manager walks the keys in the chain until it finds the first one with start time S and end time E such that S <= T <= E. Therefore, the key with the smallest value (E–T) may not be used next.

Benefits of RSVP Message Authentication

Improved Security

The RSVP Message Authentication feature greatly reduces the chance of an RSVP-based spoofing attack and provides a secure method to control QoS access to a network.

Multiple Environments

The RSVP Message Authentication feature can be used in traffic engineering (TE) and non-TE environments as well as with the subnetwork bandwidth manager (SBM).

Multiple Platforms and Interfaces

The RSVP Message Authentication feature can be used on any supported RSVP platform or interface.

How to Configure RSVP Message Authentication

The following configuration parameters instruct RSVP on how to generate and verify integrity objects in various RSVP messages.

Note There are two configuration procedures: full and minimal. There are also two types of authentication procedures: interface and neighbor.

Per-Interface Authentication—Full Configuration

Perform the following procedures for a full configuration for per-interface authentication:

- Enabling RSVP on an Interface, page 6 (required)
- Configuring an RSVP Authentication Type, page 7 (optional)
- Configuring an RSVP Authentication Key, page 8 (required)
- Enabling RSVP Key Encryption, page 10 (optional)
- Enabling RSVP Authentication Challenge, page 11 (optional)
- Configuring RSVP Authentication Lifetime, page 12 (optional)
- Configuring RSVP Authentication Window Size, page 13 (optional)
- Activating RSVP Authentication, page 15 (required)
- Verifying RSVP Message Authentication, page 16 (optional)

Per-Interface Authentication—Minimal Configuration

Perform the following tasks for a minimal configuration for per-interface authentication:

- Enabling RSVP on an Interface, page 6 (required)
- Configuring an RSVP Authentication Key, page 8 (required)
- Activating RSVP Authentication, page 15 (required)

Per-Neighbor Authentication—Full Configuration

Perform the following procedures for a full configuration for per-neighbor authentication:

- Configuring an RSVP Authentication Type, page 7 (optional)
- Enabling RSVP Authentication Challenge, page 11 (optional)
- Enabling RSVP Key Encryption, page 10 (optional)
- Configuring RSVP Authentication Lifetime, page 12 (optional)
- Configuring RSVP Authentication Window Size, page 13 (optional)
- Activating RSVP Authentication, page 15 (required)
- Verifying RSVP Message Authentication, page 16 (optional)
- Configuring a Key Chain, page 17 (required)
- Binding a Key Chain to an RSVP Neighbor, page 18 (required)

Per-Neighbor Authentication—Minimal Configuration

Perform the following tasks for a minimal configuration for per-neighbor authentication:

- Activating RSVP Authentication, page 15 (required)
- Configuring a Key Chain, page 17 (required)
- Binding a Key Chain to an RSVP Neighbor, page 18 (required)

Enabling RSVP on an Interface

Perform this task to enable RSVP on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip rsvp bandwidth** [*interface-kbps [single-flow-kbps]]*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet0/0`	Enters interface configuration mode. • The *type number* argument identifies the interface to be configured.
Step 4	`ip rsvp bandwidth [`*interface-kbps* `[`*single-flow-kbps*`]]` **Example:** `Router(config-if)# ip rsvp bandwidth 7500 7500`	Enables RSVP on an interface. • The optional *interface-kbps* and *single-flow-kbps* arguments specify the amount of bandwidth that can be allocated by RSVP flows or to a single flow, respectively. Values are from 1 to 10,000,000. **Note** Repeat this command for each interface that you want to enable.
Step 5	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Configuring an RSVP Authentication Type

Perform this task to configure an RSVP authentication type.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip rsvp authentication type** {**md5** | **sha-1**}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet0/0`	Enters interface configuration mode. • The *type number* argument identifies the interface to be configured. **Note** Omit this step if you are configuring an authentication type for a neighbor or setting a global default.
Step 4	**ip rsvp authentication type** {**md5 | sha-1**} **Example:** For interface authentication: `Router(config-if)# ip rsvp authentication type sha-1` For neighbor authentication: `Router(config)# ip rsvp authentication neighbor address 10.1.1.1 type sha-1` or `Router(config)# ip rsvp authentication neighbor access-list 1 type sha-1` For a global default: `Router(config)# ip rsvp authentication type sha-1`	Specifies the algorithm used to generate cryptographic signatures in RSVP messages on an interface or globally. • The algorithms are **md5**, the default, and **sha-1**, which is newer and more secure than **md5**. **Note** Omit the **neighbor address** *address* or the **neighbor access-list** *acl-name* or *acl-number* to set the global default.
Step 5	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Configuring an RSVP Authentication Key

Perform this task to configure an RSVP authentication key.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **interface** *type number*
4. **ip rsvp authentication key** ***passphrase***
5. **exit**
6. **ip rsvp authentication key-chain** *chain*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode. **Note** If you want to configure a key, proceed to Step 3; if you want to configure a key chain, proceed to Step 6.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet0/0`	Enters interface configuration mode. • The *type number* argument identifies the interface to be configured. **Note** Omit this step and go to Step 6 if you want to configure only a key chain.
Step 4	**ip rsvp authentication key** ***passphrase*** **Example:** `Router(config-if)# ip rsvp authentication key 11223344`	Specifies the data string (key) for the authentication algorithm. • The key consists of 8 to 40 characters. It can include spaces and multiple words. It can also be encrypted or appear in clear text when displayed. **Note** Omit this step if you want to configure a key chain.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits to global configuration mode.

	Command or Action	Purpose
Step 6	**ip rsvp authentication key-chain** *chain* **Example:** For neighbor authentication: `Router(config)# ip rsvp authentication neighbor address 10.1.1.1 key-chain xzy` or `Router(config)# ip rsvp authentication neighbor access-list 1 key-chain xzy` For a global default: `Router(config)# ip rsvp authentication key-chain xzy`	Specifies the data string (key chain) for the authentication algorithm. • The key chain must have at least one key, but can have up to 2,147,483647 keys. **Note** You cannot use the **ip rsvp authentication key** and the **ip rsvp authentication key-chain** commands on the same router interface. The commands supersede each other; however, no error message is generated. **Note** Omit the **neighbor address** *address* or the **neighbor access-list** *acl-name* or *acl-number* to set the global default.
Step 7	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.

Enabling RSVP Key Encryption

Perform this task to enable RSVP key encryption when the key is stored in the router configuration. (This prevents anyone from seeing the clear text key in the configuration file.)

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **key config-key 1** *string*
4. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**key config-key 1** *string* **Example:** `Router(config)# key config-key 1 11223344`	Enables key encryption in the configuration file. **Note** The *string* argument can contain up to eight alphanumeric characters.
Step 4	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.

Enabling RSVP Authentication Challenge

Perform this task to enable RSVP authentication challenge.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip rsvp authentication challenge**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet0/0`	Enters interface configuration mode. • The *type number* argument identifies the interface to be configured. **Note** Omit this step if you are configuring an authentication challenge for a neighbor or setting a global default.

	Command or Action	Purpose
Step 4	**ip rsvp authentication challenge** **Example:** For interface authentication: `Router(config-if)# ip rsvp authentication challenge` For neighbor authentication: `Router(config)# ip rsvp authentication neighbor address 10.1.1.1 challenge` or `Router(config)# ip rsvp authentication neighbor access-list 1 challenge` For a global default: `Router(config)# ip rsvp authentication challenge`	Makes RSVP perform a challenge-response handshake on an interface or globally when RSVP learns about any new challenge-capable neighbors on a network. **Note** Omit the **neighbor address** *address* or the **neighbor access-list** *acl-name* or *acl-number* to set the global default.
Step 5	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Configuring RSVP Authentication Lifetime

Perform this task to configure the lifetimes of security associations between RSVP neighbors.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip rsvp authentication lifetime** *hh*:*mm*:*ss*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet0/0` **Note** Omit this step if you are configuring an authentication lifetime for a neighbor or setting a global default.	Enters interface configuration mode. • The *type number* argument identifies the interface to be configured.
Step 4	**ip rsvp authentication lifetime** *hh*:*mm*:*ss* **Example:** For interface authentication: `Router(config-if)# ip rsvp authentication lifetime 00:05:00` For neighbor authentication: `Router(config)# ip rsvp authentication neighbor address 10.1.1.1 lifetime 00:05:00` or `Router(config)# ip rsvp authentication neighbor access-list 1 lifetime 00:05:00` For a global default: `Router(config)# ip rsvp authentication 00:05:00`	Controls how long RSVP maintains security associations with RSVP neighbors on an interface or globally. • The default security association for **hh:mm:ss** is 30 minutes; the range is 1 second to 24 hours. **Note** Omit the **neighbor address** *address* or the **neighbor access-list** *acl-name* or *acl-number* to set the global default.
Step 5	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Configuring RSVP Authentication Window Size

Perform this task to configure the RSVP authentication window size.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **interface** *type number*
4. **ip rsvp authentication window-size** ***n***
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet0/0`	Enters interface configuration mode. • The *type number* argument identifies the interface to be configured. **Note** Omit this step if you are configuring a window size for a neighbor or setting a global default.
Step 4	**ip rsvp authentication window-size** ***n*** **Example:** For interface authentication: `Router(config-if)# ip rsvp authentication window-size 2` For neighbor authentication: `Router(config)# ip rsvp authentication neighbor address 10.1.1.1 window-size 2` or `Router(config)# ip rsvp authentication neighbor access-list 1 window-size` For a global default: `Router(config)# ip rsvp authentication window-size 2`	Specifies the maximum number of authenticated messages that can be received out of order on an interface or globally. • The default value is one message; the range is 1 to 64 messages. **Note** Omit the **neighbor address** *address* or the **neighbor access-list** *acl-name* or *acl-number* to set the global default.
Step 5	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Activating RSVP Authentication

Perform this task to activate RSVP authentication.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip rsvp authentication**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`interface` *type number* **Example:** `Router(config)# interface Ethernet0/0`	Enters interface configuration mode. • The *type number* argument identifies the interface to be configured. **Note** Omit this step if you are configuring authentication for a neighbor or setting a global default.

	Command or Action	Purpose
Step 4	**ip rsvp authentication** **Example:** For interface authentication: `Router(config-if)# ip rsvp authentication` For neighbor authentication: `Router(config)# ip rsvp authentication neighbor address 10.1.1.1` or `Router(config)# ip rsvp authentication neighbor access-list 1` For a global default: `Router(config)# ip rsvp authentication`	Activates RSVP cryptographic authentication on an interface or globally. **Note** Omit the **neighbor address** *address* or the **neighbor access-list** *acl-name* or *acl-number* to set the global default.
Step 5	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Verifying RSVP Message Authentication

Perform this task to verify that the RSVP Message Authentication feature is functioning.

SUMMARY STEPS

1. **enable**
2. **show ip rsvp interface [detail] [*interface-type interface-number*]**
3. **show ip rsvp authentication** [**detail**] [**from** {*ip-address* | *hostname*}] [**to** {*ip-address* | *hostname*}]
4. **show ip rsvp counters** [**authentication** | **interface** *interface-unit* | **neighbor** | **summary**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip rsvp interface [detail] [*interface-type interface-number*]** **Example:** `Router# show ip rsvp interface detail`	Displays information about interfaces on which RSVP is enabled, including the current allocation budget and maximum available bandwidth. • The optional **detail** keyword displays the bandwidth, signaling, and authentication parameters.

	Command or Action	Purpose
Step 3	**show ip rsvp authentication** [**detail**] [**from** {*ip-address* | *hostname*}] [**to** {*ip-address* | *hostname*}] **Example:** `Router# show ip rsvp authentication detail`	Displays the security associations that RSVP has established with other RSVP neighbors. • The optional **detail** keyword displays state information that includes IP addresses, interfaces enabled, and configured cryptographic authentication parameters about security associations that RSVP has established with neighbors.
Step 4	**show ip rsvp counters** [**authentication** | **interface** *interface-unit* | **neighbor** | **summary**] **Example:** `Router# show ip rsvp counters summary` `Router# show ip rsvp counters authentication`	Displays all RSVP counters. **Note** The errors counter increments whenever an authentication error occurs, but can also increment for errors not related to authentication. • The optional **authentication** keyword shows a list of RSVP authentication counters. • The optional **interface** *interface-unit* keyword argument combination shows the number of RSVP messages sent and received by the specific interface. • The optional **neighbor** keyword shows the number of RSVP messages sent and received by the specific neighbor. • The optional **summary** keyword shows the cumulative number of RSVP messages sent and received by the router. It does not print per-interface counters.

Configuring a Key Chain

Perform this task to configure a key chain for neighbor authentication.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **key chain** *name-of-chain*
4. {**key** [*key-ID*] | **key-string** [*text*] | **accept-lifetime** [*start-time* {**infinite** | *end-time* | **duration** *seconds*}] | **send-lifetime** [*start-time* {**infinite** | *end-time* | **duration** *seconds*}]}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**key chain** *name-of-chain* **Example:** `Router(config)# key chain neighbor_V`	Enters key-chain mode.
Step 4	{**key** [*key-ID*] | **key-string** [*text*] | **accept-lifetime** [*start-time* {**infinite** | *end-time* | **duration** *seconds*}] | **send-lifetime** [*start-time* {**infinite** | *end-time* | **duration** *seconds*}] **Example:** `Router(config-keychain)# key 1` `Router(config-keychain)# key-string ABcXyz`	Selects the parameters for the key chain. (These are submodes.) **Note** For details on these parameters, see the *Cisco IOS IP Command Reference, Volume 2 of 4, Routing Protocols, Release 12.3T.* **Note** **accept-lifetime** is ignored when a key chain is assigned to RSVP.
Step 5	**end** **Example:** `Router(config-keychain)# end`	Returns to privileged EXEC mode.

Binding a Key Chain to an RSVP Neighbor

Perform this task to bind a key chain to an RSVP neighbor for neighbor authentication.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp authentication neighbor address** *address* **key-chain** *key-chain-name*

   or

   ip rsvp authentication neighbor access-list *acl-name* or *acl-number* **key-chain** *key-chain-name*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`ip rsvp authentication neighbor address` *address* **key-chain** *key-chain-name* or **ip rsvp authentication neighbor access-list** *acl-name* **or** *acl-number* **key-chain** *key-chain-name* **Example:** `Router(config)# ip rsvp authentication neighbor access-list 1 key-chain neighbor_V`	Binds a key chain to an IP address or to an ACL and enters key-chain mode. **Note** If you are using an ACL, you must create it before you bind it to a key chain. See the **ip rsvp authentication** command in the Command Reference section for examples.
Step 4	**end** **Example:** `Router(config-keychain)# end`	Returns to privileged EXEC mode.

Troubleshooting Tips

After you enableRSVP authentication, RSVP logs system error events whenever an authentication check fails. These events are logged instead of just being displayed when debugging is enabled because they may indicate potential security attacks. The events are generated when:

- RSVP receives a message that does not contain the correct cryptographic signature. This could be due to misconfiguration of the authentication key or algorithm on one or more RSVP neighbors, but it may also indicate an (unsuccessful) attack.
- RSVP receives a message with the correct cryptographic signature, but with a duplicate authentication sequence number. This may indicate an (unsuccessful) message replay attack.
- RSVP receives a message with the correct cryptographic signature, but with an authentication sequence number that is outside the receive window. This could be due to a reordered burst of valid RSVP messages, but it may also indicate an (unsuccessful) message replay attack.
- Failed challenges result from timeouts or bad challenge responses.

To troubleshoot the RSVP Message Authentication feature, use the following commands in privileged EXEC mode.

Command	Purpose
`Router# debug ip rsvp authentication`	Displays output related to RSVP authentication.
`Router# debug ip rsvp dump signalling`	Displays brief information about signaling (Path and Resv) messages.
`Router# debug ip rsvp errors`	Displays error events including authentication errors.

Configuration Examples for RSVP Message Authentication

This section provides the following configuration examples:

- RSVP Message Authentication Per-Interface: Example, page 20
- RSVP Message Authentication Per-Neighbor: Example, page 22

RSVP Message Authentication Per-Interface: Example

In the following example, the cryptographic authentication parameters, including type, key, challenge, lifetime, and window size are configured; and authentication is activated:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface e0/0
Router(config-if)# ip rsvp bandwidth 7500 7500
Router(config-if)# ip rsvp authentication type sha-1
Router(config-if)# ip rsvp authentication key 11223344
Router(config-if)# ip rsvp authentication challenge
Router(config-if)# ip rsvp authentication lifetime 00:30:05
Router(config-if)# ip rsvp authentication window-size 2
Router(config-if)# ip rsvp authentication
```

In the following output from the **show ip rsvp interface detail** command, notice the cryptographic authentication parameters that you configured for the Ethernet0/0 interface:

```
Router# show ip rsvp interface detail

Et0/0:
   Bandwidth:
     Curr allocated: 0 bits/sec
     Max. allowed (total): 7500K bits/sec
     Max. allowed (per flow): 7500K bits/sec
     Max. allowed for LSP tunnels using sub-pools: 0 bits/sec
     Set aside by policy (total): 0 bits/sec
   Neighbors:
     Using IP encap: 0.  Using UDP encap: 0
   Signalling:
     Refresh reduction: disabled
   Authentication: enabled
     Key:         11223344
     Type:        sha-1
     Window size: 2
     Challenge:   enabled
```

In the preceding example, the authentication key appears in clear text. If you enter the **key-config-key 1** *string* command, the key appears encrypted, as in the following example:

```
Router# show ip rsvp interface detail

 Et0/0:
   Bandwidth:
     Curr allocated: 0 bits/sec
Max. allowed (total): 7500K bits/sec
     Max. allowed (per flow): 7500K bits/sec
     Max. allowed for LSP tunnels using sub-pools: 0 bits/sec
     Set aside by policy (total): 0 bits/sec
   Neighbors:
     Using IP encap: 0.  Using UDP encap: 0
   Signalling:
     Refresh reduction: disabled
   Authentication: enabled
     Key:         <encrypted>
     Type:        sha-1
     Window size: 2
     Challenge:   enabled
```

In the following output, notice that the authentication key changes from encrypted to clear text after the **no key config-key 1** command is issued:

```
Router# show running-config interface e0/0

Building configuration...

Current configuration :247 bytes
!
interface Ethernet0/0
 ip address 192.168.101.2 255.255.255.0
 no ip directed-broadcast
 ip pim dense-mode
 no ip mroute-cache
 no cdp enable
 ip rsvp bandwidth 7500 7500
 ip rsvp authentication key 7>70>9:7<872>?74
 ip rsvp authentication
end
```

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# no key config-key 1
Router(config)# end


Router# show running-config
*Jan 30  08:02:09.559:%SYS-5-CONFIG_I:Configured from console by console
int e0/0
Building configuration...

Current configuration :239 bytes
!
interface Ethernet0/0
 ip address 192.168.101.2 255.255.255.0
 no ip directed-broadcast
 ip pim dense-mode
 no ip mroute-cache
 no cdp enable
 ip rsvp bandwidth 7500 7500
 ip rsvp authentication key 11223344
 ip rsvp authentication
end
```

RSVP Message Authentication Per-Neighbor: Example

In the following example, a key chain with two keys for each neighbor is defined, then an access list and a key chain are created for neighbors V, Y, and Z and authentication is explicitly enabled for each neighbor and globally. However, only the neighbors specified will have their messages accepted; messages from other sources will be rejected. This enhances network security.

For security reasons, you should change keys on a regular basis. When the first key expires, the second key automatically takes over. At that point, you should change the first key's key-string to a new value and then set the send lifetimes to take over after the second key expires. The router will log an event when a key expires to remind you to update it.

The lifetimes of the first and second keys for each neighbor overlap. This allows for any clock synchronization problems that might cause the neighbors not to switch keys at the right time. You can avoid these overlaps by configuring the neighbors to use Network Time Protocol (NTP) to synchronize their clocks to a time server.

For an MPLS/TE configuration, physical addresses and router IDs are given.

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# key chain neighbor_V
Router(config-keychain)# key 1
Router(config-keychain-key)# key-string R72*UiAXy
Router(config-keychain-key)# send-life 02:00:00 1 jun 2003 02:00:00 1 aug 2003
Router(config-keychain-key)# exit
Router(config-keychain)# key 2
Router(config-keychain-key)# key-string Pl349&DaQ
Router(config-keychain-key)# send-life 01:00:00 1 jun 2003 02:00:00 1 aug 2003
Router(config-keychain-key)# exit
Router(config-keychain)# exit
Router(config)# key chain neighbor_Y
Router(config-keychain)# key 3
Router(config-keychain-key)# key-string *ZXFwR!03
Router(config-keychain-key)# send-life 02:00:00 1 jun 2003 02:00:00 1 aug 2003
Router(config-keychain-key)# exit
Router(config-keychain)# key 4
```

```
Router(config-keychain-key)# key-string UnGR8f&lOmY
Router(config-keychain-key)# send-life 01:00:00 1 jun 2003 02:00:00 1 aug 2003
Router(config-keychain-key)# exit
Router(config-keychain)# exit
Router(config)# key chain neighbor_Z
Router(config-keychain)# key 5
Router(config-keychain-key)# key-string P+T=77&/M
Router(config-keychain-key)# send-life 02:00:00 1 jun 2003 02:00:00 1 aug 2003
Router(config-keychain-key)# exit
Router(config-keychain)# key 6
Router(config-keychain-key)# key-string payattention2me
Router(config-keychain-key)# send-life 01:00:00 1 jun 2003 02:00:00 1 aug 2003
Router(config-keychain-key)# exit
Router(config-keychain)# exit
Router(config)# end
```

Note You can use the **key-config-key 1** *string* command to encrypt key chains for an interface, a neighbor, or globally.

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip access-list standard neighbor_V
Router(config-std-nacl)# permit 10.0.0.1 <------- physical address
Router(config-std-nacl)# permit 10.0.0.2 <------- physical address
Router(config-std-nacl)# permit 10.0.0.3 <------- router ID
Router(config-std-nacl)# exit
Router(config)# ip access-list standard neighbor_Y
Router(config-std-nacl)# permit 10.0.0.4 <------- physical address
Router(config-std-nacl)# permit 10.0.0.5 <------- physical address
Router(config-std-nacl)# permit 10.0.0.6 <------- router ID
Router(config-std-nacl)# exit
Router(config)# ip access-list standard neighbor_Z
Router(config-std-nacl)# permit 10.0.0.7 <------- physical address
Router(config-std-nacl)# permit 10.0.0.8 <------- physical address
Router(config-std-nacl)# permit 10.0.0.9 <------- router ID
Router(config-std-nacl)# exit
Router(config)# ip rsvp authentication neighbor access-list neighbor_V key-chain
neighbor_V
Router(config)# ip rsvp authentication neighbor access-list neighbor_Y key-chain
neighbor_Y
Router(config)# ip rsvp authentication neighbor access-list neighbor_Z key-chain
neighbor_Z
Router(config)# ip rsvp authentication
Router(config)# end
```

Additional References

The following sections provide references related to the RSVP Message Authentication feature.

Related Documents

Related Topic	Document Title
RSVP commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
QoS features including signaling, classification, and congestion management	"Quality of Service Overview" module
Inter-AS features including local policy support and per-neighbor keys authentication	"MPLS Traffic Engineering—Inter-AS-TE" module

Standards

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

MIBs

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

RFCs

RFCs	Title
RFC 1321	*The MD5 Message Digest Algorithm*
RFC 2104	*HMAC: Keyed-Hashing for Messaging Authentication*
RFC 2205	*Resource Reservation Protocol*
RFC 2209	*RSVP—Version 1 Message Processing Rules*
RFC 2401	*Security Architecture for the Internet Protocol*
RFC 2747	*RSVP Cryptographic Authentication*
RFC 3097	*RSVP Crytographic Authentication—Updated Message Type Value*
RFC 3174	*US Secure Hash Algorithm 1 (SHA1)*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Quality of Service Solutions Command Reference* at http://www.cisco.com/en/US/docs/ios/qos/command/reference/qos_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

- **clear ip rsvp authentication**
- **debug ip rsvp authentication**
- **ip rsvp authentication**
- **ip rsvp authentication challenge**
- **ip rsvp authentication key**
- **ip rsvp authentication key-chain**
- **ip rsvp authentication lifetime**
- **ip rsvp authentication neighbor**
- **ip rsvp authentication type**
- **ip rsvp authentication window-size**
- **show ip rsvp authentication**
- **show ip rsvp counters**
- **show ip rsvp interface**

Glossary

bandwidth—The difference between the highest and lowest frequencies available for network signals. The term also is used to describe the rated throughput capacity of a given network medium or protocol.

DMZ—demilitarized zone. The neutral zone between public and corporate networks.

flow—A stream of data traveling between two endpoints across a network (for example, from one LAN station to another). Multiple flows can be transmitted on a single circuit.

key—A data string that is combined with source data according to an algorithm to produce output that is unreadable until decrypted.

QoS—quality of service. A measure of performance for a transmission system that reflects its transmission quality and service availability.

router—A network layer device that uses one or more metrics to determine the optimal path along which network traffic should be forwarded. Routers forward packets from one network to another based on network layer information.

RSVP—Resource Reservation Protocol. A protocol that supports the reservation of resources across an IP network. Applications running on IP end systems can use RSVP to indicate to other nodes the nature (bandwidth, jitter, maximum burst, and so on) of the packet streams they want to receive.

security association—A block of memory used to hold all the information RSVP needs to authenticate RSVP signaling messages from a specific RSVP neighbor.

spoofing—The act of a packet illegally claiming to be from an address from which it was not actually sent. Spoofing is designed to foil network security mechanisms, such as filters and access lists.

TE—traffic engineering. The techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods had been used.

trusted neighbor—A router with authorized access to information.

RSVP Application ID Support

First Published: February 27, 2006
Last Updated: February 19, 2007

The RSVP Application ID Support feature introduces application-specific reservations, which enhance the granularity for local policy match criteria so that you can manage quality of service (QoS) on the basis of application type.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported use the "Feature Information for RSVP Application ID Support" section on page 23.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for RSVP Application ID Support

You must configure RSVP on one or more interfaces on at least two neighboring routers that share a link within the network.

Restrictions for RSVP Application ID Support

- RSVP policies apply only to PATH, RESV, PATHERROR, and RESVERROR messages.
- Merging of global and interface-based local policies is not supported; therefore, you cannot match on multiple policies.

Information About RSVP Application ID Support

To use the RSVP Application IP Support feature, you should understand the following concepts:

- Feature Overview of RSVP Application ID Support, page 2
- Benefits of RSVP Application ID Support, page 5

Feature Overview of RSVP Application ID Support

This section provides the following information:

- How RSVP Functions, page 2
- Sample Solution, page 3
- Global and Per-Interface RSVP Policies, page 4
- How RSVP Policies Are Applied, page 4
- Preemption, page 4

How RSVP Functions

Multiple applications such as voice and video need RSVP support. RSVP admits requests until the bandwidth limit is reached. RSVP does not differentiate between the requests and is not aware of the type of application for which the bandwidth is requested.

As a result, RSVP can exhaust the allowed bandwidth by admitting requests that represent just one type of application, causing all subsequent requests to be rejected because of unavailable bandwidth. For example, a few video calls could prevent all or most of the voice calls from being admitted because the video calls require a large amount of bandwidth and not enough bandwidth remains to accommodate the voice calls. With this limitation, you would probably not deploy RSVP for multiple applications especially if voice happens to be one of the applications for which RSVP is required.

The solution is to allow configuration of separate bandwidth limits for individual applications or classes of traffic. Limiting bandwidth per application requires configuring a bandwidth limit per application and having each reservation flag the application to which the reservation belongs so that it can be admitted against the appropriate bandwidth limit.

Application and Sub Application Identity Policy Element for Use with RSVP (IETF RFC 2872) allows for creation of separate bandwidth reservation pools. For example, an RSVP reservation pool can be created for voice traffic, and a separate RSVP reservation pool can be created for video traffic. This prevents video traffic from overwhelming voice traffic.

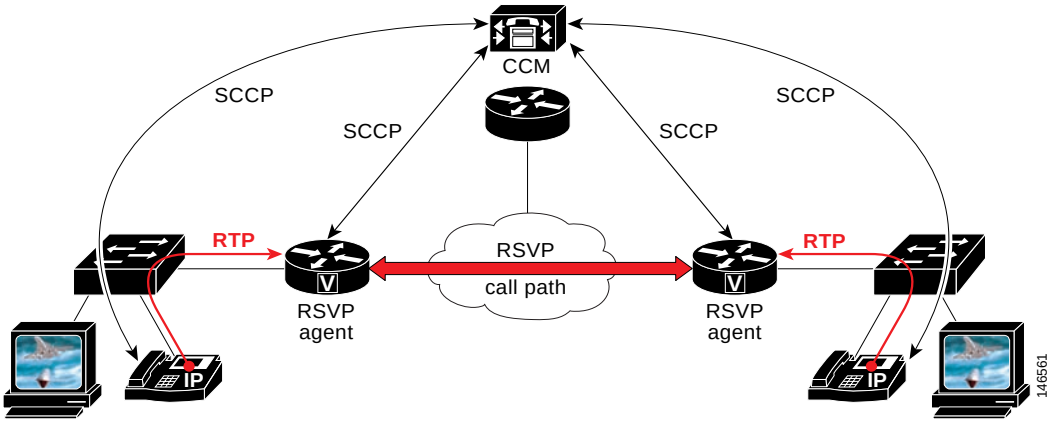
Note Before this feature, you could create access control lists (ACLs) that match on the differentiated services code points (DSCPs) of the IP header in an RSVP message. However, multiple applications could use the same DSCP; therefore, you could not uniquely identify applications in order to define separate policies for them.

Sample Solution

Figure 1 shows a sample solution in which application ID is used. In this example, bandwidth is allocated between the voice and video sessions that are being created by Cisco CallManager (CCM). Video requires much more bandwidth than voice, and if you do not separate the reservations, the video traffic could overwhelm the voice traffic.

CCM has been enhanced to use the RSVP Application ID Support feature. In this example, when CCM makes the RSVP reservation, CCM has the ability to specify whether the reservation should be made against a video RSVP bandwidth pool or a voice RSVP bandwidth pool. If there is not enough bandwidth remaining in the requested pool, even though there is enough bandwidth in the total RSVP allocation, RSVP signals CCM that there is a problem with the reservation. Figure 1 shows some of the signaling and data traffic that is sent during the session setup.

Figure 1 Sample Solution Using RSVP Application ID Support



In this scenario, the IP phones and IP video devices do not directly support RSVP. In order to allow RSVP to reserve the bandwidth for these devices, the RSVP agent component in the Cisco IOS router creates the reservation. During the setup of the voice or video session, CCM communicates with the RSVP agent and sends the parameters to reserve the necessary bandwidth.

When you want to make a voice or video call, the device signals CCM. CCM signals the RSVP agent, specifying the RSVP application ID that corresponds to the type of call, which is voice or video in this example. The RSVP agents establish the RSVP reservation across the network and tell CCM that the reservation has been made. CCM then completes the session establishment, and the Real-Time Transport

Protocol (RTP) traffic streams flow between the phones (or video devices). If the RSVP agents are unable to create the bandwidth reservations for the requested application ID, they communicate that information back to CCM, which signals this information back to you.

Global and Per-Interface RSVP Policies

You can configure RSVP policies globally and on a per-interface basis. You can also configure multiple global policies and multiple policies per interface.

Global RSVP policies restrict how much RSVP bandwidth a router uses regardless of the number of interfaces. You should configure a global policy if your router has CPU restrictions, one interface, or multiple interfaces that do not require different bandwidth limits.

Per-interface RSVP policies allow you to configure separate bandwidth pools with varying limits so that no one application, such as video, can consume all the RSVP bandwidth on a specified interface at the expense of other applications, such as voice, which would be dropped. You should configure a per-interface policy when you need greater control of the available bandwidth.

How RSVP Policies Are Applied

RSVP searches for policies whenever an RSVP message is processed. The policy tells RSVP if any special handling is required for that message.

If your network configuration has global and per-interface RSVP policies, the per-interface policies are applied first meaning that RSVP looks for policy-match criteria in the order in which the policies were configured. RSVP searches for policy-match criteria in the following order:

- Nondefault interface policies
- Default interface policy
- Nondefault global policies
- Global default policy

If RSVP finds no policy-match criteria, it accepts all incoming messages. To change this decision from accept to reject, issue the **ip rsvp policy default-reject** command.

Preemption

Preemption happens when one reservation receives priority over another because there is insufficient bandwidth in an RSVP pool. There are two types of RSVP bandwidth pools: local policy pools and interface pools. Local policies can be global or interface-specific. RSVP performs admission control against these pools when a RESV message arrives.

If an incoming reservation request matches an RSVP local policy that has an RSVP bandwidth limit (as configured with the **maximum bandwidth group** submode command) and that limit has been reached, RSVP tries to preempt other lower-priority reservations admitted by that policy. When there are too few of these lower-priority reservations, RSVP rejects the incoming reservation request. Then RSVP looks at the interface bandwidth pool that you configured by using the **ip rsvp bandwidth** command. If that bandwidth limit has been reached, RSVP tries to preempt other lower-priority reservations on that interface to accommodate the new reservation request. At this point, RSVP does not consider which local policies admitted the reservations. When not enough bandwidth on that interface pool can be preempted, RSVP rejects the new reservation even though the new reservation was able to obtain bandwidth from the local policy pool.

Preemption can also happen when you manually reconfigure an RSVP bandwidth pool of any type to a lower value such that the existing reservations using that pool no longer fit in the pool.

How Preemption Priorities Are Assigned and Signaled

If a received RSVP PATH or RESV message contains preemption priorities (signaled with an IETF RFC 3181 preemption priority policy element inside an IETF RFC 2750 POLICY_DATA object) and the priorities are higher than those contained in the matching local policy (if any), the offending message is rejected and a PATHERROR or RESVERROR message is sent in response. If the priorities are approved by the local policy, they are stored with the RSVP state in the router and forwarded to its neighbors.

If a received RSVP PATH or RESV message does not contain preemption priorities (as previously described) and you issued a global **ip rsvp policy preempt** command, and the message matches a local policy that contains a **preempt-priority** command, a POLICY_DATA object with a preemption priority element that contains the local policy's priorities is added to the message as part of the policy decision. These priorities are then stored with the RSVP state in the router and forwarded to neighbors.

Controling Preemption

The **ip rsvp policy preempt** command controls whether or not a router preempts any reservations when required. When you issue this command, a RESV message that subsequently arrives on an interface can preempt the bandwidth of one or more reservations on that interface if the assigned setup priority of the new reservation is higher than the assigned hold priorities of the installed reservations.

Benefits of RSVP Application ID Support

The RSVP Application ID Support feature provides the following benefits:

- Allows RSVP to identify applications uniquely and to separate bandwidth pools to be created for different applications so that one application cannot consume all the available bandwidth, thereby forcing others to be dropped.
- Integrates with the RSVP agent and CCM to provide a solution for call admission control (CAC) and QoS for Voice over IP (VoIP) and video conferencing applications in networks with multitiered, meshed topologies using signaling protocols such as SCCP to ensure that a single application does not overwhelm the available reserved bandwidth.
- Functions with any endpoint that complies with RFC 2872 or RFC 2205.

How to Configure RSVP Application ID Support

You can configure application IDs and local policies to use with RSVP-aware software programs such as CCM or to use with non-RSVP-aware applications such as static PATH and RESV messages.

This section contains the following procedures:

- Configuring RSVP Application IDs and Local Policies for RSVP-Aware Software Programs, page 6 (optional)
- Configuring RSVP Application IDs with Static Senders and Receivers for Non-RSVP-Aware Software Programs, page 10 (optional)
- Verifying the RSVP Application ID Support Configuration, page 14 (optional)

Configuring RSVP Application IDs and Local Policies for RSVP-Aware Software Programs

This section contains the following procedures:

- Configuring an Application ID, page 6 (required)

Note The following two local policy configuration procedures are optional; however, you must choose one or both.

- Configuring a Local Policy Globally, page 7 (optional)
- Configuring a Local Policy on an Interface, page 8 (optional)

Configuring an Application ID

Perform this task to configure an application ID.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp policy identity** *alias* **policy-locator** *locator*
4. Repeat Step 3 as needed to configure additional application IDs.
5. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**ip rsvp policy identity** *alias* **policy-locator** *locator* **Example:** `Router(config)# ip rsvp policy identity rsvp-voice policy-locator APP=Voice`	Defines RSVP application IDs to use as match criteria for local policies. • Enter a value for the *alias* argument, which is a string used within the router to reference the identity in RSVP configuration commands and show displays. The string can have as many as 64 printable characters (in the range 0x20 to 0x7E). **Note** If you use the " " or ? characters as part of the alias or locator string itself, you must type the CTRL/V key sequence before entering the embedded " " or ? characters. The alias is never transmitted to other routers. • Enter a value for the *locator* argument, which is a string that is signaled in RSVP messages and contains application IDs usually in X.500 Distinguished Name (DN) format. This can also be a regular expression. For more information on regular expressions, see the "Related Documents" section.
Step 4	Repeat Step 3 as needed to configure additional application IDs.	Defines additional application IDs.
Step 5	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

What to Do Next

Configure a local policy globally, on an interface, or both.

Configuring a Local Policy Globally

Perform this task to configure a local policy globally.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp policy local {acl** *acl1* [*acl2...acl8*] | **default** | **identity** *alias1* [*alias2...alias4*] | **origin-as** *as1* [*as2...as8*]**}**
4. Repeat Step 3 as needed to configure additional local policies.
5. {**accept** | **forward** [**all** | **path** | **path-error** | **resv** | **resv-error**] | **default** | **exit** | **fast-reroute** | **local-override** | **maximum** [**bandwidth** [**group** *x*] [**single** *y*] | **senders** *n*] | **preempt-priority** [**traffic-eng** *x*] *setup-priority* [*hold-priority*]}
6. **Repeat Step 5 as needed to configure additional submode commands.**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp policy local** {**acl** *acl1* [*acl2...acl8*] | **default** | **identity** *alias1* [*alias2...alias4*] | **origin-as** *as1* [*as2...as8*]} **Example:** `Router(config)# ip rsvp policy local identity rsvp-voice`	Creates a local policy to determine how RSVP resources are used in a network and enters local policy configuration mode. • Enter the **identity** *alias1* keyword and argument combination to specify an application ID alias.
Step 4	Repeat Step 3 as needed to configure additional local policies.	(Optional) Configures additional local policies.
Step 5	{**accept** | **forward** [**all** | **path** | **path-error** | **resv** | **resv-error**] | **default** | **exit** | **fast-reroute** | **local-override** | **maximum** [**bandwidth** [**group** *x*] [**single** *y*] | **senders** *n*] | **preempt-priority** [**traffic-eng** *x*] *setup-priority* [*hold-priority*]} **Example:** `Router(config-rsvp-policy-local)# forward all`	(Optional) Defines the properties of the local policy that you are creating. (These are the submode commands.) **Note** This is an optional step. An empty policy rejects everything, which may be desired in some cases. See the **ip rsvp policy local** command for more detailed information on submode commands.
Step 6	`Repeat Step 5 as needed to configure additional submode commands.`	(Optional) Configures additional submode commands.
Step 7	**end** **Example:** `Router(config-rsvp-policy-local)# end`	Exits local policy configuration mode and returns to privileged EXEC mode.

Configuring a Local Policy on an Interface

Perform this task to configure a local policy on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. Repeat Step 3 as needed to configure additional interfaces.
5. **ip rsvp bandwidth** [*interface-kbps] [single-flow-kbps]*
6. Repeat Step 5 as needed to configure bandwidth for additional interfaces.

7. **ip rsvp policy local {acl** *acl1* [*acl2...acl8*] | **default** | **identity** *alias1* [*alias2...alias4*] | **origin-as** *as1* [*as2...as8*]**}**
8. Repeat Step 7 as needed to configure additional local policies.
9. {**accept** | **forward** [**all** | **path** | **path-error** | **resv** | **resv-error**] | **default** | **exit** | **fast-reroute** | **local-override** | **maximum** [**bandwidth** [**group** *x*] [**single** *y*] | **senders** *n*] | **preempt-priority** [**traffic-eng** *x*] *setup-priority* [*hold-priority*]}
10. Repeat Step 9 as needed to configure additional submode commands.
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`interface` *type number* **Example:** `Router(config)# interface Ethernet0/0`	Configures the interface type and number and enters interface configuration mode.
Step 4	Repeat Step 3 as needed to configure additional interfaces.	(Optional) Configures additional interfaces.
Step 5	`ip rsvp bandwidth [`*interface-kbps*`] [`*single-flow-kbps*`]` **Example:** `Router(config-if)# ip rsvp bandwidth 500 500`	Enables RSVP on an interface. • The optional *interface-kbps* and *single-flow-kbps* arguments specify the amount of bandwidth that can be allocated by RSVP flows or to a single flow, respectively. Values are from 1 to 1000,000.
Step 6	Repeat Step 5 as needed to configure bandwidth for additional interfaces.	(Optional) Configures bandwidth for additional interfaces.
Step 7	**ip rsvp policy local** {**acl** *acl1* [*acl2...acl8*]	**default**	**identity** *alias1* [*alias2...alias4*]	**origin-as** *as1* [*as2...as8*]} **Example:** `Router(config-if)# ip rsvp policy local identity rsvp-voice`	Creates a local policy to determine how RSVP resources are used in a network. • Enter the **identity** *alias1* keyword argument combination to specify an application ID alias.
Step 8	`Repeat Step 7 as needed to configure additional local policies.`	(Optional) Configures additional local policies.

	Command or Action	Purpose
Step 9	{**accept** | **forward** [**all** | **path** | **path-error** | **resv** | **resv-error**] | **default** | **exit** | **fast-reroute** | **local-override** | **maximum** [**bandwidth** [**group** *x*] [**single** *y*] | **senders** *n*] | **preempt-priority** [**traffic-eng** *x*] *setup-priority* [*hold-priority*]} **Example:** `Router(config-rsvp-policy-local)# forward all`	(Optional) Defines the properties of the local policy that you are creating and enters local policy configuration mode. (These are the submode commands.) **Note** This is an optional step. An empty policy rejects everything, which may be desired in some cases. See the **ip rsvp policy local** command for more detailed information on submode commands.
Step 10	Repeat Step 9 as needed to configure additional submode commands.	(Optional) Configures additional submode commands.
Step 11	**end** **Example:** `Router(config-rsvp-policy-local)# end`	Exits local policy configuration mode and returns to privileged EXEC mode.

Configuring RSVP Application IDs with Static Senders and Receivers for Non-RSVP-Aware Software Programs

This section contains the following procedures:

- Configuring an Application ID, page 10 (required)
- Configuring a Static Sender with an Application ID, page 11 (optional)
- Configuring a Static Receiver with an Application ID, page 13 (optional)

Configuring an Application ID

Perform this task to configure an application ID.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp policy identity** *alias* **policy-locator** *locator*
4. Repeat step 3 to configure additional application IDs.
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp policy identity** *alias* **policy-locator** *locator* **Example:** `Router(config)# ip rsvp policy identity rsvp-voice policy-locator "APP=Voice"`	Defines RSVP application IDs to use as match criteria for local policies. • Enter a value for the *alias* argument, which is a string used within the router to reference the identity in RSVP configuration commands and show displays. The string can have as many as 64 printable characters (in the range 0x20 to 0x7E). **Note** If you use the " " or ? characters as part of the alias or locator string itself, you must type the CTRL/V key sequence before entering the embedded " " or ? characters. The alias is never transmitted to other routers. • Enter a value for the *locator* argument, which is a string that is signaled in RSVP messages and contains application IDs usually in X.500 Distinguished Name (DN) format. **Note** Repeat this step as needed to configure additional application IDs.
Step 4	Repeat step 3 to configure additional application IDs.	Configures additional application IDs.
Step 5	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring a Static Sender with an Application ID

Perform this task to configure a static RSVP sender with an application ID to make the router proxy an RSVP PATH message containing an application ID on behalf of an RSVP-unaware sender application.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **ip rsvp sender-host** *session-ip-address sender-ip-address* {**tcp** | **udp** | *ip-protocol*} *session-d-port sender-s-port bandwidth burst-size* [**identity** *alias*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp sender-host** *session-ip-address sender-ip-address* {**tcp** | **udp** | *ip-protocol*} *session-d-port sender-s-port bandwidth burst-size* [**identity** *alias*] **Example:** `Router(config)# ip rsvp sender-host 10.0.0.7 10.0.0.1 udp 1 1 10 10 identity rsvp-voice`	Enables a router to simulate a host generating RSVP PATH messages. • The optional **identity** *alias* keyword and argument combination specifies an application ID alias. The string can have as many as 64 printable characters (in the range 0x20 to 0x7E). **Note** If you use the “ ” or ? characters as part of the alias string itself, you must type the CTRL/V key sequence before entering the embedded “ ” or ? characters. The alias is never transmitted to other routers.
Step 4	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring a Static Receiver with an Application ID

Perform this task to configure a static RSVP receiver with an application ID to make the router proxy an RSVP RESV message containing an application ID on behalf of an RSVP-unaware receiver application.

Note You can also configure a static listener to use with an application ID. If an incoming PATH message contains an application ID and/or a preemption priority value, the listener includes them in the RESV message sent in reply. See the "Feature Information for RSVP Application ID Support" section on page 23 for more information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp reservation-host** *session-ip-address sender-ip-address* {**tcp** | **udp** | *ip-protocol*} *session-d-port sender-s-port* {**ff** | **se** | **wf**} {**rate** | **load**} *bandwidth burst-size* [**identity** *alias*]
   or
   ip rsvp reservation *session-ip-address sender-ip-address* {**tcp** | **udp** | *ip-protocol*} *session-d-port sender-s-port next-hop-ip-address next-hop-interface* {**ff** | **se** | **wf**} {**rate** | **load**} *bandwidth burst-size* [**identity** *alias*]

Note Use the **ip rsvp reservation-host** command if the router is the destination or the **ip rsvp reservation** command to have the router proxy on behalf of a downstream host.

4. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**ip rsvp reservation-host** *session-ip-address sender-ip-address* {**tcp** | **udp** | *ip-protocol*} *session-d-port sender-s-port* {**ff** | **se** | **wf**} {**rate** | **load**} *bandwidth burst-size* [**identity** *alias*] or **ip rsvp reservation** *session-ip-address sender-ip-address* {**tcp** | **udp** | *ip-protocol*} *session-d-port sender-s-port next-hop-ip-address next-hop-interface* {**ff** | **se** | **wf**} {**rate** | **load**} *bandwidth burst-size* [**identity** *alias*] **Example:** `Router(config)# ip rsvp reservation-host 10.1.1.1 10.30.1.4 udp 20 30 se load 100 60 identity rsvp-voice` `Router(config)# ip rsvp reservation 10.1.1.1 0.0.0.0 udp 20 0 172.16.4.1 Ethernet1 wf rate 350 65 identity xyz`	Enables a router to simulate a host generating RSVP RESV messages. • The optional **identity** *alias* keyword and argument combination specifies an application ID alias. The string can have as many as 64 printable characters (in the range 0x20 to 0x7E). **Note** If you use the “ ” or ? characters as part of the alias string itself, you must type the CTRL/V key sequence before entering the embedded “ ” or ? characters. The alias is never transmitted to other routers. **Note** Use the **ip rsvp reservation-host** command if the router is the destination or the **ip rsvp reservation** command to have the router proxy on behalf of a downstream host.
Step 4	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Verifying the RSVP Application ID Support Configuration

Perform the following task to verify the configuration.

SUMMARY STEPS

1. **enable**

Note You can use the following commands in user EXEC or privileged EXEC mode.

2. **show ip rsvp host** {**senders** | **receivers**} [*group-name* | *group-address*]
3. **show ip rsvp policy identity** [*regular-expression*]
4. **show ip rsvp policy local** [**detail**] [**interface** *name*] [**default** | **acl** *acl* | **origin-as** *as* | **identity** *alias*]
5. **show ip rsvp reservation** [**detail**] [**filter** [**destination** *ip-addr* | *hostname*] [**source** *ip-addr* | *hostname*] [**dst-port** *port*] [**src-port** *port*]]
6. **show ip rsvp sender** [**detail**] [**filter** [**destination** *ip-addr* | *hostname*] [**source** *ip-addr* | *hostname*] [**dst-port** *port*] [**src-port** *port*]]
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	(Optional) Enables privileged EXEC mode. • Enter your password if prompted. **Note** Skip this step if you are using the commands in user EXEC mode.
Step 2	**show ip rsvp host** {**senders** | **receivers**} [*group-name* | *group-address*] **Example:** `Router# show ip rsvp host senders`	Displays specific information for an RSVP host. **Note** Use this command only on routers from which PATH and RESV messages originate.
Step 3	**show ip rsvp policy identity** [*regular-expression*] **Example:** `Router# show ip rsvp policy identity voice100`	Displays selected RSVP identities in a router configuration. • The optional *regular-expression* argument allows pattern matching on the alias strings of the RSVP identities to be displayed. **Note** For more information on regular expressions, see the "Related Documents" section on page 21.
Step 4	**show ip rsvp policy local** [**detail**] [**interface** *name*] [**default** | **acl** *acl* | **origin-as** *as* | **identity** *alias*] **Example:** `Router# show ip rsvp policy local identity voice100`	Displays the local policies currently configured. • The optional **detail** keyword and the optional **interface** *name* keyword and argument combination can be used with any of the match criteria.
Step 5	**show ip rsvp reservation** [**detail**] [**filter** [**destination** *ip-addr* | *hostname*] [**source** *ip-addr* | *hostname*] [**dst-port** *port*] [**src-port** *port*]] **Example:** `Router# show ip rsvp reservation detail`	Displays RSVP-related receiver information currently in the database. • The optional **detail** keyword displays additional output with information about where the policy originated as well as which application ID was signaled in the RESV message. **Note** The optional **filter** keyword is supported in Cisco IOS Releases 12.0S and 12.2S only.
Step 6	**show ip rsvp sender** [**detail**] [**filter** [**destination** *ip-addr* | *hostname*] [**source** *ip-addr* | *hostname*] [**dst-port** *port*] [**src-port** *port*]] **Example:** `Router# show ip rsvp sender detail`	Displays RSVP PATH-related sender information currently in the database. • The optional **detail** keyword displays additional output with information that includes which application ID was signaled in the PATH message. **Note** The optional **filter** keyword is supported in Cisco IOS Releases 12.0 S and 12.2 S only.
Step 7	**exit** **Example:** `Router# exit`	Exits privileged EXEC mode and returns to user EXEC mode.

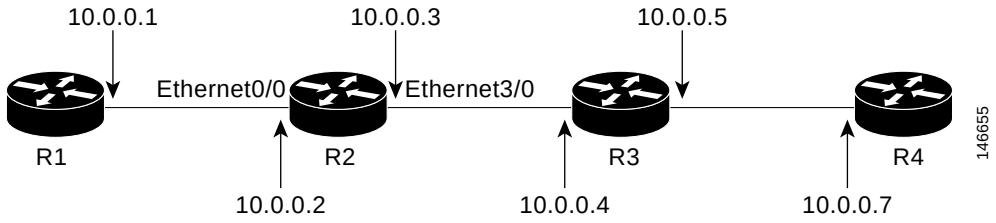
Configuration Examples for RSVP Application ID Support

This section provides configuration examples for the RSVP Application ID Support feature.

Configuring RSVP Application ID Support: Example

The four-router network in Figure 2 contains the following configurations:

Figure 2 Sample Network with Application Identities and Local Policies



Configuring a Proxy Receiver on R4

The following example configures R4 with a proxy receiver to create an RESV message to match the PATH message for the destination 10.0.0.7:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip rsvp listener 10.0.0.7 any any reply
Router(config)# end
```

Configuring an Application ID and a Global Local Policy on R3

The following example configures R3 with an application ID called video and a global local policy in which all RSVP messages are being accepted and forwarded:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip rsvp policy identity video policy-locator video
Router(config)# ip rsvp policy local identity video
Router(config-rsvp-policy-local)# forward all
Router(config-rsvp-policy-local)# end
```

Configuring an Application ID and Separate Bandwidth Pools on R2 for Per-Interface Local Policies

The following example configures R2 with an application ID called video, which is a wildcard regular expression to match any application ID that contains the substring video:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip rsvp policy identity video policy-locator .*Video.*
Router(config-rsvp-id)# end
```

The following example configures R2 with a local policy on ingress Ethernet interface 0/0:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface Ethernet0/0
Router(config-if)# ip address 10.0.0.2 255.0.0.0
Router(config-if)# no cdp enable
Router(config-if)# ip rsvp bandwidth 200
Router(config-if)# ip rsvp policy local identity video
Router(config-rsvp-policy-local)# maximum senders 10
Router(config-rsvp-policy-local)# maximum bandwidth group 100
Router(config-rsvp-policy-local)# maximum bandwidth single 10
Router(config-rsvp-policy-local)# forward all
Router(config-rsvp-policy-local)# end
```

The following example configures R2 with a local policy on egress Ethernet interface 3/0:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface Ethernet3/0
Router(config-if)# ip address 10.0.0.3 255.0.0.0
Router(config-if)# no cdp enable
Router(config-if)# ip rsvp bandwidth 200
Router(config-if)# ip rsvp policy local identity video
Router(config-rsvp-policy-local)# maximum senders 10
Router(config-rsvp-policy-local)# maximum bandwidth group 100
Router(config-rsvp-policy-local)# maximum bandwidth single 10
Router(config-rsvp-policy-local)# forward all
Router(config-rsvp-policy-local)# end
```

Note PATH messages arrive on ingress Ethernet interface 0/0 and RESV messages arrive on egress Ethernet interface 3/0.

Configuring an Application ID and a Static Reservation from R1 to R4

The following example configures R1 with an application ID called video and initiates a host generating a PATH message with that application ID:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip rsvp policy identity video policy-locator "GUID=www.cisco.com,
APP=Video, VER=1.0"
Router(config)# ip rsvp sender-host 10.0.0.7 10.0.0.1 udp 1 1 10 10 identity video
Router(config)# end
```

Verifying RSVP Application ID Support: Example

This section contains the following verification examples:

Verifying the Application ID and the Global Local Policy on R3

The following example verifies that a global local policy has been configured on R3 with an application ID called Video:

```
Router# show ip rsvp policy local detail

Global:
  Policy for ID(s): Video

    Preemption Scope: Unrestricted.
    Local Override:   Disabled.
    Fast ReRoute:     Accept.
    Handle:           23000404.

                          Accept              Forward
    Path:                 Yes                 Yes
    Resv:                 Yes                 Yes
    PathError:            Yes                 Yes
    ResvError:            Yes                 Yes

                          Setup Priority      Hold Priority
    TE:                   N/A                 N/A
    Non-TE:               N/A                 N/A

                          Current             Limit
    Senders:              1                   N/A
    Receivers:            1                   N/A
    Conversations:        1                   N/A
    Group bandwidth (bps): 10K                N/A
    Per-flow b/w (bps):   N/A                 N/A


Generic policy settings:
    Default policy: Accept all
    Preemption:     Disabled
```

Verifying the Application ID and the Per-Interface Local Policies on R2

The following example verifies that an application ID called Video has been created on R2:

```
Router# show ip rsvp policy identity

Alias: Video
  Type:    Application ID
  Locator: .*Video.*
```

The following example verifies that per-interface local policies have been created on Ethernet interface 0/0 and Ethernet interface 3/0 on R2:

```
Router# show ip rsvp policy local detail

Ethernet0/0:
  Policy for ID(s): Video

    Preemption Scope: Unrestricted.
    Local Override:   Disabled.
    Fast ReRoute:     Accept.
    Handle:           26000404.

                          Accept              Forward
    Path:                 Yes                 Yes
    Resv:                 Yes                 Yes
    PathError:            Yes                 Yes
    ResvError:            Yes                 Yes

                          Setup Priority      Hold Priority
    TE:                   N/A                 N/A
    Non-TE:               N/A                 N/A

                          Current             Limit
    Senders:              1                   10
    Receivers:            0                   N/A
    Conversations:        0                   N/A
    Group bandwidth (bps): 0                  100K
    Per-flow b/w (bps):   N/A                 10K


Ethernet3/0:
  Policy for ID(s): Video

    Preemption Scope: Unrestricted.
    Local Override:   Disabled.
    Fast ReRoute:     Accept.
    Handle:           5A00040A.

                          Accept              Forward
    Path:                 Yes                 Yes
    Resv:                 Yes                 Yes
    PathError:            Yes                 Yes
    ResvError:            Yes                 Yes

                          Setup Priority      Hold Priority
    TE:                   N/A                 N/A
    Non-TE:               N/A                 N/A

                          Current             Limit
    Senders:              0                   10
    Receivers:            1                   N/A
    Conversations:        1                   N/A
    Group bandwidth (bps): 10K                100K
    Per-flow b/w (bps):   N/A                 10K


Generic policy settings:
    Default policy: Accept all
    Preemption:     Disabled
```

Note Notice in the above display that the ingress interface has only its senders counter incremented because the PATH message is checked there. However, the egress interface has its receivers, conversations, and group bandwidth counters incremented because the reservation is checked on the incoming interface, which is the egress interface on R2.

Verifying the Application ID and the Reservation on R1

The following example verifies that a PATH message containing the application ID called Video has been created on R1:

```
Router# show ip rsvp sender detail

PATH Session address: 10.0.0.7, port: 1. Protocol: UDP
  Sender address: 10.0.0.1, port: 1
     Inbound from: 10.0.0.1 on interface:
  Traffic params - Rate: 10K bits/sec, Max. burst: 10K bytes
                   Min Policed Unit: 0 bytes, Max Pkt Size 4294967295 bytes
  Path ID handle: 02000402.
  Incoming policy: Accepted. Policy source(s): Default
    Application ID: 'GUID=www.cisco.com, APP=Video, VER=1.0'
  Status: Proxied
  Output on Ethernet0/0. Policy status: Forwarding. Handle: 01000403
    Policy source(s): Default
```

Note You can issue the **debug ip rsvp dump path** and the **debug ip rsvp dump resv** commands to get more information about a sender and the application ID that it is using.

The following example verifies that a reservation with the application ID called Video has been created on R1:

```
Router# show ip rsvp reservation detail

RSVP Reservation. Destination is 10.0.0.7, Source is 10.0.0.1,
  Protocol is UDP, Destination port is 1, Source port is 1
  Next Hop is 10.0.0.2, Interface is Ethernet0/0
  Reservation Style is Fixed-Filter, QoS Service is Guaranteed-Rate
  Resv ID handle: 01000405.
  Created: 10:07:35 EST Thu Jan 12 2006
  Average Bitrate is 10K bits/sec, Maximum Burst is 10K bytes
  Min Policed Unit: 0 bytes, Max Pkt Size: 0 bytes
  Status:
  Policy: Forwarding. Policy source(s): Default
    Application ID: 'GUID=www.cisco.com, APP=Video, VER=1.0'
```

Additional References

The following sections provide references related to the RSVP Application ID Support feature.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
QoS configuration tasks related to RSVP	"Configuring RSVP" module
Cisco United Communications Manager (CallManager) and related features	"Overview of Cisco Unified Communications Manager and Cisco IOS Interoperability" module
Regular expressions	"Using the Cisco IOS Command-Line Interface" module

Standards

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

MIBs

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

RFCs

RFC	Title
RFC 2205	Resource ReSerVation Protocol (RSVP)
RFC 2872	Application and Sub Application Identity Policy Element for Use with RSVP
RFC 3181	Signaled Preemption Priority Policy Element
RFC 3182	Identity Representation for RSVP

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Quality of Service Solutions Command Reference* at http://www.cisco.com/en/US/docs/ios/qos/command/reference/qos_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

- **ip rsvp listener**
- **ip rsvp policy identity**
- **ip rsvp policy local**
- **ip rsvp reservation**
- **ip rsvp reservation-host**
- **ip rsvp sender**
- **ip rsvp sender-host**
- **maximum (local policy)**
- **show ip rsvp host**
- **show ip rsvp policy identity**
- **show ip rsvp policy local**

Feature Information for RSVP Application ID Support

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for RSVP Application ID Support

Feature Name	Releases	Feature Information
RSVP Application ID Support	12.4(6)T, 12.2(33)SRB	The RSVP Application ID Support feature introduces application-specific reservations, which enhance the granularity for local policy-match criteria so that you can manage quality of service (QoS) on the basis of application type.

Glossary

ACL—access control list. An ACL consists of individual filtering rules grouped together in a single list. It is generally used to provide security filtering, although it may be used to provide a generic packet classification facility.

admission control—The process in which an RSVP reservation is accepted or rejected on the basis of end-to-end available network resources.

application identity (ID)—A string that can be inserted in a policy element in a POLICY_DATA object of an RSVP message to identify the application and associate it with the RSVP reservation request, thus allowing routers along the path to make appropriate decisions based on the application information.

autonomous system—A collection of networks that share the same routing protocol and that are under the same system administration.

bandwidth—The difference between the highest and lowest frequencies available for network signals. The term also is used to describe the rated throughput capacity of a given network medium or protocol.

CCM—Cisco CallManager. The software-based, call-processing component of the Cisco IP telephony solution. The software extends enterprise telephony features and functions to packet telephony network devices such as IP phones, media processing devices, Voice-over-IP (VoIP) gateways, and multimedia applications.

DSCP—differentiated services code point. The six most significant bits of the 1-byte IP type of service (ToS) field. The per-hop behavior represented by a particular DSCP value is configurable. DSCP values range between 0 and 63.

policy—Any defined rule that determines the use of resources within the network. A policy can be based on a user, a device, a subnetwork, a network, or an application.

QoS—quality of service. A measure of performance for a transmission system that reflects its transmission quality and service availability.

RSVP—Resource Reservation Protocol. A protocol for reserving network resources to provide quality of service guarantees to application flows.

RSVP agent—Implements a Resource Reservation Protocol (RSVP) agent on Cisco IOS voice gateways that support Cisco CallManager 5.0.

RTP—Real-Time Transport Protocol. An Internet protocol for transmitting real-time data such as voice and video.

router—A network layer device that uses one or more metrics to determine the optimal path along which network traffic should be forwarded. Routers forward packets from one network to another on the basis of network layer information.

TE—traffic engineering. The techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods had been used.

RSVP Fast Local Repair

First Published: February 19, 2007
Last Updated: October 2, 2009

The RSVP Fast Local Repair feature provides quick adaptation to routing changes occurring in global as well as VRF routing domains, without the overhead of the refresh period to guarantee the quality of service (QoS) for data flows. With fast local repair (FLR), Resource Reservation Protocol (RSVP) speeds up its response to routing changes from 30 seconds to a few seconds.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "Feature Information for RSVP FLR" section on page 13.

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

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for RSVP FLR

You must configure RSVP on one or more interfaces on at least two neighboring routers that share a link within the network.

Restrictions for RSVP FLR

- RSVP FLR applies only when RSVP is used to set up resource reservations for IPv4 unicast flows; IPv4 multicast flows are not supported.
- RSVP FLR does not apply to traffic engineering (TE) tunnels and, therefore, does not affect TE sessions.
- RSVP FLR does not support message bundling.

Information About RSVP FLR

To use the RSVP FLR feature, you should understand the following concepts:

- Feature Overview of RSVP FLR, page 2
- Benefits of RSVP FLR, page 3

Feature Overview of RSVP FLR

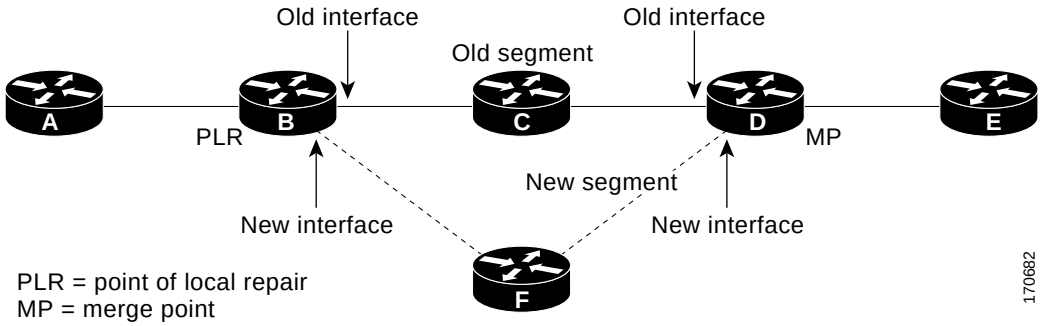
RSVP FLR provides for dynamic adaptation when routing changes occur in global or VRF routing domains. When a route changes, the next PATH and RESV message refreshes establish path and reservation states along the new route. Depending on the configured refresh interval, this reroute happens in tens of seconds. However, during this time, the QoS of flows is not guaranteed because congestion may occur while data packets travel over links where reservations are not yet in place.

In order to provide faster adaptation to routing changes, without the overhead of a refresh period, RSVP registers with the routing information base (RIB) and receives notifications when routes change, thereby triggering state refreshes for the affected destinations. These triggered refreshes use the new route information and, as a result, install reservations over the new path.

When routes change, RSVP has to reroute all affected paths and reservations. Without FLR, the reroute happens when refresh timers expire for the path states. With real time applications such as VoIP and VoD, the requirement changes and the reroute must happen quickly, within three seconds from the triggering event such as link down or link up.

Figure 1 illustrates the FLR process.

Figure 1 Overview of RSVP FLR



Initial RSVP states are installed for an IPv4 unicast flow over Routers A, B, C, D, and E. Router A is the source or headend, while Router E is the destination or tailend. The data packets are destined to an address of Router E. Assume that a route change occurs, and the new path taken by the data packets is from Router A to Router B to Router F to Router D to Router E; therefore, the old and new paths differ on the segments between Routers B and D. The Router B to Router C to Router D segment is the old segment, while the Router B to Router F to Router D segment is the new segment.

A route may change because of a link or node failure, or if a better path becomes available.

RSVP at Router B detects that the route change affects the RSVP flow and initiates the FLR procedure. The node that initiates an FLR repair procedure, Router B in Figure 1, is the point of local repair (PLR). The node where the new and old segments meet, Router D in Figure 1, is the merge point (MP). The interfaces at the PLR and the MP that are part of the old segment are the old interfaces, while the interfaces that are part of the new segment are the new interfaces.

If a route has changed because of a failure, the PLR may not be the node that detects the failure. For example, it is possible that the link from Router C to Router D fails, and although Router C detects the failure, the route change at Router B is the trigger for the FLR procedure. Router C, in this case, is also referred to as the node that detects the failure.

The support for FLR in VRF domains means that RSVP can get a route change notification, even if there is a route change in any VRF domains, as RSVP FLR was previously supported only in the global routing domain.

Benefits of RSVP FLR

Faster Response Time to Routing Changes

FLR reduces the time that it takes for RSVP to determine that a physical link has gone down and that the data packets have been rerouted. Without FLR, RSVP may not recognize the link failure for 30 seconds when all of the sessions are impacted by having too much traffic for the available bandwidth. With FLR, this time can be significantly reduced to a few seconds.

After detecting the failure, RSVP recomputes the admission control across the new link. If the rerouted traffic fits on the new link, RSVP reserves the bandwidth and guarantees the QoS of the new traffic.

If admission control fails on the new route, RSVP does not explicate tear down the flow, but instead sends a RESVERROR message towards the receiver. If a proxy receiver is running, then RSVP sends a PATHERROR message towards the headend, in response to the RESVERROR message, indicating the admission failure. In both cases, with and without a proxy receiver, the application tears down the failed session either at the headend or at the final destination.

Until this happens, the data packets belonging to this session still flow over the rerouted segment although admission has failed and QoS is affected.

The support of FLR in VRF domains means that if there is a route change in any routing domain, RSVP can use FLR to adapt to the routing change, as RSVP FLR was previously supported only in the global routing domain.

How to Configure RSVP FLR

You can configure the RSVP FLR parameters in any order that you want.

This section contains the following procedures:

- Configuring the RSVP FLR Wait Time, page 4 (required)
- Configuring the RSVP FLR Repair Rate, page 5 (required)
- Configuring the RSVP FLR Notifications, page 6 (required)
- Verifying the RSVP FLR Configuration, page 7 (optional)

Configuring the RSVP FLR Wait Time

Perform this task to configure the RSVP FLR wait time.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*] [**sub-pool** [*sub-pool-kbps*]]
5. **ip rsvp signalling fast-local-repair wait-time** *interval*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet0/0`	Configures the interface type and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*] [**sub-pool** [*sub-pool-kbps*]] Example: Router(config-if)# ip rsvp bandwidth 7500 7500	Enables RSVP on an interface. • The optional *interface-kbps* and *single-flow-kbps* arguments specify the amount of bandwidth that can be allocated by RSVP flows or to a single flow, respectively. Values are from 1 to 10000000. • The optional **sub-pool** and *sub-pool-kbps* keyword and argument specify subpool traffic and the amount of bandwidth that can be allocated by RSVP flows. Values are from 1 to 10000000. **Note** Repeat this command for each interface on which you want to enable RSVP.
Step 5	**ip rsvp signalling fast-local-repair wait-time** *interval* **Example:** Router(config-if)# ip rsvp signalling fast-local-repair wait-time 100	Configures the delay that RSVP uses before starting an FLR procedure. • Values for the *interval* argument are 0 to 5000 milliseconds (ms); the default is 0.
Step 6	**end** **Example:** Router(config-if)# end	(Optional) Returns to privileged EXEC mode.

Configuring the RSVP FLR Repair Rate

Perform this task to configure the RSVP FLR repair rate.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp signalling fast-local-repair rate** *rate*
4. **exit**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**ip rsvp signalling fast-local-repair rate** *rate* **Example:** `Router(config)# ip rsvp signalling fast-local-repair rate 100`	Configures the repair rate that RSVP uses for an FLR procedure. • Values for the *rate* argument are 1 to 2500 messages per second; the default is 400. **Note** See the **ip rsvp signalling fast-local-repair rate** command for more information.
Step 4	**exit** **Example:** `Router(config)# exit`	(Optional) Returns to privileged EXEC mode.

Configuring the RSVP FLR Notifications

Perform this task to configure the number of RSVP FLR notifications.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp signalling fast-local-repair notifications** *number*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp signalling fast-local-repair notifications** *number* **Example:** `Router(config)# ip rsvp signalling fast-local-repair notifications 100`	Configures the number of path state blocks (PSBs) that RSVP processes before it suspends. • Values for the *number* argument are 10 to 10000; the default is 1000.
Step 4	**exit** **Example:** `Router(config)# exit`	(Optional) Returns to privileged EXEC mode.

Verifying the RSVP FLR Configuration

Perform this task to verify the configuration.

Note You can use the following **show** commands in user EXEC or privileged EXEC mode.

SUMMARY STEPS

1. **enable**
2. **show ip rsvp signalling fast-local-repair** [**statistics** [**detail**]]
3. **show ip rsvp interface** [**vrf** {***** | *vrf-name*}] [**detail**] [*interface-type interface-number*]
4. **show ip rsvp**
5. **show ip rsvp sender** [**vrf** {***** | *vrf-name*}] [**detail**] [**filter** [**destination** *ip-addr* | *hostname*] [**source** *ip-addr* | *hostname*] [**dst-port** *port*] [**src-port** *port*]]
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	(Optional) Enables privileged EXEC mode. • Enter your password if prompted. **Note** Skip this step if you are using the **show** commands in user EXEC mode.
Step 2	**show ip rsvp signalling fast-local-repair** [**statistics** [**detail**]] **Example:** `Router# show ip rsvp signalling fast-local-repair statistics detail`	Displays FLR-specific information that RSVP maintains. • The optional **statistics** and **detail** keywords display additional information about the FLR parameters.
Step 3	**show ip rsvp interface** [**vrf** {*****	*vrf-name*}] [**detail**] [*interface-type interface-number*] **Example:** `Router# show ip rsvp interface ethernet 1/0`	Displays RSVP-related information. • The optional **detail** keyword displays additional information including FLR parameters.
Step 4	**show ip rsvp** **Example:** `Router# show ip rsvp`	Displays general RSVP related information.

	Command or Action	Purpose
Step 5	**show ip rsvp sender** [**vrf** {* | *vrf-name*}][**detail**] [**filter** [**destination** *ip-addr* | *hostname*] [**source** *ip-addr* | *hostname*] [**dst-port** *port*] [**src-port** *port*]] **Example:** `Router# show ip rsvp sender detail`	Displays RSVP PATH-related sender information currently in the database. • The optional **detail** keyword displays additional output including the FLR parameters. **Note** The optional **filter** keyword is supported in Cisco IOS Releases 12.0S and 12.2S only.
Step 6	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode and returns to user EXEC mode.

Configuration Examples for RSVP FLR

This section provides configuration examples for the RSVP FLR feature.

- Configuring RSVP FLR: Example, page 8
- Verifying the RSVP FLR Configuration: Example, page 9

Configuring RSVP FLR: Example

The configuration options for RSVP FLR are the following:

- Wait time
- Number of notifications
- Repair rate

Note You can configure these options in any order.

Configuring the Wait Time

The following example configures Ethernet interface 1/0 with a bandwidth of 200 kbps and a wait time of 1000 msec:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface ethernet1/0
Router(config-if)# ip rsvp bandwidth 200
Router(config-if)# ip rsvp signalling fast-local-repair wait-time 1000
Router(config-if)# end
```

Configuring the Number of Notifications

The following example configures the number of flows that are repaired before suspending to 100:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip rsvp signalling fast-local-repair notifications 100
Router(config)# end
```

Configuring the Repair Rate

The following example configures a repair rate of 100 messages per second:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip rsvp signalling fast-local-repair rate 100
Router(config)# end
```

Verifying the RSVP FLR Configuration: Example

This section contains the following examples:

- Verifying the Details for FLR Procedures
- Verifying Configuration Details for a Specific Interface
- Verifying Configuration Details Before, During, and After an FLR Procedure

Verifying the Details for FLR Procedures

The following example displays detailed information about FLR procedures:

```
Router# show ip rsvp signalling fast-local-repair statistics detail

Fast Local Repair: enabled
  Max repair rate (paths/sec): 10
  Max processed   (paths/run): 10

FLR Statistics:
  FLR 1: DONE
    Start Time: 05:18:54 IST Mon Nov 5 2007
    Number of PSBs repaired:          2
    Used Repair Rate (msgs/sec):      10
    RIB notification processing time: 0(us).
    Time of last PSB refresh:         5025(ms).
    Time of last Resv received:       6086(ms).
    Time of last Perr received:       0(us).
    Suspend count: 0
    FLR Pacing Unit: 100 msec.
    Affected neighbors:
      Nbr Address   Interface    Relative Delay Values (msec)      VRF
      10.1.2.12         Et0/3          [5000  ,..., 5000  ]       vrfRed
      10.1.2.12         Et1/3          [5000  ,..., 5000  ]       vrfBlue
```

Verifying Configuration Details for a Specific Interface

The following example from the **show ip rsvp interface detail** command displays detailed information, including FLR, for the Ethernet 1/0 interface:

```
Router# show ip rsvp interface detail ethernet1/0

  Et1/0:
    RSVP: Enabled
    Interface State: Up
    Bandwidth:
      Curr allocated: 9K bits/sec
      Max. allowed (total): 300K bits/sec
      Max. allowed (per flow): 300K bits/sec
      Max. allowed for LSP tunnels using sub-pools (pool 1): 0 bits/sec
      Set aside by policy (total): 0 bits/sec
    Traffic Control:
      RSVP Data Packet Classification is ON via CEF callbacks
    Signalling:
```

```
    DSCP value used in RSVP msgs: 0x30
    Number of refresh intervals to enforce blockade state: 4
  FLR Wait Time (IPv4 flows):
    Repair is delayed by 1000 msec.
  Authentication: disabled
    Key chain:   <none>
    Type:        md5
    Window size: 1
    Challenge:   disabled
  Hello Extension:
    State: Disabled
```

Verifying Configuration Details Before, During, and After an FLR Procedure

The following is sample output from the **show ip rsvp sender detail** command before an FLR procedure has occurred:

```
Router# show ip rsvp sender detail

PATH:
   Destination 192.168.101.21, Protocol_Id 17, Don't Police , DstPort 1
   Sender address: 10.10.10.10, port: 1
   Path refreshes:
     arriving: from PHOP 172.3.31.34 on Et0/0 every 30000 msecs
   Traffic params - Rate: 9K bits/sec, Max. burst: 9K bytes
     Min Policed Unit: 0 bytes, Max Pkt Size 2147483647 bytes
   Path ID handle: 01000401.
   Incoming policy: Accepted. Policy source(s): Default
   Status:
   Output on Ethernet1/0. Policy status: Forwarding. Handle: 02000400
     Policy source(s): Default
   Path FLR: Never repaired
```

The following is sample output from the **show ip rsvp sender detail** command at the PLR during an FLR procedure:

```
Router# show ip rsvp sender detail

PATH:
   Destination 192.168.101.21, Protocol_Id 17, Don't Police , DstPort 1
   Sender address: 10.10.10.10, port: 1
   Path refreshes:
     arriving: from PHOP 172.16.31.34 on Et0/0 every 30000 msecs
   Traffic params - Rate: 9K bits/sec, Max. burst: 9K bytes
     Min Policed Unit: 0 bytes, Max Pkt Size 2147483647 bytes
   Path ID handle: 01000401.
   Incoming policy: Accepted. Policy source(s): Default
   Status:
   Path FLR: PSB is currently being repaired...try later
   PLR - Old Segments: 1
    Output on Ethernet1/0, nhop 172.5.36.34
    Time before expiry: 2 refreshes
    Policy status: Forwarding. Handle: 02000400
       Policy source(s): Default
```

The following is sample output from the **show ip rsvp sender detail** command at the MP during an FLR procedure:

```
Router# show ip rsvp sender detail

PATH:
   Destination 192.168.101.21, Protocol_Id 17, Don't Police , DstPort 1
   Sender address: 10.10.10.10, port: 1
   Path refreshes:
     arriving: from PHOP 172.16.37.35 on Et1/0 every 30000 msecs
Traffic params - Rate: 9K bits/sec, Max. burst: 9K bytes
     Min Policed Unit: 0 bytes, Max Pkt Size 2147483647 bytes
   Path ID handle: 09000406.
   Incoming policy: Accepted. Policy source(s): Default
   Status: Proxy-terminated
   Path FLR: Never repaired
   MP - Old Segments: 1
    Input on Serial2/0, phop 172.16.36.35
    Time before expiry: 9 refreshes
```

The following is sample output from the **show ip rsvp sender detail** command at the PLR after an FLR procedure:

```
Router# show ip rsvp sender detail

PATH:
   Destination 192.168.101.21, Protocol_Id 17, Don't Police , DstPort 1
   Sender address: 10.10.10.10, port: 1
   Path refreshes:
     arriving: from PHOP 172.16.31.34 on Et0/0 every 30000 msecs
   Traffic params - Rate: 9K bits/sec, Max. burst: 9K bytes
     Min Policed Unit: 0 bytes, Max Pkt Size 2147483647 bytes
   Path ID handle: 05000401.
   Incoming policy: Accepted. Policy source(s): Default
   Status:
   Output on Serial3/0. Policy status: Forwarding. Handle: 3B000406
     Policy source(s): Default
   Path FLR: Started 12:56:16 EST Thu Nov 16 2006, PSB repaired 532(ms) after.
       Resv/Perr: Received 992(ms) after.
```

Additional References

The following sections provide references related to the RSVP FLR feature.

Related Documents

Related Topic	Document Title
RSVP commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
QoS features including signaling, classification, and congestion management	"Quality of Service Overview" module

Standards

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

MIBs

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

RFCs

RFC	Title
RFC 2205	Resource ReSerVation Protocol (RSVP)—Version 1 Functional Specification
RFC 2209	Resource ReSerVation Protocol (RSVP)—Version 1 Messaging Processing Rules

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for RSVP FLR

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for RSVP FLR

Feature Name	Releases	Feature Information
RSVP Fast Local Repair	12.2(33)SRB, 15.0(1)M	The RSVP Fast Local Repair feature provides quick adaptation to routing changes without the overhead of the refresh period to guarantee QoS for data flows. With FLR, RSVP speeds up its response to routing changes from 30 seconds to a few seconds. This feature was integrated into Cisco IOS Release 15.0(1)M. Support for FLR in VRF domains was added. The following commands were introduced or modified: **clear ip rsvp signalling fast-local-repair statistics**, **ip rsvp signalling fast-local-repair notifications**, **ip rsvp signalling fast-local-repair rate**, **ip rsvp signalling fast-local-repair wait-time**, **show ip rsvp**, **show ip rsvp interface**, **show ip rsvp sender**, **show ip rsvp signalling fast-local-repair**.

Glossary

admission control—The process by which an RSVP reservation is accepted or rejected on the basis of end-to-end available network resources.

bandwidth—The difference between the highest and lowest frequencies available for network signals. The term is also used to describe the rated throughput capacity of a given network medium or protocol.

message pacing—A system for managing volume and timing that permits messages from multiple sources to be spaced apart over time. RSVP message pacing maintains, on an outgoing basis, a count of the messages that it has been forced to drop because the output queue for the interface used for the message pacing was full.

MP—merge point. The node where the new and old FLR segments meet.

PLR—point of local repair. The node that initiates an FLR procedure.

QoS—quality of service. A measure of performance for a transmission system that reflects its transmission quality and service availability.

RSVP—Resource Reservation Protocol. A protocol that supports the reservation of resources across an IP network. Applications running on IP end systems can use RSVP to indicate to other nodes the nature (bandwidth, jitter, maximum burst, and so on) of the packet streams that they want to receive.

VRF—Virtual Routing and Forwarding. VRF is A VPN routing and forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a PE router.

RSVP Interface-Based Receiver Proxy

First Published: July 10, 2006
Last Updated: October 2, 2009

The RSVP Interface-Based Receiver Proxy feature lets you configure a proxy router by outbound interface instead of configuring a destination address for each flow going through the same interface.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "Feature Information for RSVP Interface-Based Receiver Proxy" section on page 12.

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

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for RSVP Interface-Based Receiver Proxy

You must configure an IP address and enable Resource Reservation Protocol (RSVP) on one or more interfaces on at least two neighboring routers that share a link within the network.

Restrictions for RSVP Interface-Based Receiver Proxy

- Filtering using access control lists (ACLs), application IDs, or other mechanisms is not supported.
- A provider edge (PE) router cannot switch from being a proxy node to a transit node for a given flow during the lifetime of the flow.

Information About RSVP Interface-Based Receiver Proxy

To use the RSVP Interface-Based Receiver Proxy feature, you should understand the following concepts:

- Feature Overview of RSVP Interface-Based Receiver Proxy, page 2
- Benefits of RSVP Interface-Based Receiver Proxy, page 3

Feature Overview of RSVP Interface-Based Receiver Proxy

The RSVP Interface-Based Receiver Proxy feature allows you to use RSVP to signal reservations and guarantee bandwidth on behalf of a receiver that does not support RSVP, by terminating the PATH message and generating a RESV message in the upstream direction on an RSVP-capable router on the path to the endpoint. An example is a video-on-demand flow from a video server to a set-top box, which is a computer that acts as a receiver and decodes the incoming video signal from the video server.

Because set-top boxes may not support RSVP natively, you cannot configure end-to-end RSVP reservations between a video server and a set-top box. Instead, you can enable the RSVP interface-based receiver proxy on the router that is closest to that set-top box.

The router terminates the end-to-end sessions for many set-top boxes and performs admission control on the outbound (or egress) interface of the PATH message, where the receiver proxy is configured, as a proxy for Call Admission Control (CAC) on the router-to-set-top link. The RSVP interface-based receiver proxy determines which PATH messages to terminate by looking at the outbound interface to be used by the traffic flow.

You can configure an RSVP interface-based receiver proxy to terminate PATH messages going out a specified interface with a specific action (reply with RESV, or reject). The most common application is to configure the receiver proxy on the edge of an administrative domain on interdomain interfaces. The router then terminates PATH messages going out the administrative domain while still permitting PATH messages transitioning through the router within the same administrative domain to continue downstream.

In the video-on-demand example described above, the last-hop Layer 3 router supporting RSVP implements the receiver proxy, which is then configured on the interfaces facing the Layer 2 distribution network (for example, Digital Subscriber Line access [DSLAM] or cable distribution). Also, since RSVP is running and performing CAC on the router with the receiver proxy, you can configure RSVP enhancements such as local policy and Common Open Policy Service (COPS) for more fine-grained control on video flow CAC.

The router terminates the end-to-end sessions for many set-top boxes, with the assumption that the links further downstream (for example, from the DSLAM to the set-top box) never become congested or, more likely, in the case of congestion, that the voice and video traffic from the router gets the highest priority and access to the bandwidth.

Benefits of RSVP Interface-Based Receiver Proxy

Ease of Use and Scalability Improvement

Previously, you had to configure a receiver proxy for every separate RSVP stream or set-top box. Now you can configure the proxy by outbound interface. For example, if there were 100 set-top boxes downstream from the proxy router, you had to configure 100 proxies. With this enhancement, you configure only the outbound interface(s). In addition, the receiver proxy is guaranteed to terminate the reservation only on the last hop within the core network. Nodes that may function as transit nodes for some PATH messages but should proxy others depending on their placement in the network can perform the correct functions on a flow-by-flow basis.

In the video-on-demand example described above, a PATH message that transits through an edge router to another edge router (around the edge) is not terminated, whereas an otherwise identical PATH message that actually exits the aggregation network and transitions to the access network is terminated. This allows for more accurate CAC in the network and also simplifies and reduces configuration requirements.

How to Configure RSVP Interface-Based Receiver Proxy

This section contains the following procedures:

- Enabling RSVP on an Interface, page 3 (required)
- Configuring a Receiver Proxy on an Outbound Interface, page 4 (required)
- Verifying the RSVP Interface-Based Receiver Proxy Configuration, page 5 (optional)

Enabling RSVP on an Interface

Perform this task to enable RSVP on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface number*
4. **ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*] [**sub-pool** [*sub-pool-kbps*]]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface number* **Example:** `Router(config)# interface Ethernet0/0`	Configures the interface type and enters interface configuration mode.
Step 4	**ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*] [**sub-pool** [*sub-pool-kbps*]] **Example:** `Router(config-if)# ip rsvp bandwidth 7500 7500`	Enables RSVP on an interface. • The optional *interface-kbps* and *single-flow-kbps* arguments specify the amount of bandwidth that can be allocated by RSVP flows or to a single flow, respectively. Values are from 1 to 10000000. • The optional **sub-pool** and *sub-pool-kbps* keyword and argument specify subpool traffic and the amount of bandwidth that can be allocated by RSVP flows. Values are from 1 to 10000000. **Note** Repeat this command for each interface on which you want to enable RSVP.
Step 5	**end** **Example:** `Router(config-if)# end`	(Optional) Returns to privileged EXEC mode.

Configuring a Receiver Proxy on an Outbound Interface

Perform this task to configure a receiver proxy on an outbound interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface number*
4. **ip rsvp listener outbound** {**reply** | **reject**}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface number* **Example:** `Router(config)# interface Ethernet0/0`	Configures the interface type and enters interface configuration mode.
Step 4	**ip rsvp listener outbound** {**reply** | **reject**} **Example:** `Router(config-if)# ip rsvp listener outbound reject`	Configures an RSVP router to listen for PATH messages sent through a specified interface. • Enter the **reply** keyword or the **reject** keyword to specify the response that you want to PATH messages.
Step 5	**end** **Example:** `Router(config-if)# end`	(Optional) Returns to privileged EXEC mode.

Verifying the RSVP Interface-Based Receiver Proxy Configuration

Perform the following task to verify the configuration.

Note You can use the following **show** commands in user EXEC or privileged EXEC mode.

SUMMARY STEPS

1. **enable**
2. **show ip rsvp listeners** [*dst* | **any** | **vrf** {***** | *vrf-name*}] [**udp** | **tcp** | **any** | *protocol*] [*dst-port* | **any**]
3. **show ip rsvp sender** [**vrf** {***** | *vrf-name*}] [**detail**] [**filter** [**destination** *ip-addr* | *hostname*] [**source** *ip-addr* | *hostname*] [**dst-port** *port*] [**src-port** *port*]]
4. **show ip rsvp reservation** [**vrf** {***** | *vrf-name*}] [**detail**] [**filter** [**destination** *ip-addr* | *hostname*] [**source** *ip-addr* | *hostname*] [**dst-port** *port*] [**src-port** *port*]]
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	(Optional) Enables privileged EXEC mode. • Enter your password if prompted. **Note** Skip this step if you are using the **show** commands in user EXEC mode.
Step 2	**show ip rsvp listeners** [*dst* | **any** | **vrf** {* | *vrf-name*}] [**udp** | **tcp** | **any** | *protocol*] [*dst-port* | **any**] **Example:** `Router# show ip rsvp listeners`	Displays RSVP listeners for a specified port or protocol.
Step 3	**show ip rsvp sender** [**vrf** {* | *vrf-name*}] [**detail**] [**filter** [**destination** *ip-addr* | *hostname*] [**source** *ip-addr* | *hostname*] [**dst-port** *port*] [**src-port** *port*]] **Example:** `Router# show ip rsvp sender detail`	Displays RSVP PATH-related sender information currently in the database. • The optional **detail** keyword displays additional output. **Note** The optional **filter** keyword is supported in Cisco IOS Releases 12.0S and 12.2S only.
Step 4	**show ip rsvp reservation** [**vrf** {* | *vrf-name*}] [**detail**] [**filter** [**destination** *ip-addr* | *hostname*] [**source** *ip-addr* | *hostname*] [**dst-port** *port*] [**src-port** *port*]] **Example:** `Router# show ip rsvp reservation detail`	Displays RSVP-related receiver information currently in the database. • The optional **detail** keyword displays additional output. **Note** The optional **filter** keyword is supported in Cisco IOS Releases 12.0S and 12.2S only.
Step 5	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode and returns to user EXEC mode.

Configuration Examples for RSVP Interface-Based Receiver Proxy

This section provides the following configuration examples:

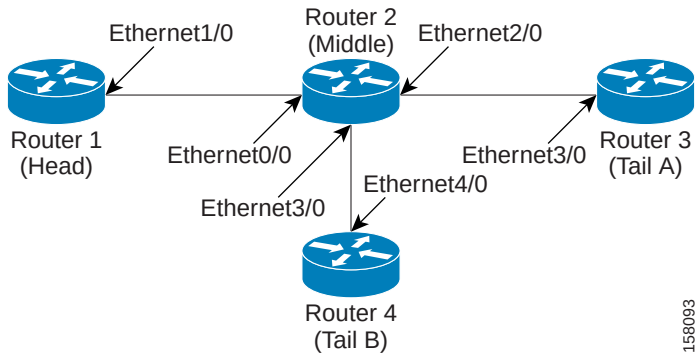
- Configuring RSVP Interface-Based Receiver Proxy: Examples, page 6
- Verifying RSVP Interface-Based Receiver Proxy: Examples, page 7

Configuring RSVP Interface-Based Receiver Proxy: Examples

The four-router network in Figure 1 contains the following configurations:

- Configuring a Receiver Proxy (Listener) on a Middle Router on Behalf of Tailend Routers, page 7
- Configuring PATH Messages from a Headend Router to Tailend Routers to Test the Receiver Proxy, page 7

Figure 1 Sample Network with an Interface-Based Receiver Proxy Configured



Configuring a Receiver Proxy (Listener) on a Middle Router on Behalf of Tailend Routers

The following example configures a receiver proxy, also called a listener, on the middle router (Router 2) on behalf of the two tailend routers (Routers 3 and 4):

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface ethernet 2/0
Router(config-if)# ip rsvp listener outbound reply
Router(config-if)# exit
Router(config)# interface ethernet 3/0
Router(config-if)# ip rsvp listener outbound reject
Router(config-if)# end
```

Configuring PATH Messages from a Headend Router to Tailend Routers to Test the Receiver Proxy

Note If you do not have another headend router generating RSVP PATH messages available, configure one in the network for the specific purpose of testing RSVP features such as the receiver proxy. Note that these commands are not expected (or supported) in a final deployment.

The following example configures four PATH messages from the headend router (Router 1) to the tailend routers (Routers 3 and 4):

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip rsvp sender-host 10.0.0.5 10.0.0.1 TCP 2 2 100 10
Router(config)# ip rsvp sender-host 10.0.0.5 10.0.0.1 UDP 1 1 100 10
Router(config)# ip rsvp sender-host 10.0.0.7 10.0.0.1 TCP 4 4 100 10
Router(config)# ip rsvp sender-host 10.0.0.7 10.0.0.1 UDP 3 3 100 10
Router(config)# end
```

Verifying RSVP Interface-Based Receiver Proxy: Examples

This section contains the following verification examples:

Verifying the PATH Messages in the Database

The following example verifies that the PATH messages you configured are in the database:

```
Router# show ip rsvp sender

To              From            Pro DPort Sport Prev Hop        I/F      BPS
10.0.0.5        10.0.0.1        TCP 2     2     none            none     100K
10.0.0.5        10.0.0.1        UDP 1     1     none            none     100K
10.0.0.7        10.0.0.1        TCP 4     4     none            none     100K
10.0.0.7        10.0.0.1        UDP 3     3     none            none     100K
```

The following example verifies that a PATH message has been terminated by a receiver proxy configured to reply.

Note A receiver proxy that is configured to reject does not cause any state to be stored in the RSVP database; therefore, this **show** command does not display these PATHS. Only one PATH message is shown.

```
Router# show ip rsvp sender detail

PATH:
  Destination 10.0.0.5, Protocol_Id 17, Don't Police , DstPort 1
  Sender address: 10.0.0.1, port: 1
  Path refreshes:
    arriving: from PHOP 10.1.2.1 on Et0/0 every 30000 msecs
  Traffic params - Rate: 100K bits/sec, Max. burst: 10K bytes
    Min Policed Unit: 0 bytes, Max Pkt Size 2147483647 bytes
  Path ID handle: 01000402.
  Incoming policy: Accepted. Policy source(s): Default
  Status: Proxy-terminated
  Output on Ethernet2/0. Policy status: NOT Forwarding. Handle: 02000401
    Policy source(s):
  Path FLR: Never repaired
```

Verifying the Running Configuration

The following example verifies the configuration for Ethernet interface 2/0:

```
Router# show running-config interface Ethernet2/0

Building configuration...

Current configuration : 132 bytes
!
interface Ethernet2/0
 ip address 172.16.0.1 255.0.0.0
 no cdp enable
 ip rsvp bandwidth 2000
 ip rsvp listener outbound reply
end
```

The following example verifies the configuration for Ethernet interface 3/0:

```
Router# show running-config interface Ethernet3/0

Building configuration...

Current configuration : 133 bytes
!
interface Ethernet3/0
 ip address 172.16.0.2 255.0.0.0
```

```
 no cdp enable
 ip rsvp bandwidth 2000
 ip rsvp listener outbound reject
end
```

Verifying the Listeners (Proxies)

The following example verifies the listeners (proxies) that you configured on the middle router (Router 2) on behalf of the two tailend routers (Routers 3 and 4):

```
To                Protocol   DPort   Description                Action    OutIf
10.0.0.0          0          0       RSVP Proxy                 reply     Et2/0
10.0.0.0          0          0       RSVP Proxy                 reject    Et3/0
```

Verifying the Reservations

The following example displays reservations established by the middle router (Router 2) on behalf of the tailend routers (Routers 3 and 4) as seen from the headend router (Router 1):

```
Router# show ip rsvp reservation

To            From          Pro DPort Sport Next Hop      I/F      Fi Serv BPS
10.0.0.7      10.0.0.1      TCP 4     4     10.0.0.2      Et1/0    FF RATE 100K
10.0.0.7      10.0.0.1      UDP 3     3     10.0.0.2      Et1/0    FF RATE 100K
```

The following example verifies that a reservation is locally generated (proxied). Only one reservation is shown:

```
Router# show ip rsvp reservation detail

RSVP Reservation. Destination is 10.0.0.7, Source is 10.0.0.1,
  Protocol is UDP, Destination port is 1, Source port is 1
  Next Hop: 10.2.3.3 on Ethernet2/0
  Reservation Style is Fixed-Filter, QoS Service is Guaranteed-Rate
  Resv ID handle: 01000405.
  Created: 09:24:24 EST Fri Jun 2 2006
  Average Bitrate is 100K bits/sec, Maximum Burst is 10K bytes
  Min Policed Unit: 0 bytes, Max Pkt Size: 0 bytes
  Status: Proxied
  Policy: Forwarding. Policy source(s): Default
```

Verifying CAC on an Outbound Interface

The following example verifies that the proxied reservation performed CAC on the local outbound interface:

```
Router# show ip rsvp installed

RSVP: Ethernet3/0 has no installed reservations
RSVP: Ethernet2/0
BPS    To              From            Protoc DPort  Sport
100K   10.0.0.7        10.0.0.1        UDP    1       1
```

Additional References

The following sections provide references related to the RSVP Interface-Based Receiver Proxy feature.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
QoS configuration tasks related to RSVP	"Configuring RSVP" module
Internet draft	*RSVP Proxy Approaches*, Internet draft, October 2006 [draft-lefaucheur-tsvwg-rsvp-proxy-00.txt]

Standards

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

MIBs

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

RFCs

RFC	Title
RFC 2205	Resource ReSerVation Protocol (RSVP)

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for RSVP Interface-Based Receiver Proxy

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for RSVP Interface-Based Receiver Proxy

Feature Name	Releases	Feature Information
RSVP Interface-Based Receiver Proxy	12.2(28)SXF5 12.2(33)SRB, 15.0(1)M	The RSVP Interface-Based Receiver Proxy feature lets you configure a proxy router by outbound interface instead of configuring a destination address for each flow going through the same interface. In Cisco IOS Release 12.2(33)SRB, support was added for the Cisco 7600 series routers. This feature was integrated into Cisco IOS Release 15.0(1)M. The following commands were introduced or modified: **ip rsvp bandwidth**, **ip rsvp listener outbound**, **show ip rsvp listeners**, **show ip rsvp reservation**, **show ip rsvp sender**.

Glossary

flow—A stream of data traveling between two endpoints across a network (for example, from one LAN station to another). Multiple flows can be transmitted on a single circuit.

PE router—provider edge router. A router that is part of a service provider's network and is connected to a customer edge (CE) router.

proxy—A component of RSVP that manages all locally originated and terminated state.

receiver proxy—A configurable feature that allows a router to proxy RSVP RESV messages for local or remote destinations.

RSVP—Resource Reservation Protocol. A protocol for reserving network resources to provide quality of service guarantees to application flows.

set-top box—A computer that acts as a receiver and decodes the incoming signal from a satellite dish, a cable network, or a telephone line.

RSVP Aggregation

First Published: January 7, 2008
Last Updated: January 7, 2008

The RSVP Aggregation feature allows the Resource Reservation Protocol (RSVP) state to be reduced within an RSVP/DiffServ network by aggregating many smaller reservations into a single, larger reservation at the edge.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your Cisco IOS software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for RSVP Aggregation" section on page 30.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for RSVP Aggregation

You must configure at least two aggregating nodes (provider edge [PE] devices), one interior node (provider [P] device) and two end user nodes (customer edge [CE] devices) within your network.

You must configure your network to support the following Cisco IOS features:

- RSVP
- Class Based Weighted Fair Queuing (CBWFQ)
- RSVP Scalability Enhancements

Note You configure these features because Cisco IOS Release 12.2(33)SRC supports control plane aggregation only. Dataplane aggregation must be achieved by using the RSVP Scalability Enhancements.

Restrictions for RSVP Aggregation

Functionality Restrictions

The following functionality is not supported:

- Multilevel aggregation
- Multiple, adjacent aggregation regions
- Dynamic resizing of aggregate reservations
- Policing of end-to-end (E2E) reservations by the aggregator
- Policing of aggregate reservations by interior routers
- Differentiated Services Code Point (DSCP) marking by the aggregator
- Equal Cost Multiple Paths (ECMP) load-balancing within the aggregation region
- RSVP Fast Local Repair in case of a routing change resulting in a different aggregator or deaggregator, admission control is performed on E2E PATH refresh
- Multicast RSVP reservations
- RSVP policy servers including Common Open Policy Server (COPS)
- Dataplane aggregation

The following functionality is supported:

- Multiple, non-adjacent aggregation regions
- Control plane aggregation

Note RSVP/DiffServ using CBWFQ provides the dataplane aggregation.

Configuration Restrictions

- Sources should not send marked packets without an installed reservation.
- Sources should not send marked packets that exceed the reserved bandwidth.
- Sources should not send marked packets to a destination other than the reserved path.

- All RSVP capable routers within an aggregation region regardless of role must support the aggregation feature to recognize the RFC 3175 RSVP message formats properly.
- E2E reservations must be present to establish dynamic aggregates; aggregates cannot be established manually.
- Aggregates are established at a fixed bandwidth regardless of the number of current E2E reservations being aggregated.
- Aggregators and deaggregators must be paired to avoid blackholing of E2E reservations because of dynamic aggregate establishment.

Note Blackholing means that the reservation is never established. If an E2E reservation crosses from an exterior to an interior interface, the E2E reservation turns into an RSVP-E2E-IGNORE protocol packet. If there is no corresponding deaggregator, a router where this RSVP-E2E-IGNORE reservation crosses an interior to an exterior interface, then the RSVP-E2E-IGNORE reservation is never restored to an E2E reservation. The RSVP-E2E-IGNORE reservation eventually reaches its destination, which is the RSVP receiver; however, the RSVP receiver does not know what to do with the RSVP-E2E-IGNORE reservation and discards the packet.

Information About RSVP Aggregation

To use the RSVP Aggregation feature, you should understand the following concepts:

- Feature Overview of RSVP Aggregation, page 3
- Benefits of RSVP Aggregation, page 6

Feature Overview of RSVP Aggregation

This section provides the following information:

- High Level Overview, page 3
- How Aggregation Functions, page 4
- Integration with RSVP Features, page 6

High Level Overview

The establishment of a single RSVP reservation requires a large amount of resources including memory allocated for the associated data structures, CPU for handling signaling messages, I/O operations for datapath programming, interprocess communication, and signaling message transmission.

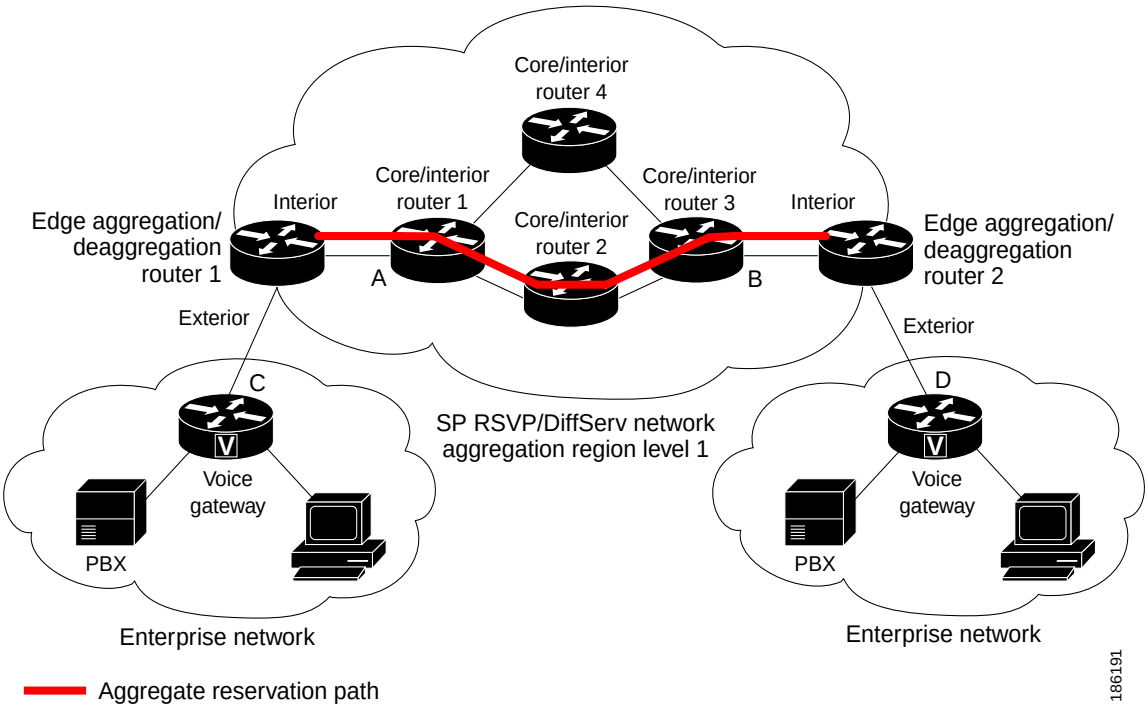
When a large number of small reservations are established, the resources required for setting and maintaining these reservations may exceed a node's capacity to the point where the node's performance is significantly degraded or it becomes unusable. The RSVP Aggregation feature addresses this scalability issue by introducing flow aggregation.

Flow aggregation is a mechanism wherein RSVP state can be reduced within a core router by aggregating many smaller reservations into a single, larger reservation at the network edge. This preserves the ability to perform connection admission control on core router links within the RSVP/DiffServ network while reducing signaling resource overhead.

How Aggregation Functions

Common segments of multiple end-to-end (E2E) reservations are aggregated over an aggregation region into a larger reservation that is called an aggregate reservation. An aggregation region is a connected set of nodes that are capable of performing RSVP aggregation as shown in Figure 1.

Figure 1 RSVP Aggregation Network Overview



There are three types of nodes within an aggregation region:

- Aggregator—Aggregates multiple E2E reservations.
- Deaggregator—Deaggregates E2E reservations; provides mapping of E2E reservations onto aggregates.
- Interior—Neither aggregates or deaggregates, but is an RSVP core router that understands RFC 3175 formatted RSVP messages. Core/interior routers 1 through 4 are examples shown in Figure 1.

There are two types of interfaces on the aggregator/deaggregator nodes:

- Exterior interface—The interface is not part of the aggregate region.
- Interior interface—The interface is part of the aggregate region.

Any router that is part of the aggregate region must have at least one interior interface and may have one or more exterior interfaces. Depending on the types of interfaces spanned by an IPv4 flow, a node can be an aggregator, a deaggregator, or an interior router with respect to that flow.

Aggregate RSVP/DiffServ Integration Topology

RSVP aggregation further enhances RSVP scalability within an RSVP/DiffServ network as shown in Figure 1 by allowing the establishment of aggregate reservations across an aggregation region. This allows for aggregated connection admission control on core/interior router interfaces. Running RSVP on the core/interior routers allows for more predictable bandwidth use during normal and failure scenarios.

The voice gateways are running classic RSVP, which means RSVP is keeping a state per flow and also classifying, marking, and scheduling packets on a per-flow basis. The edge/aggregation routers are running RSVP with scalability enhancements for admission control on the exterior interfaces connected to the voice gateways and running RSVP aggregation on the interfaces connected to core/interior routers 1 and 3. The core/interior routers in the RSVP/DiffServ network are running RSVP for the establishment of the aggregate reservations. The edge and core/interior routers inside the RSVP/DiffServ network also implement a specific per hop behavior (PHB) for a collection of flows that have the same DSCP.

The voice gateways identify voice data packets and set the appropriate DSCP in their IP headers so that the packets are classified into the priority class in the edge/aggregation routers and in core/interior routers 1, 2, 3 or 1, 4, 3.

The interior interfaces on the edge/aggregation/deaggregation routers (labeled A and B) connected to core/interior routers 1 and 3are running RSVP aggregation. They are performing admission control only per flow against the RSVP bandwidth of the aggregate reservation for the corresponding DSCP.

Admission control is performed at the deaggregator because it is the first edge node to receive the returning E2E RSVP RESV message. CBWFQ is performing the classification, policing, and scheduling functions on all nodes within the RSVP/DiffServ network including the edge routers.

Aggregate reservations are dynamically established over an aggregation region when an E2E reservation enters an aggregation region by crossing from an exterior to an interior interface; for example, when voice gateway C initiates an E2E reservation to voice gateway D. The aggregation is accomplished by "hiding" the E2E RSVP messages from the RSVP nodes inside the aggregation region. This is achieved with a new IP protocol, RSVP-E2E-IGNORE, that replaces the standard RSVP protocol in E2E PATH, PATHTEAR, and RESVCONF messages. This protocol change to RSVP-E2E-IGNORE is performed by the aggregator when the message enters the aggregation region and later restored back to RSVP by the deaggregator when the message exits the aggregation region. Thus, the aggregator and deaggregator pairs for a given flow are dynamically discovered during the E2E PATH establishment.

The deaggregator router 2 is responsible for mapping the E2E PATH onto an aggregate reservation per the configured policy. If an aggregate reservation with the corresponding aggregator router 1 anda DSCP is established, the E2E PATH is forwarded. Otherwise a new aggregate at the requisite DSCP is established, and then the E2E PATH is forwarded. The establishment of this new aggregate is for the fixed bandwidth parameters configured at the deaggregator router 2. Aggregate PATH messages are sent from the aggregator to the deaggregator using RSVP's normal IP protocol. Aggregate RESV messages are sent back from the deaggregator to the aggregator, thus establishing an aggregate reservation on behalf of the set of E2E flows that use this aggregator and deaggregator. All RSVP capable interior nodes process the aggregate reservation request following normal RSVP processing including any configured local policy.

The RSVP-E2E-IGNORE messages are ignored by the core/interior routers, no E2E reservation states are created, and the message is forwarded as IP. As a consequence, the previous hop/next hop (PHOP/NHOP) for each RSVP-E2E-IGNORE message received at the deaggregator or aggregator is the aggregator or deaggregator node. Therefore, all messages destined to the next or previous hop (RSVP error messages, for example) do not require the protocol to be changed when they traverse the aggregation region.

By setting up a small number of aggregate reservations on behalf of a large number of E2E flows, the number of states stored at core/interior routers and the amount of signal processing within the aggregation region is reduced.

In addition, by using differentiated services mechanisms for classification and scheduling of traffic supported by aggregate reservations rather than performing per aggregate reservation classification and scheduling, the amount of classification and scheduling state in the aggregation region is further reduced. This reduction is independent of the number of E2E reservations and the number of aggregate reservations in the aggregation region. One or more RSVP/DiffServ DSCPs are used to identify the traffic covered by aggregate reservations, and one or more RSVP/DiffServ per hop behaviors (PHBs) are

used to offer the required forwarding treatment to this traffic. There may be more than one aggregate reservation between the same pair of routers, each representing different classes of traffic and each using a different DSCP and a different PHB.

Integration with RSVP Features

RSVP aggregation has been integrated with many RSVP features, including the following:

- RSVP Fast Local Repair
- RSVP Local Policy Support
- RSVP Refresh Reduction and Reliable Messaging

Benefits of RSVP Aggregation

Enhanced Scalability

Aggregating a large number of small reservations into one reservation requires fewer resources for signaling, setting, and maintaining the reservation thereby increasing scalability.

Enhanced Bandwidth Usage within RSVP/DiffServ Core Network

Aggregate reservations across an RSVP/DiffServ network allow for more predictable bandwidth use of core links across RSVP/DiffServ PHBs. Aggregate reservations can use RSVP fast local repair and local policy preemption features for determining bandwidth use during failure scenarios.

How to Configure RSVP Aggregation

This section contains the following procedures:

- Configuring RSVP Scalability Enhancements, page 7 (required)
- Configuring Interfaces with Aggregation Role, page 14 (required)
- Configuring Aggregation Mapping on a Deaggregator, page 15 (required)
- Configuring Aggregate Reservation Attributes on a Deaggregator, page 17 (required)
- Configuring an RSVP Aggregation Router ID, page 18 (required)
- Enabling RSVP Aggregation, page 19 (required)
- Configuring RSVP Local Policy, page 20 (optional)
- Verifying the RSVP Aggregation Configuration, page 22 (optional)

Configuring RSVP Scalability Enhancements

Note All interfaces on nodes running Cisco IOS Release 12.2(33)SRC software must be configured with RSVP Scalability Enhancements.

Note Interior nodes only require RSVP Scalability Enhancements (RSVP/DiffServ) configuration. Interior nodes simply need to have RSVP/DiffServ configured and be running Cisco IOS Release 12.2(33)SRC with RSVP aggregation support to enable the nodes to process per normal RSVP processing rules RFC 3175 formatted messages properly. This is because Cisco IOS Release 12.2(33)SRC supports control plane aggregation only. Dataplane aggregation must be achieved by using the RSVP Scalability Enhancements.

Perform these tasks on all nodes within the aggregation region including aggregators, deaggregators, and interior nodes.

This section includes the following procedures:

- Enabling RSVP on an Interface, page 7 (required)
- Setting the Resource Provider, page 8 (required)
- Disabling Data Packet Classification, page 9 (required)
- Configuring Class and Policy Maps, page 10 (required)
- Attaching a Policy Map to an Interface, page 13 (required)

Enabling RSVP on an Interface

Perform this task to enable RSVP on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet0/0`	Configures the interface type and enters interface configuration mode.
Step 4	**ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*] **Example:** `Router(config-if)# ip rsvp bandwidth 7500`	Enables RSVP bandwidth on an interface. • The optional *interface-kbps* and *single-flow-kbps* arguments specify the amount of bandwidth that can be allocated by RSVP flows or to a single flow, respectively. Values are from 1 to 10000000. **Note** Repeat this command for each interface that you want to enable.
Step 5	**end** **Example:** `Router(config-if)# end`	(Optional) Returns to privileged EXEC mode.

Setting the Resource Provider

Perform this task to set the resource provider.

Note Resource provider was formerly called QoS provider.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip rsvp resource-provider none** [**none** | **wfq-interface** | **wfq-pvc**]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet0/0`	Configures the interface type and enters interface configuration mode.
Step 4	**ip rsvp resource-provider** [**none** | **wfq-interface** | **wfq-pvc**] **Example:** `Router(config-if)# ip rsvp resource-provider none`	Sets the resource provider. • Enter the optional **none** keyword to set the resource provider to none regardless of whether one is configured on the interface. **Note** Setting the resource provider to **none** instructs RSVP to *not* associate any resources, such as weighted fair queueing (WFQ) queues or bandwidth, with a reservation. • Enter the optional **wfq-interface** keyword to specify WFQ as the resource provider on the interface. • Enter the optional **wfq-pvc** keyword to specify WFQ as the resource provider on the permanent virtual circuit (PVC) or connection.
Step 5	**end** **Example:** `Router(config-if)# end`	(Optional) Returns to privileged EXEC mode.

Disabling Data Packet Classification

Perform this task to disable data packet classification.

Note Disabling data packet classification instructs RSVP not to process every packet, but to perform admission control only.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*

4. **ip rsvp data-packet classification none**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet0/0`	Configures the interface type and enters interface configuration mode.
Step 4	**ip rsvp data-packet classification none** **Example:** `Router(config-if)# ip rsvp data-packet classification none`	Disables data packet classification.
Step 5	**end** **Example:** `Router(config-if)# end`	(Optional) Returns to privileged EXEC mode.

Configuring Class and Policy Maps

Perform this task to configure class and policy maps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** [**type** {**stack** | **access-control** | **port-filter** | **queue-threshold**}] [**match-all** | **match-any**] *class-map-name*
4. **match access-group** {*access-group* | **name** *access-group-name*}
5. **exit**
6. **policy-map** [**type access-control**] *policy-map-name*
7. **class** {*class-name* | **class-default**}
8. **priority** {*bandwidth-kbps* | **percent** *percentage*} [*burst*]
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map** [**type** {**stack** | **access-control** | **port-filter** | **queue-threshold**}] [**match-all** | **match-any**] *class-map-name* **Example:** `Router(config)# class-map match-all voice`	Creates a class map to be used for matching packets to a specified class and enters class-map configuration mode. • The optional **type stack** keywords enable the flexible packet matching (FPM) functionality to determine the correct protocol stack in which to examine. **Note** If the appropriate protocol header description files (PHDFs) have been loaded onto the router (via the **load protocol** command), a stack of protocol headers can be defined so the filter can determine which headers are present and in what order. • The optional **type access-control** keywords determine the exact pattern to look for in the protocol stack of interest. **Note** You must specify a stack class map (via the **type stack** keywords) before you can specify an access-control class map (via the type **access-control** keywords). • The optional **type port-filter** keywords create a port-filter class-map that enables the TCP/UDP port policing of control plane packets. **Note** When enabled, these keywords provide filtering of traffic destined to specific ports on the control plane host subinterface. • The optional **type queue-threshold** keywords enable queue thresholding that limits the total number of packets for a specified protocol that is allowed in the control plane IP input queue. This feature applies only to control plane host subinterface. • The optional **match-all** | **match-any** keywords determine how packets are evaluated when multiple match criteria exist. Packets must either meet all of the match criteria (**match-all**) or one of the match criteria (**match-any**) in order to be considered a member of the class.

	Command or Action	Purpose
Step 4	**match access-group** {*access-group* | **name** *access-group-name*} **Example:** `Router(config-cmap)# match access-group 100`	Specifies the numbered access list against whose contents packets are checked to determine if they match the criteria specified in the class map. **Note** After you create the class map, you configure its match criteria. Here are some of the commands that you can use: – **match access-group** – **match input-interface** – **match mpls experimental** – **match protocol**
Step 5	**exit** **Example:** `Router(config-cmap)# exit`	Exits to global configuration mode.
Step 6	**policy-map** [**type access-control**] *policy-map-name* **Example:** `Router(config)# policy-map wfq-voip`	Creates or modifies a policy map that can be attached toone or more interfaces to specify a service policy and enters policy-map configuration mode. • The optional **type access-control** keywords determine the exact pattern to look for in the protocol stack of interest.
Step 7	**class** {*class-name* | **class-default**} **Example:** `Router(config-pmap-c)# class voice`	Specifies the class so that you can configure or modify its policy. Enters policy-map class configuration mode. • Enter the *class name* or use the **class-default** keyword.
Step 8	**priority** {*bandwidth-kbps* | **percent** *percentage*} [*burst*] **Example:** `Router(config-pmap-c)# priority 24`	(Optional) Prioritizes a class of traffic belonging to a policy map. • The optional *burst* argument specifies the burst size in bytes. The burst size configures the network to accommodate temporary bursts of traffic. The default burst value, which is computed as 200 milliseconds of traffic at the configured bandwidth rate, is used when the *burst* argument is not specified. The range of the burst is from 32 to 2000000 bytes.
Step 9	**end** **Example:** `Router(config--pmap-c)# end`	(Optional) Returns to privileged EXEC mode.

Attaching a Policy Map to an Interface

Perform this task to attach a policy map to an interface.

Note If at the time you configure the RSVP scalability enhancements, there are existing reservations that use classic RSVP, no additional marking, classification, or scheduling is provided for these flows. You can also delete these reservations after you configure the RSVP scalability enhancements.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **service-policy** [**type access-control**] {**input** | **output**} *policy-map-name*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`interface` *type number* **Example:** `Router(config)# interface Ethernet0/0`	Configures the interface type and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**service-policy** [**type access-control**] {**input** | **output**} *policy-map-name* **Example:** `Router(config-if)# service-policy output POLICY-ATM`	Specifies the name of the policy map to be attached to the input or output direction of the interface. **Note** Policy maps can be attached in the input or output direction of an interface. The direction and the router to which the policy map should be attached vary according to the network configuration. When using the **service-policy** command to attach the policy map to an interface, be sure to choose the router and the interface direction that are appropriate for the network configuration. • The optional **type access-control** keywords determine the exact pattern to look for in the protocol stack of interest. • Enter the *policy-map name*.
Step 5	**end** **Example:** `Router(config-if)# end`	(Optional) Returns to privileged EXEC mode.

Configuring Interfaces with Aggregation Role

Perform this task on aggregator and deaggregators to specify which interfaces are facing the aggregation region.

Note You do not need to perform this task on interior routers; that is, nodes having interior interfaces only.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip rsvp aggregation role interior**
5. Repeat Step 4 for each of the aggregator and deaggregator's interfaces that are facing the aggregation region.
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet0/0`	Configures the interface type and enters interface configuration mode.
Step 4	**ip rsvp aggregation role interior** **Example:** `Router(config-if)# ip rsvp aggregation role interior`	Enables RSVP aggregation on an aggregator or deaggregator's interface.
Step 5	Repeat Step 4 as needed to configure additional aggregator and deaggregator interfaces.	Configures additional aggregator and deaggregator interfaces.
Step 6	**end** **Example:** `Router(config-if)# end`	(Optional) Returns to privileged EXEC mode.

Configuring Aggregation Mapping on a Deaggregator

Perform this task to configure aggregation mapping on a deaggregator.

Note Typically, an edge router acts as both an aggregator and deaggregator because of the unidirectional nature of RSVP reservations. Most applications require bidirectional reservations. Therefore, these parameters are used by a deaggregator when mapping E2E reservations onto aggregates during the dynamic aggregate reservation process.

Prerequisites

You should configure an access control list (ACL) to define a group of RSVP endpoints whose reservations will be aggregated onto a single aggregate reservation session identified by the specified DSCP. Then for each ACL, define a map configuration.

Note In classic (unaggregated) RSVP, a session is identified in the reservation message session object by the destination IP address and protocol information. In RSVP aggregation, a session is identified by the destination IP address and DSCP within the session object of the aggregate RSVP message. E2E reservations are mapped onto a particular aggregate RSVP session identified by the E2E reservation session object alone or a combination of the session object and sender template or filter spec.

Extended ACLs

The ACLs used within the **ip rsvp aggregation ip map** command match the RSVP message objects as follows for an extended ACL:

- Source IP address and port match the RSVP PATH message sender template or RSVP RESV message filter spec; this is the IP source or the RSVP sender.
- Destination IP address and port match the RSVP PATH/RESV message session object IP address; this is the IP destination address or the RSVP receiver.
- Protocol matches the RSVP PATH/RESV message session object protocol; if protocol = IP, then it matches the source or destination address as above.

Standard ACLs

The ACLs used within the **ip rsvp aggregation ip map** command match the RSVP message objects as follows for a standard ACL:

- IP address matches the RSVP PATH message sender template or RSVP RESV message filter spec; this is the IP source address or the RSVP sender.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp aggregation ip map** {**access-list** {*acl-number*} | **any**} **dscp** *value*
4. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**ip rsvp aggregation ip map** {**access-list** {*acl-number*} | **any**} **dscp** *value* **Example:** `Router(config)# ip rsvp aggregation ip map any dscp af41`	Configures RSVP aggregation rules that tell a router how to map E2E reservations onto aggregate reservations. • The keywords and arguments specify additional information such as DSCP values.
Step 4	**end** **Example:** `Router(config)# end`	(Optional) Returns to privileged EXEC mode.

Configuring Aggregate Reservation Attributes on a Deaggregator

Perform this task on a deaggregator to configure the aggregate reservation attributes (also called token bucket parameters) on a per-DSCP basis.

Note Typically, an edge router acts as both an aggregator and deaggregator because of the unidirectional nature of RSVP reservations. Most applications require bidirectional reservations. Therefore, these parameters are used by a deaggregator when mapping E2E reservations onto aggregates during the dynamic aggregate reservation process.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp aggregation ip reservation dscp** *value* [**aggregator** *agg-ip-address*] **traffic-params static rate** *data-rate* [**burst** *burst-size*] [**peak** *peak-rate*]
4. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**ip rsvp aggregation ip reservation dscp** *value* [**aggregator** *agg-ip-address*] **traffic-params static rate** *data-rate* [**burst** *burst-size*] [**peak** *peak-rate*] **Example:** `Router(config)# ip rsvp aggregation ip reservation dscp af11 aggregator 10.10.10.10 traffic-params static rate 10 burst 8 peak 10`	Configures RSVP aggregate reservation attributes (also called token bucket parameters) on a per-DSCP basis. • The keywords and arguments specify additional information.
Step 4	**end** **Example:** `Router(config)# end`	(Optional) Returns to privileged EXEC mode.

Configuring an RSVP Aggregation Router ID

Perform this task on aggregators and deaggregators to configure an RSVP aggregation router ID.

Note Both aggregators and deaggregators need to be identified with a stable and routable IP address. This is the RFC 3175 router ID, which is also the IP address of the loopback interface with the lowest number. If there is no loopback interface configured or all those configured are down, then there will be no router ID assigned for the aggregating/deaggregating function and aggregate reservations will not be established.

Note The router ID may change if the associated loopback interface goes down or its IP address is removed. In this case, the E2E and aggregate sessions are torn down. If a new router ID is determined, new E2E and aggregate sessions will use the new router ID.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface loopback** *number*
4. **ip address** *ip-address subnet-mask/prefix*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface loopback** *number* **Example:** `Router(config)# interface loopback 1`	Creates a loopback interface and enters interface configuration mode. • Enter a value for the *number* argument. The range is 0 to 2147483647.
Step 4	**ip address** *ip-address subnet-mask/prefix* **Example:** `Router(config-if)# ip address 192.168.50.1 255.255.255.0`	Configures an IP address and subnet mask or prefix on the loopback interface.
Step 5	**end** **Example:** `Router(config-if)# end`	(Optional) Returns to privileged EXEC mode.

Enabling RSVP Aggregation

Perform this task on aggregators and deaggregators to enable RSVP aggregation globally after you have completed all the previous aggregator and deaggregator configurations.

Note This task registers a router to receive RSVP-E2E-IGNORE messages. It is not necessary to perform this task on interior routers because they are only processing RSVP aggregate reservations. If you do so, you may decrease performance because the interior router will then unnecessarily process all the RSVP-E2E-IGNORE messages.

Note If you enable RSVP aggregation globally on an interior router, then you should configure all interfaces as interior.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp aggregation ip**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`ip rsvp aggregation ip` **Example:** `Router(config)# ip rsvp aggregation ip`	Enables RSVP aggregation globally on an aggregator or deaggregator.
Step 4	`end` **Example:** `Router(config)# end`	(Optional) Returns to privileged EXEC mode.

Configuring RSVP Local Policy

Perform this task to apply a local policy to an RSVP aggregate reservation.

Note In classic (unaggregated) RSVP, a session is identified in the reservation message session object by the destination IP address and protocol information. In RSVP aggregation, a session is identified by the destination IP address and DSCP within the session object of the aggregate RSVP message. The **dscp-ip** keyword matches the DSCP within the session object.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp policy local** {**acl** *acl1* [*acl2...acl8*] | **dscp-ip** *value1* [*value2 ... value8*] | **default** | **identity** *alias1* [*alias2...alias4*] | **origin-as** *as1* [*as2...as8*]}
4. {**accept** | **forward** [**all** | **path** | **path-error** | **resv** | **resv-error**] | **default** | **exit** | **fast-reroute** | **local-override** | **maximum** {**bandwidth** [**group** *x*] [**single** *y*] | **senders** *n*} | **preempt-priority** [**traffic-eng** *x*] *setup-priority* [*hold-priority*]}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp policy local** {**acl** *acl1* [*acl2...acl8*] | **dscp-ip** *value1* [*value2 ... value8*] | **default** | **identity** *alias1* [*alias2...alias4*] | **origin-as** *as1* [*as2...as8*]} **Example:** `Router(config)# ip rsvp policy local dscp-ip 46`	Creates a local policy to determine how RSVP resources are used in a network and enters local policy configuration mode. • Enter the **dscp-ip** *value* keyword and argument combination to specify a DSCP for matching the session object DCSP within the aggregate reservations. Values can be the following: – 0 to 63—Numerical. The default value is 0. – af11 to af43—Assured forwarding (AF). – cs1 to cs7—Type of service (ToS) precedence. – default—Default DSCP. – ef—Expedited Forwarding (EF). **Note** You must associate at least one DSCP with a DSCP-based policy. However, you can associate as many as eight.
Step 4	{**accept** | **forward** [**all** | **path** | **path-error** | **resv** | **resv-error**] | **default** | **exit** | **fast-reroute** | **local-override** | **maximum** {**bandwidth** [**group** *x*] [**single** *y*] | **senders** *n*} | **preempt-priority** [**traffic-eng** *x*] *setup-priority* [*hold-priority*]} **Example:** `Router(config-rsvp-policy-local)# forward all`	(Optional) Defines the properties of the dscp-ip local policy that you are creating. (These are the submode commands.) **Note** This is an optional step. An empty policy rejects everything, which may be desired in some cases. See the **ip rsvp policy local** command for more detailed information on submode commands.
Step 5	**end** **Example:** `Router(config-rsvp-policy-local)# end`	(Optional) Exits local policy configuration mode and returns to privileged EXEC mode.

Verifying the RSVP Aggregation Configuration

Perform this task to verify the RSVP aggregation configuration.

Note You can use the following **show** commands in user EXEC or privileged EXEC mode.

SUMMARY STEPS

1. **enable**
2. **show ip rsvp aggregation ip** [**endpoints** | **interface** [*if-name*] | **map** [**dscp** *value*] | **reservation** [**dscp** *value* [**aggregator** *ip-address*]]
3. **show ip rsvp aggregation ip endpoints** [**role** {**aggregator** | **deaggregator**}] [*ip-address*] [**dscp** *value*] [**detail**]
4. **show ip rsvp** [**atm-peak-rate-limit** | **counters** | **host** | **installed** | **interface** | **listeners** | **neighbor** | **policy** | **precedence** | **request** | **reservation** | **sbm** | **sender** | **signalling** | **tos**]
5. **show ip rsvp reservation** [**detail**] [**filter** [**destination** *ip-address* | *hostname*] [**dst-port** *port-number*] [**source** *ip-address* | *hostname*] [**src-port** *port-number*]]
6. **show ip rsvp sender** [**detail**] [**filter** [**destination** *ip-address* | *hostname*] [**dst-port** *port-number*] [**source** *ip-address* | *hostname*] [**src-port** *port-number*]]
7. **show ip rsvp installed** [*interface-type interface-number*] [**detail**]
8. **show ip rsvp interface** [**detail**] [*interface-type interface-number*]
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	(Optional) Enables privileged EXEC mode. • Enter your password if prompted. **Note** Skip this step if you are using the **show** commands in user EXEC mode.
Step 2	**show ip rsvp aggregation ip** [**endpoints** | **interface** [*if-name*] | **map** [**dscp** *value*] | **reservation** [**dscp** *value* [**aggregator** *ip-address*]] **Example:** `Router# show ip rsvp aggregation ip`	(Optional) Displays RSVP summary aggregation information. • The optional keywords and arguments display additional information.
Step 3	**show ip rsvp aggregation ip endpoints** [**role** {**aggregator** | **deaggregator**}] [*ip-address*] [**dscp** *value*] [**detail**] **Example:** `Router# show ip rsvp aggregation ip endpooints`	(Optional) Displays RSVP information about aggregator and deaggregator routers for currently established aggregate reservations. • The optional keywords and arguments display additional information.

	Command or Action	Purpose
Step 4	**show ip rsvp** [**atm-peak-rate-limit** | **counters** | **host** | **installed** | **interface** | **listeners** | **neighbor** | **policy** | **precedence** | **request** | **reservation** | **sbm** | **sender** | **signalling** | **tos**] **Example:** `Router# show ip rsvp`	(Optional) Displays specific information for RSVP categories. • The optional keywords display additional information.
Step 5	**show ip rsvp reservation** [**detail**] [**filter** [**destination** *ip-address* | *hostname*] [**dst-port** *port-number*] [**source** *ip-address* | *hostname*] [**src-port** *port-number*]] **Example:** `Router# show ip rsvp reservation detail`	(Optional) Displays RSVP-related receiver information currently in the database. • The optional keywords and arguments display additional information. **Note** The optional **filter** keyword is supported in Cisco IOS Releases 12.0S and 12.2S only.
Step 6	**show ip rsvp sender** [**detail**] [**filter** [**destination** *ip-address* | *hostname*] [**dst-port** *port-number*] [**source** *ip-address* | *hostname*] [**src-port** *port-number*]] **Example:** `Router# show ip rsvp sender detail`	(Optional) Displays RSVP PATH-related sender information currently in the database. • The optional keywords and arguments display additional information. **Note** The optional **filter** keyword is supported in Cisco IOS Releases 12.0S and 12.2S only.
Step 7	**show ip rsvp installed** [*interface-type interface-number*] [**detail**] **Example:** `Router# show ip rsvp installed detail`	(Optional) Displays RSVP-related installed filters and corresponding bandwidth information. • The optional keywords and arguments display additional information.
Step 8	**show ip rsvp interface** [**detail**] [*interface-type interface-number*] **Example:** `Router# show ip rsvp interface detail`	(Optional) Displays RSVP-related interface information. • The optional keywords and arguments display additional information.
Step 9	**end** **Example:** `Router# end`	(Optional) Exits privileged EXEC mode and returns to user EXEC mode.

Configuration Examples for RSVP Aggregation

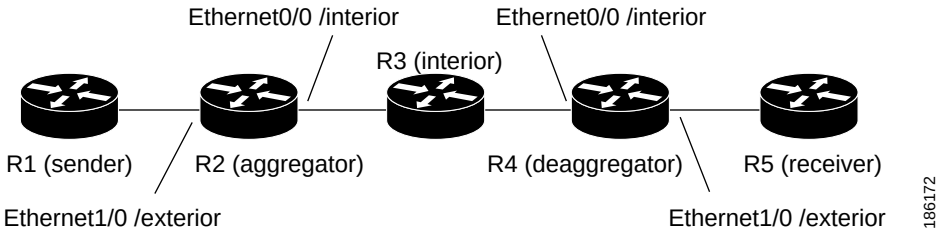
This section provides the following configuration examples for RSVP aggregation:

Configuring RSVP Aggregation: Examples

This section contains the following configuration examples:

Figure 2 shows a five-router network in which RSVP aggregation is configured.

Figure 2 Sample RSVP Aggregation Network



Configuring RSVP/ DiffServ Attributes on an Interior Router

The following example configures RSVP/DiffServ attributes on an interior router (R3 in Figure 2).

- Ethernet interface 0/0 is enabled for RSVP and the amount of bandwidth available for reservations is configured.
- A resource provider is configured and data packet classification is disabled because RSVP aggregation supports control plane aggregation only.

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface Ethernet0/0
Router(config-if)# ip rsvp bandwidth 400
Router(config-if)# ip rsvp resource-provider none
Router(config-if)# ip rsvp data-packet classification none
Router(config-if)# end
```

Configuring RSVP Aggregation on an Aggregator or Deaggregator

The following example configures RSVP aggregation attributes on an aggregator or deaggregator (R2 and R4 in Figure 2):

- Loopback 1 is configured to establish an RSVP aggregation router ID.
- Ethernet interface 0/0 is enabled for RSVP and the amount of bandwidth available for reservations is configured.
- Ethernet interface 0/0 on an aggregator or deaggregator is configured to face an aggregation region.
- A resource provider is configured and data packet classification is disabled because RSVP aggregation supports control plane aggregation only.

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface Loopback 1
Router(config)# ip address 192.168.50.1 255.255.255.0
Router(config)# interface Ethernet0/0
Router(config-if)# ip rsvp bandwidth 400
Router(config-if)# ip rsvp aggregation role interior
Router(config-if)# ip rsvp resource-provider none
Router(config-if)# ip rsvp data-packet classification none
Router(config-if)# end
```

Configuring RSVP Aggregation Attributes and Parameters

The following example configures additional RSVP aggregation attributes, including a global rule for mapping all E2E reservations onto a single aggregate with DSCP AF41 and the token bucket parameters for aggregate reservations, because dynamic resizing is not supported. This configuration is only required on nodes performing the deaggregation function (R4 in Figure 2).

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# ip rsvp aggregation ip map any dscp af41
Router(config)# ip rsvp aggregation ip reservation dscp af41 aggregator 10.10.10.10
traffic-params static rate 10 burst 8 peak 10
Router(config)# end
```

Configuring an Access List for a Deaggregator

In the following example, access list 1 is defined for all RSVP messages whose RSVP PATH message sender template source address is in the 10.1.0.0 subnet so that the deaggregator (R4 in Figure 2) maps those reservations onto an aggregate reservation for the DSCP associated with the AF41 PHB:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# access-list 1 permit 10.1.0.0 0.0.255.255
Router(config)# ip rsvp aggregation ip map access-list 1 dscp af41
Router(config)# end
```

Configuring RSVP Aggregation

After you configure your RSVP aggregation attributes, you are ready to enable aggregation globally.

When you enable aggregation on a router, the router can act as an aggregator or a deaggregator. To perform aggregator and deaggregator functions, the RSVP process must see messages with the RSVP-E2E-IGNORE protocol type (134) on a router; otherwise, the messages are forwarded as data by the router's data plane. The **ip rsvp aggregation ip** command enables RSVP to identify messages with the RSVP-E2E-IGNORE protocol.

Note This registers a router to receive RSVP-E2E-IGNORE messages. It is not necessary to configure this command on interior nodes that are only processing RSVP aggregate reservations and forwarding RSVP-E2E-IGNORE messages as IP datagrams). Since the router is loaded with an image that supports aggregation, the router will process aggregate (RFC 3175 formatted) messages correctly. Enabling aggregation on an interior mode may decrease performance because the interior node will then unnecessarily process all RSVP-E2E-IGNORE messages.

Note If you enable aggregation on an interior node, you must configure all its interfaces as interior. Otherwise, all the interfaces have the exterior role, and any E2E PATH (E2E-IGNORE) messages arriving at the router are discarded.

In summary, there are two options for an interior router (R3 in Figure 2):

- No RSVP aggregation configuration commands are entered.
- RSVP aggregation is enabled and all interfaces are configured as interior.

Configuring RSVP Local Policy

You can configure a local policy optionally on any RSVP capable node. In this example, a local policy is configured on a deaggregator to set the preemption priority values within the RSVP RESV aggregate messages based upon matching the DSCP within the aggregate RSVP messages session object. This allows the bandwidth available for RSVP reservations to be used first by reservations of DSCP EF over DSCP AF41 on interior or aggregation nodes. Any aggregate reservation for another DSCP will have a preemption priority of 0, the default.

Note Within the RSVP RESV aggregate message at the deaggregator, this local policy sets an RFC 3181 "Signaled Preemption Priority Policy Element" that can be used by interior nodes or the aggregator that has **ip rsvp preemption** enabled.

The following example sets the preemption priority locally for RSVP aggregate reservations during establishment on an interior router (R3 in Figure 2):

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# ip rsvp policy local dscp-ip ef
Router(config-rsvp-local-policy)# 5 5
Router(config-rsvp-local-policy)# exit
Router(config)# ip rsvp policy local dscp-ip af41
Router(config-rsvp-local-policy)# 2 2
Router(config-rsvp-local-policy)# end
```

Verifying the RSVP Aggregation Configuration: Example

This section contains the following verification examples:

- Verifying RSVP Aggregation and Configured Reservations, page 26
- Verifying Configured Interfaces and Their Roles, page 27
- Verifying Aggregator and Deaggregator Reservations, page 27

Verifying RSVP Aggregation and Configured Reservations

The following example verifies that RSVP aggregation is enabled and displays information about the reservations currently established and configured map and reservation policies:

```
Router# show ip rsvp aggregation ip

RFC 3175 Aggregation:  Enabled
  Level: 1
  Default QoS service: Controlled-Load

  Number of signaled aggregate reservations:  2
  Number of signaled E2E reservations:        8
  Number of configured map commands:          4
  Number of configured reservation commands:  1
```

Verifying Configured Interfaces and Their Roles

The following example displays the configured interfaces and whether they are interior or exterior in regard to the aggregation region:

```
Router# show ip rsvp aggregation ip interface

Interface Name       Role
-------------------- --------
Ethernet0/0          interior
Serial2/0            exterior
Serial3/0            exterior
```

Verifying Aggregator and Deaggregator Reservations

The following example displays information about the aggregators and deaggregators when established reservations are present:

```
Router# show ip rsvp aggregation ip endpoints detail

Role  DSCP Aggregator      Deaggregator    State  Rate    Used    QBM PoolID
----- ---- --------------- --------------- ------ ------- ------- ----------
Agg   46   10.3.3.3         10.4.4.4         ESTABL 100K    100K    0x00000003
   Aggregate Reservation for the following E2E Flows (PSBs):
To             From          Pro DPort Sport  Prev Hop       I/F      BPS
10.4.4.4       10.1.1.1       UDP 1     1     10.23.20.3     Et1/0    100K

   Aggregate Reservation for the following E2E Flows (RSBs):
To             From          Pro DPort Sport  Next Hop       I/F      Fi Serv BPS
10.4.4.4       10.1.1.1       UDP 1     1     10.4.4.4       Se2/0    FF RATE 100K

   Aggregate Reservation for the following E2E Flows (Reqs):
To             From          Pro DPort Sport  Next Hop       I/F      Fi Serv BPS
10.4.4.4       10.1.1.1       UDP 1     1     10.23.20.3     Et1/0    FF RATE 100K
```

Additional References

The following sections provide references related to the RSVP Aggregation feature.

Related Documents

Related Topic	Document Title
RSVP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
QoS features including signaling, classification, and congestion management	"Quality of Service Overview" module
Information on RSVP local policies	"RSVP Local Policy Support" module
Information on RSVP scalability enhancements	"RSVP Scalability Enhancements" module

Standards

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

MIBs

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

RFCs

RFC	Title
RFC 2205	*Resource ReSerVation Protocol (RSVP)—Version 1 Functional Specification*
RFC 2209	*Resource ReSerVation Protocol (RSVP)—Version 1 Message Processing Rules*
RFC 3175	*Aggregation of RSVP for IPv4 and IPv6 Reservations*
RFC 3181	*Signaled Preemption Priority Policy Element*
RFC 4804	*Aggregation of Resource ReSerVation Protocol (RSVP) Reservations over MPLS TE/DS-TE Tunnels*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Quality of Service Solutions Command Reference* at http://www.cisco.com/en/US/docs/ios/qos/command/reference/qos_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

- **debug ip rsvp aggregation**
- **debug qbm**
- **ip rsvp aggregation ip**
- **ip rsvp aggregation ip map**
- **ip rsvp aggregation ip reservation dscp traffic-params static rate**
- **ip rsvp aggregation ip role interior**
- **ip rsvp policy local**
- **show ip rsvp**
- **show ip rsvp aggregation ip**
- **show ip rsvp aggregation ip endpoints**
- **show ip rsvp installed**
- **show ip rsvp interface**
- **show ip rsvp policy local**
- **show ip rsvp request**
- **show ip rsvp reservation**
- **show ip rsvp sender**
- **show qbm client**
- **show qbm pool**

Feature Information for RSVP Aggregation

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for RSVP Aggregation

Feature Name	Releases	Feature Information
RSVP Aggregation	12.2(33)SRC	The RSVP Aggregation feature allows the Resource Reservation Protocol (RSVP) state to be reduced within an RSVP/DiffServ network by aggregating many smaller reservations into a single, larger reservation at the edge.

Glossary

admission control—The process by which an RSVP reservation is accepted or rejected on the basis of end-to-end available network resources.

aggregate—An RSVP flow that represents multiple end-to-end (E2E) flows; for example, a Multiprotocol Label Switching Traffic Engineering (MPLS-TE) tunnel may be an aggregate for many E2E flows.

aggregation region—An area where E2E flows are represented by aggregate flows, with aggregators and deaggregators at the edge; for example, an MPLS-TE core, where TE tunnels are aggregates for E2E flows. An aggregation region contains a connected set of nodes that are capable of performing RSVP aggregation.

aggregator—The router that processes the E2E PATH message as it enters the aggregation region. This router is also called the TE tunnel head-end router; it forwards the message from an exterior interface to an interior interface.

bandwidth—The difference between the highest and lowest frequencies available for network signals. The term is also used to describe the rated throughput capacity of a given network medium or protocol.

deaggregator—The router that processes the E2E PATH message as it leaves the aggregation region. This router is also called the TE tunnel tail-end router; it forwards the message from an interior interface to an exterior interface.

E2E—end-to-end. An RSVP flow that crosses an aggregation region, and whose state is represented in aggregate within this region, such as a classic RSVP unicast flow crossing an MPLS-TE core.

LSP—label-switched path. A configured connection between two routers, in which label switching is used to carry the packets. The purpose of an LSP is to carry data packets.

QoS—quality of service. A measure of performance for a transmission system that reflects its transmission quality and service availability.

RSVP—Resource Reservation Protocol. A protocol that supports the reservation of resources across an IP network. Applications running on IP end systems can use RSVP to indicate to other nodes the nature (bandwidth, jitter, maximum burst, and so on) of the packet streams that they want to receive.

state—Information that a router must maintain about each LSP. The information is used for rerouting tunnels.

TE—traffic engineering. The techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods had been used.

tunnel—Secure communications path between two peers, such as two routers.

MPLS TE—Tunnel-Based Admission Control (TBAC)

First Published: January 7, 2008
Last Updated: January 7, 2008

The MPLS TE—Tunnel-Based Admission Control (TBAC) feature enables classic Resource Reservation Protocol (RSVP) unicast reservations that are traveling across a Multiprotocol Label Switching Traffic Engineering (MPLS TE) core to be aggregated over an MPLS TE tunnel.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your Cisco IOS software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for MPLS TE—Tunnel-Based Admission Control (TBAC)" section on page 19.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for MPLS TE—Tunnel-Based Admission Control (TBAC)

- You must configure an MPLS TE tunnel in the network.
- You must configure RSVP on one or more interfaces on at least two neighboring routers that share a link within the network.

Restrictions for MPLS TE—Tunnel-Based Admission Control (TBAC)

- Only IPv4 unicast RSVP flows are supported.
- Primary, one-hop tunnels are not supported. The TE tunnel cannot be a member of a class-based tunnel selection (CBTS) bundle.
- Multi-Topology Routing (MTR) is not supported.
- Only preestablished aggregates are supported. They can be configured statically or dynamically using command-line interface (CLI) commands.
- This feature is supported on Cisco 7600 series routers only.

Information About MPLS TE—Tunnel-Based Admission Control (TBAC)

To use the MPLS TE—Tunnel-Based Admission Control (TBAC) feature, you should understand the following concepts:

Feature Overview of MPLS TE—Tunnel-Based Admission Control (TBAC)

TBAC aggregates reservations from multiple, classic RSVP sessions over different forms of tunneling technologies that include MPLS TE tunnels, which act as aggregate reservations in the core. Figure 1 gives an overview of TBAC.

Figure 1 TBAC Overview

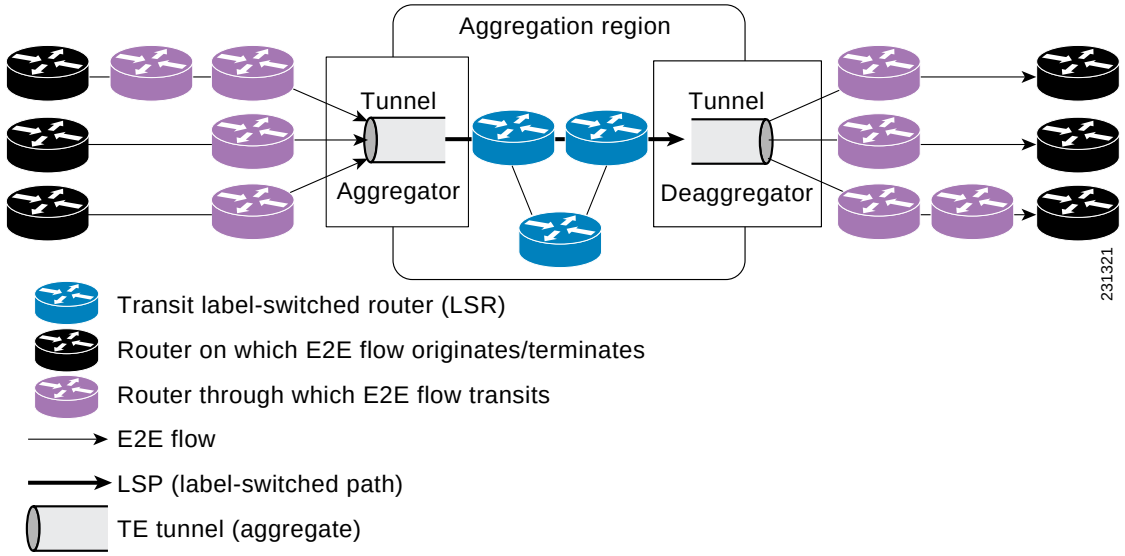


Figure 1 shows three RSVP end-to-end (E2E) flows that originate at routers on the far left, and terminate on routers at the far right. These flows are classic RSVP unicast flows, meaning that RSVP is maintaining a state for each flow. There is nothing special about these flows, except that along their path, these flows encounter an MPLS-TE core, where there is a desire to avoid creating a per-flow RSVP state.

When the E2E flows reach the edge of the MPLS-TE core, they are aggregated onto a TE tunnel. This means that when transiting through the MPLS-TE core, their state is represented by a single state; the TE tunnel is within the aggregation region, and their packets are forwarded (label-switched) by the TE tunnel. For example, if 100 E2E flows traverse the same aggregator and deaggregator, rather than creating 100 RSVP states (PATH and RESV messages) within the aggregation region, a single RSVP-TE state is created, that of the aggregate, which is the TE tunnel, to allocate and maintain the resources used by the 100 E2E flows. In particular, the bandwidth consumed by E2E flows within the core is allocated and maintained in aggregate by the TE tunnel. The bandwidth of each E2E flow is normally admitted into the TE tunnel at the headend, just as any E2E flow's bandwidth is admitted onto an outbound link in the absence of aggregation.

Benefits of MPLS TE—Tunnel-Based Admission Control (TBAC)

To understand the benefits of TBAC, you should be familiar with how Call Admission Control (CAC) works for RSVP and QoS.

Cost Effective

Real-time traffic is very sensitive to loss and delay. CAC avoids QoS degradation for real-time traffic because CAC ensures that the accepted load always matches the current network capacity. As a result, you do not have to overprovision the network to compensate for absolute worst peak traffic or for reduced capacity in case of failure.

Improved Accuracy

CAC uses RSVP signaling, which follows the exact same path as the real-time flow, and routers make a CAC decision at every hop. This ensures that the CAC decision is very accurate and dynamically adjusts to the current conditions such as a reroute or an additional link. Also, RSVP provides an explicit CAC response (admitted or rejected) to the application, so that the application can react appropriately and fast; for example, sending a busy signal for a voice call, rerouting the voice call on an alternate VoIP route, or displaying a message for video on demand.

RSVP and MPLS TE Combined

TBAC allows you to combine the benefits of RSVP with those of MPLS TE. Specifically, you can use MPLS TE inside the network to ensure that the transported traffic can take advantage of Fast Reroute protection (50-millisecond restoration), Constraint Based Routing (CBR), and aggregate bandwidth reservation.

Seamless Deployment

TBAC allows you to deploy IPv4 RSVP without any impact on the MPLS part of the network because IPv4 RSVP is effectively tunneled inside MPLS TE tunnels that operate unchanged as per regular RSVP TE. No upgrade or additional protocol is needed in the MPLS core.

Enhanced Scaling Capability

TBAC aggregates multiple IPv4 RSVP reservations ingressing from the same MPLS TE headend router into a single MPLS TE tunnel and egressing from the same MPLS TE tailend router.

How to Configure MPLS TE—Tunnel-Based Admission Control (TBAC)

This section contains the following procedures:

- Enabling RSVP QoS, page 4 (required)
- Enabling MPLS TE, page 5 (required)
- Configuring an MPLS TE Tunnel Interface, page 6 (required)
- Configuring RSVP Bandwidth on an MPLS TE Tunnel Interface, page 7 (required)
- Verifying the TBAC Configuration, page 8 (optional)

Enabling RSVP QoS

Perform this task to enable RSVP QoS globally on a router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp qos**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`ip rsvp qos` **Example:** `Router(config)# ip rsvp qos`	Enables RSVP QoS globally on a router.
Step 4	`end` **Example:** `Router(config)# end`	(Optional) Returns to privileged EXEC mode.

Enabling MPLS TE

Perform this task to enable MPLS TE globally on a router that is running RSVP QoS.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls traffic-eng tunnels**
4. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**mpls traffic-eng tunnels** **Example:** `Router(config)# mpls traffic-eng tunnels`	Enables MPLS TE globally on a router.
Step 4	**end** **Example:** `Router(config)# end`	(Optional) Returns to privileged EXEC mode.

Configuring an MPLS TE Tunnel Interface

Perform this task to configure MPLS-TE tunneling on an interface.

Prerequisites

You must configure an MPLS-TE tunnel in your network before you can proceed. For detailed information, see the "MPLS Traffic Engineering (TE)—Automatic Bandwidth Adjustment for TE Tunnels" module.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**interface tunnel** ***number*** **Example:** `Router(config)# interface tunnel1`	Specifies a tunnel interface and enters interface configuration mode.
Step 4	**end** **Example:** `Router(config-if)# end`	(Optional) Returns to privileged EXEC mode.

Configuring RSVP Bandwidth on an MPLS TE Tunnel Interface

Perform this task to configure RSVP bandwidth on the MPLS TE tunnel interface that you are using for the aggregation.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** ***number*** **Example:** `Router(config)# interface tunnel1`	Specifies a tunnel interface and enters interface configuration mode.

	Command or Action	Purpose
Step 4	`ip rsvp bandwidth [interface-kbps] [single-flow-kbps]` **Example:** `Router(config-if)# ip rsvp bandwidth 7500`	Enables RSVP bandwidth on an interface. • The optional *interface-kbps* and *single-flow-kbps* arguments specify the amount of bandwidth that can be allocated by RSVP flows or to a single flow, respectively. Values are from 1 to 10000000. **Note** You must enter a value for the *interface-kbps* argument on a tunnel interface.
Step 5	**end** **Example:** `Router(config-if)# end`	(Optional) Returns to privileged EXEC mode.

Verifying the TBAC Configuration

Perform this task to verify the TBAC configuration.

Note You can use the following **show** commands in user EXEC or privileged EXEC mode.

SUMMARY STEPS

1. **enable**
2. **show ip rsvp** [**atm-peak-rate-limit** | **counters** | **host** | **installed** | **interface** | **listeners** | **neighbor** | **policy** | **precedence** | **request** | **reservation** | **sbm** | **sender** | **signalling** | **tos**]
3. **show ip rsvp reservation** [**detail**] [**filter** [**destination** *ip-address* | *hostname*] [**dst-port** *port-number*] [**source** *ip-address* | *hostname*] [**src-port** *port-number*]]
4. **show ip rsvp sender** [**detail**] [**filter** [**destination** *ip-address* | *hostname*] [**dst-port** *port-number*] [**source** *ip-address* | *hostname*] [**src-port** *port-number*]]
5. **show mpls traffic-eng link-management bandwidth-allocation** [*interface-name* | **summary** [*interface-name*]]
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	(Optional) Enables privileged EXEC mode. • Enter your password if prompted. **Note** Skip this step if you are using the **show** commands in user EXEC mode.
Step 2	**show ip rsvp** [**atm-peak-rate-limit** | **counters** | **host** | **installed** | **interface** | **listeners** | **neighbor** | **policy** | **precedence** | **request** | **reservation** | **sbm** | **sender** | **signalling** | **tos**] **Example:** `Router# show ip rsvp`	Displays specific information for RSVP categories. • The optional keywords display additional information.
Step 3	**show ip rsvp reservation** [**detail**] [**filter** [**destination** *ip-address* | *hostname*] [**dst-port** *port-number*] [**source** *ip-address* | *hostname*] [**src-port** *port-number*]] **Example:** `Router# show ip rsvp reservation detail`	Displays RSVP-related receiver information currently in the database. • The optional keywords display additional information. **Note** The optional **filter** keyword is supported in Cisco IOS Releases 12.0S and 12.2S only.
Step 4	**show ip rsvp sender** [**detail**] [**filter** [**destination** *ip-address* | *hostname*] [**dst-port** *port-number*] [**source** *ip-address* | *hostname*] [**src-port** *port-number*]] **Example:** `Router# show ip rsvp sender detail`	Displays RSVP PATH-related sender information currently in the database. • The optional keywords display additional information. **Note** The optional **filter** keyword is supported in Cisco IOS Releases 12.0S and 12.2S only.
Step 5	**show mpls traffic-eng link-management bandwidth-allocation** [*interface-name* | **summary** [*interface-name*]] **Example:** `Router# show mpls traffic-eng link-management bandwidth-allocation`	Displays current local link information. • The optional keywords display additional information.
Step 6	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode and returns to user EXEC mode.

Configuration Examples for MPLS TE—Tunnel-Based Admission Control (TBAC)

This section provides the following configuration examples for TBAC:

Configuring TBAC: Example

Note You must have an MPLS TE tunnel already configured in your network. For detailed information, see the "MPLS Traffic Engineering (TE)—Automatic Bandwidth Adjustment for TE Tunnels" module.

The following example enables RSVP and MPLS TE globally on a router and then configures a tunnel interface and bandwidth of 7500 kbps on the tunnel interface traversed by the RSVP flows:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip rsvp qos
Router(config)# mpls traffic-eng tunnels
Router(config)# interface tunnel1
Router(config-if)# ip rsvp bandwidth 7500
Router(config-if)# end
```

Configuring RSVP Local Policy on a Tunnel Interface: Example

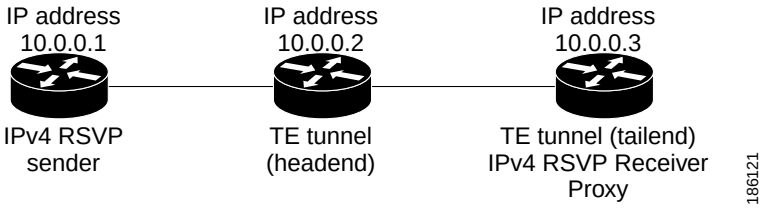
The following example configures an RSVP default local policy on a tunnel interface:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface tunnel1
Router(config-if)# ip rsvp policy local default
Router(config-rsvp-local-if-policy)# max bandwidth single 10
Router(config-rsvp-local-if-policy)# forward all
Router(config-rsvp-local-if-policy)# end
```

Verifying the TBAC Configuration: Example

Figure 2 shows a network in which TBAC is configured.

Figure 2 Sample TBAC Network



The following example verifies that RSVP and MPLS TE are enabled and coexist on the headend router (10.0.0.2 in Figure 2):

```
Router# show ip rsvp

RSVP: enabled (on 3 interface(s))
  RSVP QoS enabled <-------
  MPLS/TE signalling enabled <------

Signalling:
   Refresh interval (msec): 30000
   Refresh misses: 4
...
```

The following example verifies that RSVP and MPLS TE are enabled and coexist on the tailend router (10.0.0.3 in Figure 2):

```
Router# show ip rsvp

RSVP: enabled (on 3 interface(s))
  RSVP QoS enabled <-------
  MPLS/TE signalling enabled <------

Signalling:
   Refresh interval (msec): 30000
   Refresh misses: 4
...
```

The following examples verify that an IPv4 flow is traveling through a TE tunnel (a label-switched path [LSP]) on the headend router (10.0.0.2 in Figure 2):

```
Router# show ip rsvp sender

To              From          Pro DPort Sport Prev Hop       I/F      BPS
10.0.0.3        10.0.0.1      UDP 2     2     10.0.0.1       Et0/0    10K <-- IPv4 flow
10.0.0.3        10.0.0.2      0   1     11    none           none     100K <-- TE tunnel
```

```
Router# show ip rsvp reservation

To            From        Pro DPort Sport Next Hop     I/F    Fi Serv BPS
10.0.0.3      10.0.0.1    UDP 2     2     10.0.0.3     Tu1    SE RATE 10K <-- IPv4 flow
10.0.0.3      10.0.0.2    0   1     11    10.1.0.2     Et1/0  SE LOAD 100K <-- TE tunnel
```

The following examples verify that an IPv4 flow is traveling through a TE tunnel (LSP) on the tailend router (10.0.0.3 in Figure 2):

```
Router# show ip rsvp sender

To              From          Pro DPort Sport Prev Hop       I/F      BPS
10.0.0.3        10.0.0.1      UDP 2     2     10.0.0.2       Et1/0    10K <-- IPv4 flow
10.0.0.3        10.0.0.2      0   1     11    10.1.0.1       Et1/0    100K <-- TE tunnel
```

```
Router# show ip rsvp reservation

To            From          Pro DPort Sport Next Hop   I/F    Fi Serv BPS
10.0.0.3      10.0.0.1      UDP 2     2     none       none   SE RATE 10K <-- IPv4 flow
10.0.0.3      10.0.0.2      0   1     11    none       none   SE LOAD 100K <-- TE tunnel
```

The following examples display detailed information about the IPv4 flow and the TE tunnel (LSP) on the headend router (10.0.0.2 in Figure 2):

```
Router# show ip rsvp sender detail

PATH: <---------------------------------------------- IPv4 flow information begins here.
  Destination 10.0.0.3, Protocol_Id 17, Don't Police , DstPort 2
  Sender address: 10.0.0.1, port: 2
  Path refreshes:
    arriving: from PHOP 10.0.0.10 on Et0/0 every 30000 msecs. Timeout in 189 sec
  Traffic params - Rate: 10K bits/sec, Max. burst: 10K bytes
    Min Policed Unit: 0 bytes, Max Pkt Size 2147483647 bytes
  Path ID handle: 02000412.
  Incoming policy: Accepted. Policy source(s): Default
  Status:
  Output on Tunnel1, out of band. Policy status: Forwarding. Handle: 0800040E <--- TE
tunnel verified
    Policy source(s): Default
  Path FLR: Never repaired

PATH: <---------------------------------------------- TE tunnel information begins here.
  Tun Dest:   10.0.0.3  Tun ID: 1  Ext Tun ID: 10.0.0.2
  Tun Sender: 10.0.0.2  LSP ID: 11
  Path refreshes:
    sent:     to   NHOP 10.1.0.2 on Ethernet1/0
  ...


Router# show ip rsvp reservation detail

RSVP Reservation. Destination is 10.0.0.3, Source is 10.0.0.1,<--- IPv4 flow information
begins here.
  Protocol is UDP, Destination port is 2, Source port is 2
  Next Hop: 10.0.0.3 on Tunnel1, out of band <-------------------- TE tunnel verified
  Reservation Style is Shared-Explicit, QoS Service is Guaranteed-Rate
  ...

Reservation: <-------------------------------------- TE Tunnel information begins here.
  Tun Dest:   10.0.0.3  Tun ID: 1  Ext Tun ID: 10.0.0.2
  Tun Sender: 10.0.0.2  LSP ID: 11
  Next Hop: 10.1.0.2 on Ethernet1/0
  Label: 0 (outgoing)
  Reservation Style is Shared-Explicit, QoS Service is Controlled-Load
  ...


Router# show ip rsvp installed detail

RSVP: Ethernet0/0 has no installed reservations

RSVP: Ethernet1/0 has the following installed reservations
RSVP Reservation. Destination is 10.0.0.3. Source is 10.0.0.2,
  Protocol is 0  , Destination port is 1, Source port is 11
  Traffic Control ID handle: 03000405
  Created: 04:46:55 EST Fri Oct 26 2007 <------ IPv4 flow information
  Admitted flowspec:
```

```
    Reserved bandwidth: 100K bits/sec, Maximum burst: 1K bytes, Peak rate: 100K bits/sec
    Min Policed Unit: 0 bytes, Max Pkt Size: 1500 bytes
  Resource provider for this flow: None
  ...

RSVP: Tunnel1 has the following installed reservations <------ TE tunnel verified
RSVP Reservation. Destination is 10.0.0.3. Source is 10.0.0.1,
  Protocol is UDP, Destination port is 2, Source port is 2
  Traffic Control ID handle: 01000415
  Created: 04:57:07 EST Fri Oct 26 2007 <----- IPv4 flow information
  Admitted flowspec:
    Reserved bandwidth: 10K bits/sec, Maximum burst: 10K bytes, Peak rate: 10K bits/sec
    Min Policed Unit: 0 bytes, Max Pkt Size: 0 bytes
  Resource provider for this flow: None
  ...


Router# show ip rsvp interface detail

 Et0/0:
   RSVP: Enabled
   Interface State: Up
   Bandwidth:
     Curr allocated: 0 bits/sec
     Max. allowed (total): 3M bits/sec
     Max. allowed (per flow): 3M bits/sec
     ...

 Et1/0:
   RSVP: Enabled
   Interface State: Up
   Bandwidth:
     Curr allocated: 0 bits/sec
     Max. allowed (total): 3M bits/sec
     Max. allowed (per flow): 3M bits/sec
    ...

Tu1: <-------------------------------- TE tunnel information begins here.
   RSVP: Enabled
   RSVP aggregation over MPLS TE: Enabled
   Interface State: Up
   Bandwidth:
     Curr allocated: 20K bits/sec
     Max. allowed (total): 3M bits/sec
     Max. allowed (per flow): 3M bits/sec
     ...
```

The following examples display detailed information about the IPv4 flow and the TE tunnel (LSP) on the tailend router (10.0.0.3 in Figure 2):

```
Router# show ip rsvp sender detail

PATH: <---------------------------------------------- IPv4 flow information begins here.
  Destination 10.0.0.3, Protocol_Id 17, Don't Police , DstPort 2
  Sender address: 10.0.0.1, port: 2
  Path refreshes:
    arriving: from PHOP 10.0.0.2 on Et1/0 every 30000 msecs, out of band. Timeout in 188
sec
  Traffic params - Rate: 10K bits/sec, Max. burst: 10K bytes
    Min Policed Unit: 0 bytes, Max Pkt Size 2147483647 bytes
  ...
```

```
PATH: <---------------------------------------------- TE tunnel information begins here.
  Tun Dest:   10.0.0.3  Tun ID: 1  Ext Tun ID: 10.0.0.2
  Tun Sender: 10.0.0.2  LSP ID: 11
  Path refreshes:
    arriving: from PHOP 10.1.0.1 on Et1/0 every 30000 msecs. Timeout in 202 sec
  ...


Router# show ip rsvp reservation detail

RSVP Reservation. Destination is 10.0.0.3, Source is 10.0.0.1, <--- IPv4 flow information
begins here.
  Protocol is UDP, Destination port is 2, Source port is 2
  Next Hop: none
  Reservation Style is Shared-Explicit, QoS Service is Guaranteed-Rate
  ...

Reservation: <--------------------------------------- TE tunnel information begins here.
  Tun Dest:   10.0.0.3  Tun ID: 1  Ext Tun ID: 10.0.0.2
  Tun Sender: 10.0.0.2  LSP ID: 11
  Next Hop: none
  Label: 1 (outgoing)
  Reservation Style is Shared-Explicit, QoS Service is Controlled-Load
  ...


Router# show ip rsvp request detail

RSVP Reservation. Destination is 10.0.0.3, Source is 10.0.0.1,
  Protocol is UDP, Destination port is 2, Source port is 2
  Prev Hop: 10.0.0.2 on Ethernet1/0, out of band <-------------- TE tunnel verified
  Reservation Style is Shared-Explicit, QoS Service is Guaranteed-Rate
  Average Bitrate is 10K bits/sec, Maximum Burst is 10K bytes
  ...

Request: <----------------------------------- TE tunnel information begins here.
  Tun Dest:   10.0.0.3  Tun ID: 1  Ext Tun ID: 10.0.0.2
  Tun Sender: 10.0.0.2  LSP ID: 11
  Prev Hop: 10.1.0.1 on Ethernet1/0
  Label: 0 (incoming)
  Reservation Style is Shared-Explicit, QoS Service is Controlled-Load
  ...
```

Verifying the RSVP Local Policy Configuration: Example

The following example verifies that a default local policy has been configured on tunnel interface 1:

```
Router# show run interface tunnnel 1

Building configuration...

Current configuration : 419 bytes
!
interface Tunnel1
 bandwidth 3000
 ip unnumbered Loopback0
 tunnel destination 10.0.0.3
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 1 1
 tunnel mpls traffic-eng bandwidth 100
 tunnel mpls traffic-eng path-option 1 dynamic
```

```
 tunnel mpls traffic-eng fast-reroute
 ip rsvp policy local default <---------------- Local policy information begins here.
  max bandwidth single 10
  forward all
 ip rsvp bandwidth 3000
end
```

The following example provides additional information about the default local policy configured on tunnel interface 1:

```
Router# show ip rsvp policy local detail

Tunnel1:
  Default policy:

    Preemption Scope: Unrestricted.
    Local Override:   Disabled.
    Fast ReRoute:     Accept.
    Handle:           BC000413.

                          Accept              Forward
    Path:                 Yes                 Yes
    Resv:                 Yes                 Yes
    PathError:            Yes                 Yes
    ResvError:            Yes                 Yes

                          Setup Priority      Hold Priority
    TE:                   N/A                 N/A
    Non-TE:               N/A                 N/A

                          Current             Limit
    Senders:              0                   N/A
    Receivers:            1                   N/A
    Conversations:        1                   N/A
    Group bandwidth (bps): 10K                N/A
    Per-flow b/w (bps):    N/A                10K


Generic policy settings:
    Default policy: Accept all
    Preemption:     Disabled
```

Additional References

The following sections provide references related to the MPLS TE Tunnel-Based Admission Control (TBAC) feature.

Related Documents

Related Topic	Document Title
RSVP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
QoS features including signaling, classification, and congestion management	"Quality of Service Overview" module

Standards

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

MIBs

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

RFCs

RFC	Title
RFC 2205	*Resource ReSerVation Protocol (RSVP)—Version 1 Functional Specification*
RFC 2209	*Resource ReSerVation Protocol (RSVP)—Version 1 Message Processing Rules*
RFC 3175	*Aggregation of RSVP for IPv4 and IPv6 Reservations*
RFC 3209	*RSVP-TE: Extensions to RSVP for LSP Tunnels*
RFC 4804	*Aggregation of Resource ReSerVation Protocol (RSVP) Reservations over MPLS TE/DS-TE Tunnels*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Quality of Service Solutions Command Reference* at http://www.cisco.com/en/US/docs/ios/qos/command/reference/qos_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

- **ip rsvp qos**
- **show ip rsvp**
- **show ip rsvp reservation**
- **show ip rsvp sender**
- **show mpls traffic-eng link-management bandwidth-allocation**

Feature Information for MPLS TE—Tunnel-Based Admission Control (TBAC)

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for MPLS TE—Tunnel-Based Admission Control (TBAC)

Feature Name	Releases	Feature Information
MPLS TE Tunnel-Based Admission Control (TBAC)	12.2(33)SRC	The MPLS TE—Tunnel-Based Admission Control (TBAC) feature enables classic Resource Reservation Protocol (RSVP) unicast reservations that are traveling across a Multiprotocol Label Switching Traffic Engineering (MPLS TE) core to be aggregated over an MPLS TE tunnel.

Glossary

admission control—The process by which an RSVP reservation is accepted or rejected on the basis of end-to-end available network resources.

aggregate—An RSVP flow that represents multiple E2E flows; for example, an MPLS-TE tunnel may be an aggregate for many E2E flows.

aggregation region—A area where E2E flows are represented by aggregate flows, with aggregators and deaggregators at the edge; for example, an MPLS-TE core, where TE tunnels are aggregates for E2E flows. An aggregation region contains a connected set of nodes that are capable of performing RSVP aggregation.

aggregator—The router that processes the E2E PATH message as it enters the aggregation region. This router is also called the TE tunnel headend router; it forwards the message from an exterior interface to an interior interface.

bandwidth—The difference between the highest and lowest frequencies available for network signals. The term is also used to describe the rated throughput capacity of a given network medium or protocol.

deaggregator—The router that processes the E2E PATH message as it leaves the aggregation region. This router is also called the TE tunnel tailend router; it forwards the message from an interior interface to an exterior interface.

E2E—end-to-end. An RSVP flow that crosses an aggregation region and whose state is represented in aggregate within this region; for example, a classic RSVP unicast flow that crosses an MPLS-TE core.

LSP—label-switched path. A configured connection between two routers, in which label switching is used to carry the packets. The purpose of an LSP is to carry data packets.

MPLS—Multiprotocol Label Switching. Packet-forwarding technology, used in the network core, that applies data link layer labels to tell switching nodes how to forward data, resulting in faster and more scalable forwarding than network layer routing normally can do.

QoS—quality of service. A measure of performance for a transmission system that reflects its transmission quality and service availability.

RSVP—Resource Reservation Protocol. A protocol that supports the reservation of resources across an IP network. Applications that run on IP end systems can use RSVP to indicate to other nodes the nature (bandwidth, jitter, maximum burst, and so on) of the packet streams that they want to receive.

state—Information that a router must maintain about each LSP. The information is used for rerouting tunnels.

TE—traffic engineering. The techniques and processes that are used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods had been used.

tunnel—Secure communications path between two peers, such as two routers.

Configuring Subnetwork Bandwidth Manager

This chapter describes the tasks for configuring the Subnetwork Bandwidth Manager (SBM) feature, which is a signalling feature that enables Resource Reservation Protocol (RSVP)-based admission control over IEEE 802-styled networks.

For complete conceptual information, see "Signalling Overview" module.

For a complete description of the SBM commands in this chapter, see the *Cisco IOS Quality of Service Solutions Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Subnetwork Bandwidth Manager Configuration Task List

To configure SBM, perform the tasks described in the following sections. The task in the first section is required; the tasks in the remaining sections are optional.

- Configuring an Interface as a Designated SBM Candidate (Required)
- Configuring the NonResvSendLimit Object (Optional)
- Verifying Configuration of SBM State (Optional)

See the end of this chapter for the section "Subnetwork Bandwidth Manager Candidate Configuration Example."

Configuring an Interface as a Designated SBM Candidate

SBM is used in conjunction with RSVP. Therefore, before you configure an interface as a Designated SBM (DSBM) contender, ensure that RSVP is enabled on that interface.

To configure the interface as a DSBM candidate, use the following command in interface configuration mode:

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Command	Purpose
`Router(config-if)# ip rsvp dsbm candidate [priority]`	Configures the interface to participate as a contender in the DSBM dynamic election process, whose winner is based on the highest priority.

Configuring the NonResvSendLimit Object

The NonResvSendLimit object specifies how much traffic can be sent onto a managed segment without a valid RSVP reservation.

To configure the NonResvSendLimit object parameters, use the following commands in interface configuration mode, as needed:

Command	Purpose
`Router(config-if)# ip rsvp dsbm non-resv-send-limit rate kBps`	Configures the average rate, in kbps, for the DSBM candidate.
`Router(config-if)# ip rsvp dsbm non-resv-send-limit burst kilobytes`	Configures the maximum burst size, in KB, for the DSBM candidate.
`Router(config-if)# ip rsvp dsbm non-resv-send-limit peak kBps`	Configures the peak rate, in kbps, for the DSBM candidate.
`Router(config-if)# ip rsvp dsbm non-resv-send-limit min-unit bytes`	Configures the minimum policed unit, in bytes, for the DSBM candidate.
`Router(config-if)# ip rsvp dsbm non-resv-send-limit max-unit bytes`	Configures the maximum packet size, in bytes, for the DSBM candidate.

To configure the per-flow limit on the amount of traffic that can be sent without a valid RSVP reservation, configure the **rate**, **burst**, **peak**, **min-unit**, and **max-unit** keywords for finite values from 0 to infinity.

To allow all traffic to be sent without a valid RSVP reservation, configure the **rate**, **burst**, **peak**, **min-unit**, and **max-unit** keywords for unlimited. To configure the parameters for unlimited, you can either omit the command or enter the **no** version of the command (for example, **no ip rsvp dsbm non-resv-send-limit rate**). Unlimited is the default value.

The absence of the NonResvSendLimit object allows any amount of traffic to be sent without a valid RSVP reservation.

Verifying Configuration of SBM State

To display information that enables you to determine if an interface has been configured as a DSBM candidate and which of the contenders has been elected the DSBM, use the following command in EXEC mode:

Command	Purpose
`Router# show ip rsvp sbm [detail] [interface]`	Displays information about an SBM configured for a specific RSVP-enabled interface or forall RSVP-enabled interfaces on the router. Using the **detail** keyword allows you to view the values for the NonResvSendLimit object.

The displayed output from the **show ip rsvp sbm** command identifies the interface by name and IP address, and it shows whether the interface has been configured as a DSBM contender. If the interface is a contender, the DSBM Priority field displays its priority. The DSBM election process is dynamic, addressing any new contenders configured as participants. Consequently, at any given time, an incumbent DSBM might be replaced by one configured with a higher priority. The following example shows sample output from the **show ip rsvp sbm** command:

```
Router# show ip rsvp sbm

Interface DSBM Addr      DSBM Priority    DSBM Candidate   My Priority
Et1       1.1.1.1        70               yes              70
Et2       145.2.2.150    100              yes              100
```

If you use the **detail** keyword, the output is shown in a different format. In the left column, the local DSBM candidate configuration is shown; in the right column, the corresponding information for the current DSBM is shown. In the following example, the local DSBM candidate won election and is the current DSBM:

```
Router# show ip rsvp sbm detail

Interface:Ethernet2
Local Configuration             Current DSBM
  IP Address:10.2.2.150         IP Address:10.2.2.150
  DSBM candidate:yes             I Am DSBM:yes
  Priority:100                   Priority:100
  Non Resv Send Limit             Non Resv Send Limit
    Rate:500 Kbytes/sec            Rate:500 Kbytes/sec
    Burst:1000 Kbytes              Burst:1000 Kbytes
    Peak:500 Kbytes/sec            Peak:500 Kbytes/sec
    Min Unit:unlimited             Min Unit:unlimited
    Max Unit:unlimited             Max Unit:unlimited
```

Subnetwork Bandwidth Manager Candidate Configuration Example

For information about configuring SBM, see the section "Subnetwork Bandwidth Manager Configuration Task List" in this chapter.

In the following example, RSVP and SBM are enabled on Ethernet interface 2. After RSVP is enabled, the interface is configured as a DSBM and SBM candidate with a priority of 100. The configured priority is high, making this interface a good contender for DSBM status. However, the maximum configurable priority value is 128, so another interface configured with a higher priority could win the election and become the DSBM.

```
interface Ethernet2
 ip address 145.2.2.150 255.255.255.0
 no ip directed-broadcast
```

```
 ip pim sparse-dense-mode
 no ip mroute-cache
 media-type 10BaseT
 ip rsvp bandwidth 7500 7500
 ip rsvp dsbm candidate 100
 ip rsvp dsbm non-resv-send-limit rate 500
 ip rsvp dsbm non-resv-send-limit burst 1000
 ip rsvp dsbm non-resv-send-limit peak 500
end
```

Link Efficiency Mechanisms

Link Efficiency Mechanisms Overview

Cisco IOS software offers a number of link-layer efficiency mechanisms or features (listed below) designed to reduce latency and jitter for network traffic. These mechanisms work with queuing and fragmentation to improve the efficiency and predictability of the application service levels.

This chapter gives a brief introduction to these link-layer efficiency mechanisms described in the following sections:

- Multilink PPP
- Frame Relay Fragmentation
- Header Compression

Multilink PPP

At the top level, Multilink PPP (also known as MLP or simply Multilink) provides packet interleaving, packet fragmentation, and packet resequencing across multiple logical data links. The packet interleaving, packet fragmentation, and packet resequencing are used to accommodate the fast transmission times required for sending real-time packets (for example, voice packets) across the network links. Multilink is especially useful over slow network links (that is, a network link with a link speed less than or equal to 768 kbps).

For more information about the functionality of Multilink when providing quality of service (QoS) on your network, see the "Reducing Latency and Jitter for Real-Time Traffic Using Multilink PPP" module.

Frame Relay Fragmentation

Cisco has developed the following three methods of performing Frame Relay fragmentation:

- End-to-end FRF.12 (and higher) fragmentation
- Frame Relay fragmentation using FRF.11 Annex C (and higher)
- Cisco proprietary encapsulation

For more information about Frame Relay fragmentation, see the "Frame Relay Queueing and Fragmentation at the Interface" module.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Header Compression

Header compression is a mechanism that compresses the IP header in a packet before the packet is transmitted. Header compression reduces network overhead and speeds up the transmission of Real-Time Transport Protocol (RTP) and Transmission Control Protocol (TCP) packets. Header compression also reduces the amount of bandwidth consumed when the RTP or TCP packets are transmitted.

Cisco provides two basic types of header compression: RTP header compression (used for RTP packets) and TCP header compression (used for TCP packets).

For more information about header compression, see the "Header Compression" module.

Header Compression

Header-Compression Features Roadmap

First Published: January 30, 2006
Last Updated: June 19, 2006

This feature roadmap lists the Cisco IOS features related to header-compression documented in the *Cisco IOS Quality of Service Solutions Configuration Guide* and maps them to the documents in which they appear. The roadmap is organized so that you can select your release train and see the features in that release. Find the feature name you are searching for and click on the URL in the "Where Documented" column to access the document containing that feature.

Feature and Release Support

Table 1 lists header-compression feature support for the following Cisco IOS software release trains:

- Cisco IOS Releases 12.4T
- Cisco IOS Releases 12.2T, 12.3, and 12.3T

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 lists the most recent release of each software train first and the features in alphabetical order within the release.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Table 1 Supported Header-Compression Features

Release	Feature Name	Feature Description	Where Documented
Cisco IOS Releases 12.4T			
12.4(9)T	IPHC Profiles	The IPHC Profiles feature simplifies the way header compression is enabled on your network. IPHC profiles allow you to enable header compression and configure the related options in a profile (a kind of template file). Once you've created the IPHC profile, you can then apply (attach) the profile to one or more interfaces, subinterfaces, or Frame Relay permanent virtual circuits (PVCs) in your network.	"Header Compression" "Configuring Header Compression Using IPHC Profiles"
Cisco IOS Releases 12.2T, 12.3, and 12.3T			
12.3(11)T	Enhanced CRTP for Links with High Delay, Packet Loss, and Reordering	The Enhanced Compressed Real-Time Transport Protocol (ECRTP) for Links with High Delay, Packet Loss, and Reordering feature includes modifications and enhancements to CRTP to achieve robust operation over unreliable point-to-point links. This is accomplished by repeating updates and sending absolute (uncompressed) values in addition to delta values for selected context parameters.	"Header Compression" "Configuring RTP Header Compression"
12.3(2)T	RTP Header Compression over Satellite Links	The RTP Header Compression over Satellite Links feature allows customers to use Real-Time Transport Protocol (RTP) header compression over an asymmetric link (such as a satellite link), where the uplink and downlink connections are on separate interfaces. This feature provides improved system performance by reducing network overhead and speeding up transmission of RTP packets.	"Header Compression" "Configuring RTP Header Compression"
12.2(13)T	Class-Based RTP and TCP Header Compression	This feature allows you to configure Real-Time Transport Protocol (RTP) or Transmission Control Protocol (TCP) header compression on a per-class basis, when a class is configured within a policy map. Policy maps are created using the Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC).	"Header Compression" "Configuring Class-Based RTP and TCP Header Compression"

Header Compression

First Published: January 30, 2006
Last Updated: June 19, 2006

Header compression is a mechanism that compresses the IP header in a packet before the packet is transmitted. Cisco provides two types of header compression: RTP header compression (used for RTP packets) and TCP header compression (used for TCP packets).

This module contains a high-level overview of header compression. Before configuring header compression, you need to understand the information contained in this module.

Contents

Information About Header Compression

Before configuring header compression, you should understand the following concepts:

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Header Compression Defined

Header compression is a mechanism that compresses the IP header in a data packet before the packet is transmitted. Header compression reduces network overhead and speeds up the transmission of Real-Time Transport Protocol (RTP) and Transmission Control Protocol (TCP) packets. Header compression also reduces the amount of bandwidth consumed when the RTP or TCP packets are transmitted.

Types of Header Compression

Cisco provides the following two types of header compression:

- RTP header compression (used for RTP packets)
- TCP header compression (used for TCP packets)

Both RTP header compression and TCP header compression treat packets in a similar fashion, as described in the sections that follow.

Note RTP and TCP header compression are typically configured on a *per-interface* (or *subinterface*) basis. However, you can choose to configure either RTP header compression or TCP header compression on a *per-class* basis using the Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC). More information about class-based RTP and TCP header compression is provided later in this module.

RTP Functionality and Header Compression

RTP provides end-to-end network transport functions for applications that support audio, video, or simulation data over unicast or multicast services.

RTP provides support for real-time conferencing of groups of any size within the Internet. This support includes source identification support for gateways such as audio and video bridges, and support for multicast-to-unicast translators. RTP provides QoS feedback from receivers to the multicast group and support for the synchronization of different media streams.

RTP includes a data portion and a header portion. The data portion of RTP is a thin protocol that provides support for the real-time properties of applications, such as continuous media, including timing reconstruction, loss detection, and content identification. The header portion of RTP is considerably larger than the data portion. The header portion consists of the IP segment, the User Datagram Protocol (UDP) segment, and the RTP segment. Given the size of the IP/UDP/RTP segment combinations, it is inefficient to send the IP/UDP/RTP header without compressing it.

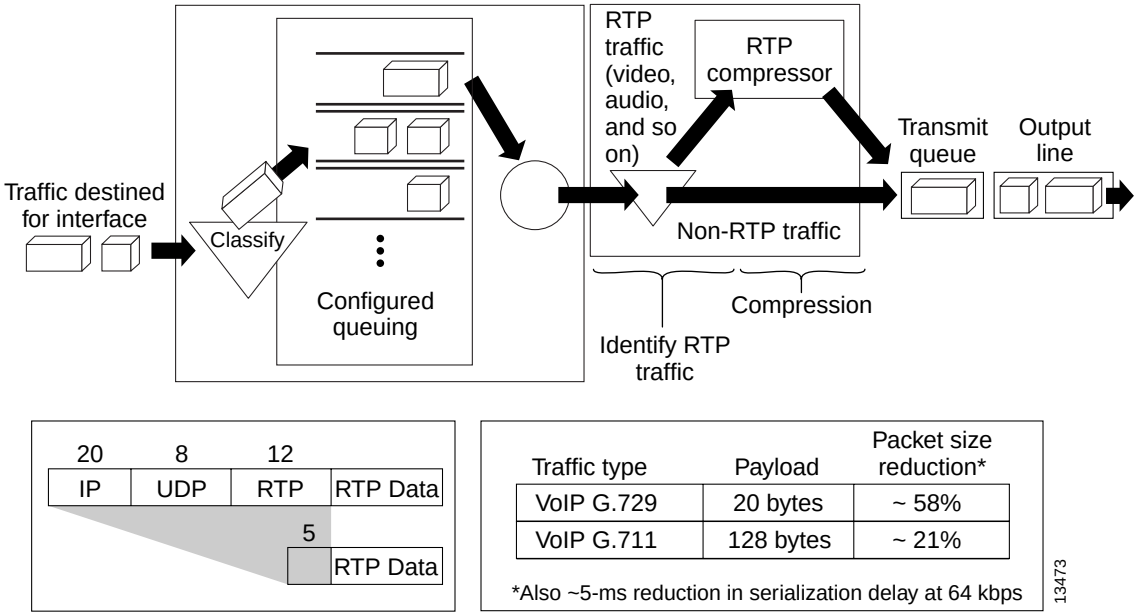
To avoid the unnecessary consumption of available bandwidth, RTP header compression is used on a link-by-link basis.

How RTP Header Compression Works

RTP header compression compresses the RTP header (that is, the combined IP, UDP, and RTP segments) in an RTP packet. Figure 1 illustrates this process and shows how RTP header compression treats incoming packets.

In this example, packets arrive at an interface and the packets are classified. After the packets are classified, they are queued for transmission according to the configured queuing mechanism.

Figure 1 RTP Header Compression



For most audio applications, the RTP packet typically has a 20- to 128-byte payload.

RTP header compression identifies the RTP traffic and then compresses the IP header portion of the RTP packet. The IP header portion consists of an IP segment, a UDP segment, and an RTP segment. In Figure 1, the minimal 20 bytes of the IP segment, combined with the 8 bytes of the UDP segment, and the 12 bytes of the RTP segment, create a 40-byte IP/UDP/RTP header. In Figure 1, the RTP header portion is compressed from 40 bytes to approximately 5 bytes.

Note RTP header compression is supported on serial interfaces using Frame Relay, HDLC, or PPP encapsulation. It is also supported over ISDN interfaces.

Why Use RTP Header Compression

RTP header compression accrues major gains in terms of packet compression because although several fields in the header change in every packet, the difference from packet to packet is often constant, and therefore the second-order difference is zero. The decompressor can reconstruct the original header without any loss of information.

RTP header compression also reduces overhead for multimedia RTP traffic. The reduction in overhead for multimedia RTP traffic results in a corresponding reduction in delay; RTP header compression is especially beneficial when the RTP payload size is small, for example, for compressed audio payloads of 20 to 50 bytes.

Use RTP header compression on any WAN interface where you are concerned about bandwidth and where there is a high portion of RTP traffic. RTP header compression can be used for media-on-demand and interactive services such as Internet telephony. RTP header compression provides support for real-time conferencing of groups of any size within the Internet. This support includes source

identification support for gateways such as audio and video bridges, and support for multicast-to-unicast translators. RTP header compression can benefit both telephony voice and multicast backbone (MBONE) applications running over slow links.

Note Using RTP header compression on any high-speed interfaces—that is, anything over T1 speed—is not recommended. Any bandwidth savings achieved with RTP header compression may be offset by an increase in CPU utilization on the router.

TCP Functionality and Header Compression

TCP provides a reliable end-to-end network transport for applications such as FTP, Telnet, and HTTP. TCP uses the connectionless service provided by IP.

TCP includes a data portion and a header portion. The header portion of TCP is considerably larger than the data portion. The header portion consists of the IP segment and the TCP segment. Given the size of the TCP/IP segment combinations, it is inefficient to send the TCP/IP header without compressing it.

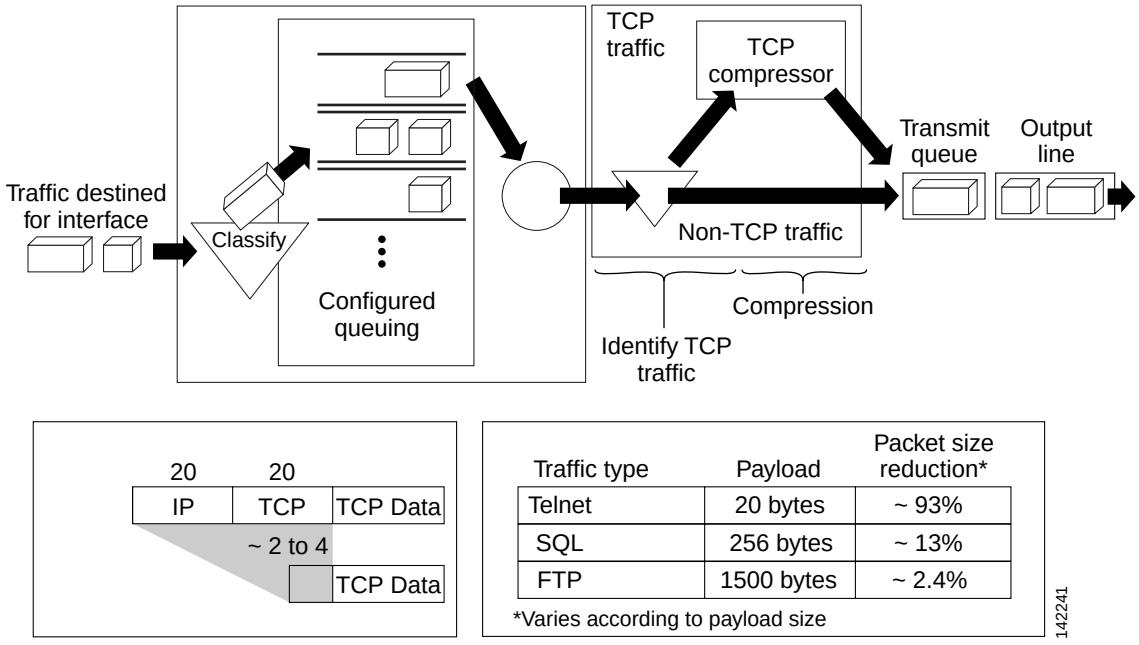
To avoid the unnecessary consumption of available bandwidth, TCP header compression is used on a link-by-link basis.

How TCP Header Compression Works

TCP header compression compresses the TCP header (that is, the combined IP and TCP segments) in a TCP packet. Figure 2 illustrates this process and shows how TCP header compression treats incoming packets.

In this example, packets arrive at an interface and the packets are classified. After the packets are classified, they are queued for transmission according to the configured queuing mechanism.

Figure 2 TCP Header Compression



For TCP applications, the TCP packet typically has a 1- to 1500-byte payload.

TCP header compression identifies the TCP traffic and then compresses the IP header portion of the TCP packet. The IP header portion consists of an IP segment and a TCP segment. In Figure 2, the 20 bytes of the IP segment, combined with the 20 bytes of the TCP segment, creates a 40-byte TCP/IP header. In Figure 2, the TCP/IP header portion is compressed from 40 bytes to approximately 2 to 4 bytes.

Note TCP header compression is supported on serial interfaces using Frame Relay, HDLC, or PPP encapsulation. It is also supported over ISDN interfaces.

Why Use TCP Header Compression

TCP header compression accrues major gains in terms of packet compression because although several fields in the header change in every packet, the difference from packet to packet is often constant, and therefore the second-order difference is zero. The decompressor can reconstruct the original header without any loss of information.

TCP header compression also reduces overhead. The reduction in overhead for TCP traffic results in a corresponding reduction in delay; TCP header compression is especially beneficial when the TCP payload size is small, for example, for interactive traffic such as Telnet.

Use TCP header compression on any WAN interface where you are concerned about bandwidth and where there is a high portion of TCP traffic.

Note Using TCP header compression on any high-speed interfaces—that is, anything over T1 speed—is not recommended. Any bandwidth savings achieved with TCP header compression may be offset by an increase in CPU utilization on the router.

Class-Based Header Compression Functionality

Class-based header compression uses the same functionality as RTP and TCP header compression described earlier in this module. That is, class-based header compression treats packets the same way as described in the "RTP Functionality and Header Compression" and the "TCP Functionality and Header Compression" sections of this module, respectively.

Class-based header compression is simply another method you can choose when you configure either RTP header compression or TCP header compression on your network.

RTP and TCP header compression are typically configured on a *per-interface* (or subinterface) basis. Class-based header compression (RTP or TCP) is configured on a *per-class* basis using the MQC.

The MQC is a CLI that allows you to create classes within policy maps (traffic policies) and then attach the policy maps to interfaces. The policy maps are used to configure specific QoS features (such as RTP or TCP header compression) on your network.

For more information about the MQC, see the "Applying QoS Features Using the MQC" module.

Why Use Class-Based Header Compression

Class-based header compression allows you to compress (and then decompress) a subset of the packets on your network. Class-based header compression acts as a filter; it allows you to specify at a much finer level the packets that you want to compress using either RTP header compression or TCP header compression.

For example, instead of compressing all RTP (or all TCP) packets that are traversing your network, you can configure RTP header compression to compress only those packets that meet certain criteria (for example, protocol type "ip" in a class called "voice)."

IPHC Profiles and Header Compression

One method of configuring header compression on your network is to use an IP header compression (IPHC) profile. An IPHC profile is a kind of template within which you can configure the type of header compression you want to use, set all of the optional features and settings for header compression, and then apply the profile to an interface, subinterface, or Frame Relay permanent virtual circuit (PVC).

Why Use IPHC Profiles

Provides More Flexibility

You can enable header compression (along with any optional settings you want to use) *once* in an IPHC profile, and then apply that IPHC profile to as many interfaces, subinterfaces, or Frame Relay PVCs as needed.

Where to Go Next

Where you go next depends on the type of header compression that you want to configure, and whether you want to use IPHC profiles to configure header compression.

- To configure RTP header compression, see the "Configuring RTP Header Compression" module.
- To configure TCP header compression, see the "Configuring TCP Header Compression" module.
- To configure class-based RTP or TCP header compression, see the "Configuring Class-Based RTP and TCP Header Compression" module.
- To configure header compression using IPHC profiles, see the "Configuring Header Compression Using IPHC Profiles" module.

Additional References

The following sections provide references related to header compression.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
MQC	"Applying QoS Features Using the MQC" module
RTP header compression	"Configuring RTP Header Compression" module
TCP header compression	"Configuring TCP Header Compression" module
Class-based RTP and TCP header compression	"Configuring Class-Based RTP and TCP Header Compression" module
IPHC profiles and header compression	"Configuring Header Compression Using IPHC Profiles" module

Standards

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

MIBs

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

RFCs

RFC	Title
RFC 1144	*Compressing TCP/IP Headers for Low-Speed Serial Links*
RFC 2507	*IP Header Compression*
RFC 2508	*Compressing IP/UDP/RTP Headers for Low-Speed Serial Links*
RFC 3544	*IP Header Compression over PPP*
RFC 3545	*Enhanced Compressed RTP (CRTP) for Links with High Delay, Packet Loss and Reordering*
RFC 3550	*A Transport Protocol for Real-Time Applications*

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Glossary

compression—The running of a data set through an algorithm that reduces the space required to store the data set or the bandwidth required to transmit the data set.

decompression—The act of reconstructing a compressed header.

HDLC—High-Level Data Link Control. A bit-oriented synchronous data link layer protocol developed by International Organization for Standardization (ISO). Derived from Synchronous Data Link Control (SDLC), HDLC specifies a data encapsulation method on synchronous serial links using frame characters and checksums.

header—A chain of subheaders.

incorrect decompression—The circumstance in which a compressed and then decompressed header is different from the uncompressed header. This variance is usually due to a mismatched context between the compressor and decompressor or bit errors during transmission of the compressed header.

ISDN—Integrated Services Digital Network. A communication protocol offered by telephone companies that permits telephone networks to carry data, voice, and other source traffic.

MQC—Modular Quality of Service Command-Line Interface. The MQC allows you to create traffic classes and policy maps and then attach the policy maps to interfaces. The policy maps apply QoS features to your network.

PPP—Point-to-Point Protocol. A protocol that provides router-to-router and host-to-network connections over synchronous and asynchronous circuits.

regular header—A normal, uncompressed header. A regular header does not carry a context identifier (CID) or generation association.

RTP—Real-Time Transport Protocol. A protocol that is designed to provide end-to-end network transport functions for applications that transmit real-time data, such as audio, video, or simulation data, over unicast or multicast network services. RTP provides such services as payload type identification, sequence numbering, timestamping, and delivery monitoring to real-time applications.

subheader—An IPv6 base header, an IPv6 extension header, an IPv4 header, a UDP header, an RTP header, or a TCP header.

TCP—Transmission Control Protocol. A connection-oriented transport layer protocol that provides reliable full-duplex data transmission. TCP is part of the TCP/IP protocol stack.

UDP—User Datagram Protocol. A connectionless transport layer protocol in the TCP/IP protocol stack. UDP is a simple protocol that exchanges datagrams without acknowledgments or guaranteed delivery, requiring that error processing and retransmission be handled by other protocols. UDP is defined in RFC 768.

Configuring RTP Header Compression

First Published: January 30, 2006
Last Updated: June 19, 2006

Header compression is a mechanism that compresses the IP header in a packet before the packet is transmitted. Header compression reduces network overhead and speeds up the transmission of either Real-Time Transport Protocol (RTP) or Transmission Control Protocol (TCP) packets.

Cisco provides two types of header compression: RTP header compression and TCP header compression. This module describes the concepts and tasks related to configuring RTP header compression.

Note RTP header compression is configured on a per-interface (or subinterface) basis. If you want to configure RTP header compression on a per-class basis, see the "Configuring Class-Based RTP and TCP Header Compression" module.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Configuring RTP Header Compression" section on page 22.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Configuring RTP Header Compression

- Before configuring RTP header compression, read the information in the "Header Compression" module.
- You must configure RTP header compression on both ends of the network.

Information About Configuring RTP Header Compression

Before configuring RTP header compression, you should understand the following concepts:

Configurable RTP Header-Compression Settings

With RTP header compression, you can configure the maximum size of the compressed IP header, the maximum time between transmitting full-header packets, and the maximum number of compressed packets between full headers. These settings are configured using the following three commands:

- **ip header-compression max-header**
- **ip header-compression max-time**
- **ip header-compression max-period**

The **ip header-compression max-header** command allows you to define the maximum size of the IP header of a packet to be compressed. Any packet with an IP header that exceeds the maximum size is sent uncompressed.

The **ip header-compression max-time** command allows you to specify the maximum time between transmitting full-header packets, and the **ip header-compression max-period** command allows you to specify the maximum number of compressed packets between full headers. With the **ip header-compression max-time** and **ip header-compression max-period** commands, the full-header packet is transmitted at the specified time period or when the maximum number of packets is reached, respectively. The counters for both the time period and the number of packets sent are reset after the full-header packet is sent.

For more information about these commands, see the *Cisco IOS Quality of Service Solutions Command Reference*.

RTP Header-Compression Keywords

When you configure RTP header compression, you can specify the circumstances under which the RTP packets are compressed and the format that is used when the packets are compressed. These circumstances and formats are defined by the following keywords:

- **passive**
- **iphc-format**
- **ietf-format**

These keywords (described below) are available with many of the quality of service (QoS) commands used to configure RTP header compression, such as the **ip rtp header-compression** command. For more information about the **ip rtp header-compression** command, these keywords, and the other QoS commands, see the *Cisco IOS Quality of Service Solutions Command Reference*.

The passive Keyword

By default, the **ip rtp header-compression** command compresses outgoing RTP traffic. If you specify the **passive** keyword, outgoing RTP traffic is compressed only if *incoming* RTP traffic on the *same* interface is compressed. If you do not specify the **passive** keyword, *all* outgoing RTP traffic is compressed.

The **passive** keyword is ignored on PPP interfaces.

The iphc-format Keyword

The **iphc-format** keyword indicates that the IP Header Compression (IPHC) format of header compression will be used. For PPP and HDLC interfaces, when the **iphc-format** keyword is specified, TCP header compression is also enabled. Since both RTP and TCP header compression are enabled, both UDP and TCP packets are compressed.

The **iphc-format** keyword includes checking whether the destination port number is even and is in the ranges of 16,385 to 32,767 (for Cisco audio) or 49,152 to 65,535 (for Cisco video). Valid RTP packets that meet the criteria (that is, the port number is even and is within the specified range) are compressed using the compressed RTP packet format. Otherwise, packets are compressed using the less-efficient compressed non-TCP packet format.

The **iphc-format** keyword is not available for interfaces that use Frame Relay encapsulation.

Note The header compression format (in this case, IPHC) must be the same at *both* ends of the network. That is, if you specify the **iphc-format** keyword on the local router, you must also specify the **iphc-format** keyword on the remote router.

The ietf-format Keyword

The **ietf-format** keyword indicates that the Internet Engineering Task Force (IETF) format of header compression will be used. For HDLC interfaces, the **ietf-format** keyword compresses only UDP packets. For PPP interfaces, when the **ietf-format** keyword is specified, TCP header compression is also enabled. Since both RTP header compression and TCP header compression are enabled, both UDP packets and TCP packets are compressed.

With the **ietf-format** keyword, any even destination port number higher than 1024 can be used. Valid RTP packets that meet the criteria (that is, the port number is even and is higher than 1024) are compressed using the compressed RTP packet format. Otherwise, packets are compressed using the less-efficient compressed non-TCP packet format.

The **ietf-format** keyword is not available for interfaces that use Frame Relay encapsulation.

Note The header compression format (in this case, IETF) must be the same at *both* ends of the network. That is, if you specify the **ietf-format** keyword on the local router, you must also specify the **ietf-format** keyword on the remote router.

Enhanced RTP Header Compression

The Cisco IOS Release 12.3(11)T introduced a feature that enhances the functionality of RTP header compression. This feature is called Enhanced CRTP for Links with High Delay, Packet Loss, and Reordering (ECRTP).

The ECRTP feature is also known as Enhanced RTP Header Compression. It includes modifications and enhancements to RTP header compression to achieve robust operation over unreliable point-to-point links. This is accomplished by repeating updates and sending absolute (uncompressed) values in addition to delta values for selected context parameters.

During compression of an RTP stream, a session context is defined. For each context, the session state is established and shared between the compressor and the decompressor. The context state consists of the full IP/UDP/RTP headers, a few first-order differential values, a link sequence number, a generation number, and a delta encoding table. Once the context state is established, compressed packets may be sent.

RTP header compression was designed for reliable point-to-point links with short delays. It does not perform well over links with a high rate of packet loss, packet reordering, and long delays. Packet loss results in context corruption, and because of long delay, packets are discarded before the context is repaired. To correct the behavior of RTP header compression over such links, several enhancements have been made to the RTP header compression functionality. The enhancements reduce context corruption by changing the way that the compressor updates the context at the decompressor; updates are repeated and include additions to full and differential context parameters.

With these enhancements, RTP header compression performs well over links with packet loss, packet reordering, and long delays.

RTP Header Compression over Satellite Links

The Cisco IOS Release 12.3(2)T introduced a feature called RTP Header Compression over Satellite Links. The RTP Header Compression over Satellite Links feature allows you to use RTP header compression over an asymmetric link (such as a satellite link), where the uplink and downlink connections are on separate interfaces. This feature provides improved system performance by reducing network overhead and speeding up transmission of RTP packets.

Periodic Refreshes of a Compressed Packet Stream

RTP header compression is a mechanism that compresses the IP header in a packet before the packet is transmitted. RTP header compression requires a context status feedback mechanism to recover when the compressed packet stream experiences packet channel loss. If the round-trip time of the packet between the uplink and the downlink is lengthy or if a feedback path does not exist, the chance of loss propagation is greatly increased when a packet is dropped from the link. For instance, if a feedback path does not exist, a compressed packet stream may never recover. This situation presents a need for a configurable option that allows periodic refreshes of the compressed packet stream using full-header packets.

The periodic-refresh Keyword

When you configure header compression, you can configure periodic refreshes of the compressed packet stream using the **periodic-refresh** keyword. The **periodic-refresh** keyword is available with the following commands:

- **ip rtp header-compression**
- **frame-relay ip rtp header-compression**
- **frame-relay map ip rtp header-compression**

For more information about these commands, see the *Cisco IOS Quality of Service Solutions Command Reference*.

Optional Disabling of Context-Status Messages

During header compression, a session context is defined. For each context, the session state is established and shared between the compressor and the decompressor. The context state consists of the full IP/UDP/RTP headers, a few first-order differential values, a link sequence number, a generation number, and a delta encoding table. This information is included in the context-status messages.

You can disable the sending of context-status messages in instances either when the time it takes for the packet to traverse the uplink and the downlink portions of the data path is greater than the refresh period (in which case, the sending of the context-status message would not be useful) or when a feedback path does not exist.

Disabling the context-status messages can be accomplished by using the **ip header-compression disable-feedback** command. For more information about this command, see the *Cisco IOS Quality of Service Solutions Command Reference*.

How to Configure RTP Header Compression

This section contains the following tasks:

Enabling RTP Header Compression on an Interface

To enable RTP header compression on an interface, perform the following steps.

Note To enable RTP header compression on an interface that uses Frame Relay encapsulation, skip these steps and complete the steps in the "Enabling RTP Header Compression on an Interface That Uses Frame Relay Encapsulation" section on page 7 instead.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **encapsulation** *encapsulation-type*
5. **ip address** *ip-address mask* [**secondary**]
6. **ip rtp header-compression** [**passive | iphc-format | ietf-format**] [**periodic-refresh**]
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [*name-tag*] **Example:** `Router(config)# interface serial0`	Configures an interface type and enters interface configuration mode. • Enter the interface type and the interface number.
Step 4	**encapsulation** *encapsulation-type* **Example:** `Router(config-if)# encapsulation ppp`	Sets the encapsulation method used by the interface. • Enter the encapsulation method.
Step 5	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 209.165.200.225 255.255.255.224`	Sets a primary or secondary IP address for an interface. • Enter the IP address and mask for the associated IP subnet.
Step 6	**ip rtp header-compression** [**passive** | **iphc-format** | **ietf-format**] [**periodic-refresh**] **Example:** `Router(config-if)# ip rtp header-compression`	Enables RTP header compression.
Step 7	**end** **Example:** `Router(config-if)# end`	(Optional) Exits interface configuration mode.

Enabling RTP Header Compression on an Interface That Uses Frame Relay Encapsulation

To enable RTP header compression on an interface that uses Frame Relay encapsulation, perform the following steps.

Restrictions

The encapsulation type is specified by using either the **cisco** or **ietf** keyword of the **frame-relay interface-dlci** command. The **cisco** keyword specifies Cisco proprietary encapsulations, and the **ietf** keyword specifies IETF encapsulations. However, note the following points about these keywords:

- Frame Relay interfaces do not support IETF encapsulations when RTP header compression is enabled. Therefore, the **ietf** keyword is not available for Frame Relay interfaces and is not listed in the command syntax shown below.
- The **cisco** keyword is available for use on point-to-point subinterfaces *only.*

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **encapsulation frame-relay**
5. **ip address** *ip-address mask* [**secondary**]
6. **frame-relay interface-dlci** *dlci* [**cisco**]
7. **frame-relay ip rtp header-compression** [**active | passive**] [**periodic-refresh**]

   or

   frame-relay map ip *ip-address dlci* [**broadcast**] **rtp header-compression** [**active | passive**] [**periodic-refresh**] [**connections** *number*]
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [*name-tag*] **Example:** `Router(config)# interface serial0`	Configures an interface type and enters interface configuration mode. • Enter the interface type and the interface number.
Step 4	**encapsulation frame-relay** **Example:** `Router(config-if)# encapsulation frame-relay`	Enables Frame Relay encapsulation.
Step 5	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 209.165.200.225 255.255.255.224`	Sets a primary or secondary IP address for an interface. • Enter the IP address and mask for the associated IP subnet.
Step 6	**frame-relay interface-dlci** *dlci* [**cisco**] **Example:** `Router(config-if)# frame-relay interface-dlci 20`	Assigns a data-link connection identifier (DLCI) to a specified Frame Relay interface on the router.
Step 7	**frame-relay ip rtp header-compression** [**active** | **passive**] [**periodic-refresh**] **Example:** `Router(config-if)# frame-relay ip rtp header-compression`	Enables RTP header compression for all Frame Relay maps on a physical interface.
	or	
	frame-relay map ip *ip-address dlci* [**broadcast**] **rtp header-compression** [**active** | **passive**] [**periodic-refresh**] [**connections** *number*] **Example:** `Router(config-if)# frame-relay map ip 10.108.175.220 180 rtp header-compression periodic-refresh`	Assigns to an IP map header-compression characteristics that differ from the compression characteristics of the interface with which the IP map is associated. • Enter the IP address, DLCI number, and any optional keywords and arguments.
Step 8	**end** **Example:** `Router(config-if)# end`	(Optional) Exits interface configuration mode.

Enabling Enhanced RTP Header Compression

The Enhanced RTP Header Compression feature (also known as ECRTP) includes modifications and enhancements to RTP header compression to achieve robust operation over unreliable point-to-point links. Enhanced RTP header compression is intended for use on networks subject to high rates of packet loss, packet reordering, and long delays. For more information about Enhanced RTP header compression, see the "Enhanced RTP Header Compression" section on page 4.

To enable enhanced RTP header compression, perform the following steps.

Prerequisites

- Configure a serial link using HDLC encapsulation or configure an interface using PPP encapsulation.
- Ensure that RTP header compression is enabled on the interface. See the "Enabling RTP Header Compression on an Interface" section on page 5.

Restrictions

Enhanced RTP header compression is not supported on interfaces that use Frame Relay encapsulation.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **encapsulation** *encapsulation-type*
5. **ip address** *ip-address mask* [**secondary**]
6. **ip rtp header-compression** [**passive** | **iphc-format** | **ietf-format**] [**periodic-refresh**]
7. **ip header-compression recoverable-loss** {**dynamic** | *packet-drops*}
8. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**interface** *type number* [*name-tag*] **Example:** `Router(config)# interface serial0`	Configures an interface type and enters interface configuration mode. • Enter the interface type and the interface number.
Step 4	**encapsulation** *encapsulation-type* **Example:** `Router(config-if)# encapsulation ppp`	Sets the encapsulation method used on the interface. • Enter the encapsulation method.
Step 5	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 209.165.200.225 255.255.255.224`	Sets a primary or secondary IP address for an interface. • Enter the IP address and mask for the associated IP subnet.
Step 6	**ip rtp header-compression** [**passive** | **iphc-format** | **ietf-format**] [**periodic-refresh**] **Example:** `Router(config-if)# ip rtp header-compression ietf-format`	Enables RTP header compression.
Step 7	**ip header-compression recoverable-loss** {**dynamic** | *packet-drops*} **Example:** `Router(config-if)# ip header-compression recoverable-loss dynamic`	Enables ECRTP on an interface. **Note** Enter the **dynamic** keyword to enable dynamic packet loss recovery, or enter the *packet-drops* argument to specify the maximum number of consecutive packet drops that are acceptable.
Step 8	**end** **Example:** `Router(config-if)# end`	(Optional) Exits interface configuration mode.

Enabling RTP Header Compression over a Satellite Link

To enable RTP header compression over a satellite link, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **ip address** *ip-address mask* [**secondary**]
5. **ip rtp header-compression** [**passive** | **iphc-format** | **ietf-format**] [**periodic-refresh**]
6. **ip header-compression disable-feedback**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [*name-tag*] **Example:** `Router(config)# interface serial0`	Configures an interface type and enters interface configuration mode. • Enter the interface type and the interface number.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 209.165.200.225 255.255.255.224`	Sets a primary or secondary IP address for an interface. • Enter the IP address and mask for the associated IP subnet.
Step 5	**ip rtp header-compression** [**passive** | **iphc-format** | **ietf-format**] [**periodic-refresh**] **Example:** `Router(config-if)# ip rtp header-compression ietf-format periodic-refresh`	Enables RTP header compression. **Note** For RTP header compression over a satellite link, use the **periodic-refresh** keyword.
Step 6	**ip header-compression disable-feedback** **Example:** `Router(config-if)# ip header-compression disable-feedback`	(Optional) Disables the context status feedback messages from the interface or link.
Step 7	**end** **Example:** `Router(config-if)# end`	(Optional) Exits interface configuration mode.

Specifying the Header-Compression Settings

With RTP header compression, you can configure the maximum size of the compressed IP header, the time period for an automatic resend of full-header packets, and the number of packets transmitted before a new full-header packet is sent.

To specify these header-compression settings, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **interface** *type number* [*name-tag*]
4. **ip header-compression max-header** *max-header-size*

   or

   ip header-compression max-time *length-of-time*

   or

   ip header-compression max-period *number-of-packets*
5. end

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [*name-tag*] **Example:** `Router(config)# interface serial0`	Configures an interface type and enters interface configuration mode. • Enter the interface type and the interface number.
Step 4	**ip header-compression max-header** *max-header-size* **Example:** `Router(config-if)# ip header-compression max-header 100`	Specifies the maximum size of the compressed IP header. • Enter the maximum size of the compressed IP header, in bytes.
	or	
	ip header-compression max-time *length-of-time* **Example:** `Router(config-if)# ip header-compression max-time 30`	Specifies the maximum amount of time to wait before the compressed IP header is refreshed. • Enter the amount of time, in seconds.
	or	
	ip header-compression max-period *number-of-packets* **Example:** `Router(config-if)# ip header-compression max-period 160`	Specifies the maximum number of compressed packets between full headers. • Enter the maximum number of compressed packets between full headers.
Step 5	**end** **Example:** `Router(config-if)# end`	(Optional) Exits interface configuration mode.

Changing the Number of Header-Compression Connections

For PPP and HDLC interfaces, the default is 16 compression connections. For interfaces that use Frame Relay encapsulation, the default is 256 compression connections.

To change the default number of header-compression connections, perform the following steps.

Implications of Changing the Number of Header-Compression Connections

Each header-compression connection sets up a compression cache entry, so you are in effect specifying the maximum number of cache entries and the size of the cache. Too few cache entries for the specified interface can lead to degraded performance, and too many cache entries can lead to wasted memory. Choose the number of header-compression connections according to the network requirements.

Restrictions

Header-Compression Connections on HDLC and Frame Relay Interfaces

For HDLC interfaces and Frame Relay interfaces (that is, interfaces that use Frame Relay encapsulation), the number of header-compression connections on *both sides* of the network must match. That is, the number configured for use on the local router must match the number configured for use on the remote router.

Header-Compression Connections on PPP Interfaces

For PPP interfaces, if the header-compression connection numbers on both sides of the network do not match, the number used is "autonegotiated." That is, any mismatch in the number of header-compression connections between the local router and the remote router will be automatically negotiated to the lower of the two numbers. For example, if the local router is configured to use 128 header-compression connections, and the remote router is configured to use 64 header-compression connections, the negotiated number will be 64.

Note This autonegotiation function applies to PPP interfaces *only*. For HDLC interfaces and interfaces that use Frame Relay encapsulation, no autonegotiation occurs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **ip rtp compression-connections** *number*

   or

   frame-relay ip rtp compression-connections *number*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [*name-tag*] **Example:** `Router(config)# interface serial0`	Configures an interface type and enters interface configuration mode. • Enter the interface type and the interface number.
Step 4	**ip rtp compression-connections** *number* **Example:** `Router(config-if)# ip rtp compression-connections 150`	Specifies the total number of RTP header-compression connections that can exist on an interface. • Enter the number of compression connections. **Note** This command can be used for PPP interfaces, HDLC interfaces, or interfaces that use Frame Relay encapsulation.
	or	
	frame-relay ip rtp compression-connections *number* **Example:** `Router(config-if)# frame-relay ip rtp compression-connections 150`	Specifies the maximum number of RTP header-compression connections that can exist on a Frame Relay interface (that is, an interface using Frame Relay encapsulation). • Enter the number of compression connections. **Note** This command can be used for interfaces that use Frame Relay encapsulation *only*.
Step 5	**end** **Example:** `Router(config-if)# end`	(Optional) Exits interface configuration mode.

Displaying Header-Compression Statistics

You can display header-compression statistics, such as the number of packets sent, received, and compressed, by using either the **show ip rtp header-compression** command or the **show frame-relay ip rtp header-compression** command.

To display header-compression statistics, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show ip rtp header-compression** [*interface-type interface-number*] [**detail**]

   or

show frame-relay ip rtp header-compression [**interface** *type number*]

3. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip rtp header-compression** [*interface-type interface-number*] [**detail**] **Example:** `Router# show ip rtp header-compression`	Displays RTP header-compression statistics for one or all interfaces.
	or	
	show frame-relay ip rtp header-compression [**interface** *type number*] **Example:** `Router# show frame-relay ip rtp header-compression`	Displays Frame Relay RTP header-compression statistics for one or all interfaces.
Step 3	**end** **Example:** `Router# end`	(Optional) Exits privileged EXEC mode.

Configuration Examples for RTP Header Compression

This section provides the following configuration examples:

Enabling RTP Header Compression on an Interface: Example

In the following example, RTP header compression is enabled on serial interface 0.

```
Router> enable
Router# configure terminal
Router(config)# interface serial0
Router(config-if)# encapsulation ppp
Router(config-if)# ip address 209.165.200.225 255.255.255.224
Router(config-if)# ip rtp header-compression
Router(config-if)# end
```

Enabling RTP Header Compression on an Interface That Uses Frame Relay Encapsulation: Example

In the following example, RTP header compression is enabled on serial interface 0. Frame Relay encapsulation has been enabled on this interface by using the **encapsulation frame-relay** command.

```
Router> enable
Router# configure terminal
Router(config)# interface serial0
Router(config-if)# encapsulation frame-relay
Router(config-if)# ip address 209.165.200.225 255.255.255.224
Router(config-if)# frame-relay interface-dlci 20
Router(config-if)# frame-relay ip rtp header-compression
Router(config-if)# end
```

Enabling Enhanced RTP Header Compression: Example

In the following example, ECRTP is enabled on serial interface 0. PPP encapsulation is enabled on the interface (a prerequisite for configuring ECRTP on a serial interface). Also, dynamic loss recovery has been specified by using the **dynamic** keyword of the **ip header-compression recoverable-loss** command.

```
Router> enable
Router# configure terminal
Router(config)# interface serial0
Router(config-if)# encapsulation ppp
Router(config-if)# ip address 209.165.200.225 255.255.255.224
Router(config-if)# ip rtp header-compression ietf-format
Router(config-if)# ip header-compression recoverable-loss dynamic
Router(config-if)# end
```

Enabling RTP Header Compression over a Satellite Link: Example

In the following example, RTP header compression is enabled on the serial interface 0. In this example, serial interface 0 is a satellite link in the network topology. The **periodic-refresh** keyword has been specified, which means that the compressed IP header will be refreshed periodically. Also, the context-status messages have been turned off (disabled).

```
Router> enable
Router# configure terminal
Router(config)# interface serial0
Router(config-if)# ip address 209.165.200.225 255.255.255.224
Router(config-if)# ip rtp header-compression ietf-format periodic-refresh
Router(config-if)# ip header-compression disable-feedback
Router(config-if)# end
```

Specifying the Header-Compression Settings: Example

In the following example, the maximum size of the compressed IP header (100 bytes) has been specified by using the **ip header-compression max-header** command.

```
Router> enable
Router# configure terminal
Router(config)# interface serial0
Router(config-if)# ip header-compression max-header 100
Router(config-if)# end
```

Changing the Number of Header-Compression Connections: Example

In the following example, the number of header-compression connections has been changed to 150 by using the **ip rtp compression-connections** command.

```
Router> enable
Router# configure terminal
Router(config)# interface serial0
Router(config-if)# ip rtp compression-connections 150
Router(config-if)# end
```

Displaying Header-Compression Statistics: Example

You can use the **show ip rtp header-compression** command to display header-compression statistics such as the number of packets received, sent, and compressed. The following is sample output from the **show ip rtp header-compression** command. In this example, ECRTP has been enabled on serial interface 0.

```
Router# show ip rtp header-compression serial0

RTP/UDP/IP header compression statistics:
  Interface Serial0 (compression on, IETF, ECRTP)
    Rcvd:    1473 total, 1452 compressed, 0 errors, 0 status msgs
             0 dropped, 0 buffer copies, 0 buffer failures
    Sent:    1234 total, 1216 compressed, 0 status msgs, 379 not predicted
             41995 bytes saved, 24755 bytes sent
             2.69 efficiency improvement factor
```

```
Connect: 16 rx slots, 16 tx slots,
         6 misses, 0 collisions, 0 negative cache hits, 13 free contexts
         99% hit ratio, five minute miss rate 0 misses/sec, 0 max
```

Additional References

The following sections provide references related to configuring RTP header compression.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
Frame Relay	"Frame Relay Queueing and Fragmentation at the Interface" module
Header compression overview	"Header Compression" module
TCP header compression	"Configuring TCP Header Compression" module
Class-based RTP and TCP header compression	"Configuring Class-Based RTP and TCP Header Compression" module
IPHC profiles and header compression	"Configuring Header Compression Using IPHC Profiles" module

Standards

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

MIBs

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

RFCs

RFC	Title
RFC 2507	*IP Header Compression*
RFC 2508	*Compressing IP/UDP/RTP Headers for Low-Speed Serial Links*
RFC 3544	*IP Header Compression over PPP*
RFC 3545	*Enhanced Compressed RTP (CRTP) for Links with High Delay, Packet Loss and Reordering*

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Glossary

compression—The running of a data set through an algorithm that reduces the space required to store the data set or the bandwidth required to transmit the data set.

context—The state that the compressor uses to compress a header and that the decompressor uses to decompress a header. The context is the uncompressed version of the last header sent and includes other information used to compress and decompress the packet.

context-state packet—A special packet sent from the decompressor to the compressor to communicate a list of (TCP or NON_TCP/RTP) context identifiers (CIDs) for which synchronization has been lost. This packet is sent only over a single link, so it requires no IP header.

DLCI—data-link connection identifier. A value that specifies a permanent virtual circuit (PVC) or switched virtual circuit (SVC) in a Frame Relay network. In the basic Frame Relay specification, DLCIs are locally significant (connected devices might use different values to specify the same connection). In the Local Management Interface (LMI) extended specification, DLCIs are globally significant (DLCIs specify individual end devices).

ECRTP—Enhanced Compressed Real-Time Transport Protocol. A compression protocol that is designed for unreliable point-to-point links with long delays.

encapsulation—A method of wrapping data in a particular protocol header. For example, Ethernet data is wrapped in a specific Ethernet header before network transit. Also, when dissimilar networks are bridged, the entire frame from one network is simply placed in the header used by the data link layer protocol of the other network.

full header (header refresh)—An uncompressed header that updates or refreshes the context for a packet stream. It carries a CID that will be used to identify the context. Full headers for non-TCP packet streams also carry the generation of the context that they update or refresh.

HDLC—High-Level Data Link Control. A bit-oriented synchronous data link layer protocol developed by the International Organization for Standardization (ISO). Derived from Synchronous Data Link Control (SDLC), HDLC specifies a data encapsulation method on synchronous serial links using frame characters and checksums.

header—A chain of subheaders.

IETF—Internet Engineering Task Force. A task force that consists of over 80 working groups responsible for developing Internet standards.

IPHC—IP Header Compression. A protocol capable of compressing both TCP and UDP headers.

ISDN—Integrated Services Digital Network. A communication protocol offered by telephone companies that permits telephone networks to carry data, voice, and other source traffic.

lossy serial links—Links in a network that are prone to lose packets.

packet stream—The sequence of packets whose headers are similar and share context. For example, headers in an RTP packet stream have the same source and final destination address and the same port numbers in the RTP header.

PPP—Point-to-Point Protocol. A protocol that provides router-to-router and host-to-network connections over synchronous and asynchronous circuits.

regular header—A normal, uncompressed header. A regular header does not carry a context identifier (CID) or generation association.

RTP—Real-Time Transport Protocol. A protocol that is designed to provide end-to-end network transport functions for applications that transmit real-time data, such as audio, video, or simulation data, over unicast or multicast network services. RTP provides such services as payload type identification, sequence numbering, timestamping, and delivery monitoring to real-time applications.

subheader—An IPv6 base header, an IPv6 extension header, an IPv4 header, a UDP header, an RTP header, or a TCP header.

Feature Information for Configuring RTP Header Compression

lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

For information on a feature in this technology that is not documented here, see the "Header-Compression Features Roadmap" module.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Configuring RTP Header Compression

Feature Name	Releases	Feature Information
RTP Header Compression over Satellite Links	12.3(2)T	The RTP Header Compression over Satellite Links feature allows customers to use RTP header compression over an asymmetric link (such as a satellite link), where the uplink and downlink connections are on separate interfaces. The following sections provide information about this feature: • RTP Header Compression over Satellite Links, page 4 • Enabling RTP Header Compression over a Satellite Link, page 10
Enhanced CRTP for Links with High Delay, Packet Loss and Reordering	12.3(11)T	The Enhanced Compressed Real-Time Transport Protocol (ECRTP) for Links with High Delay, Packet Loss, and Reordering feature includes modifications and enhancements to CRTP to achieve robust operation over unreliable point-to-point links. This is accomplished by repeating updates and sending absolute (uncompressed) values in addition to delta values for selected context parameters. The following sections provide information about this feature: • Enhanced RTP Header Compression, page 4 • Enabling Enhanced RTP Header Compression, page 9

Quality of Service Solutions

QoS: CBQoS MIB Index Enhancements

First Published: October 31, 2005
Last Updated: December 4, 2006

The QoS: Class-Based Quality of Service (CBQoS) MIB Index Enhancements feature introduces persistence across all CBQoS MIB indexes including cbQosConfigIndex, cbQosObjectsIndex, and cbQosPolicyIndex.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for the QoS: CBQoS MIB Index Enhancements" section on page 9.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for the QoS: CBQoS MIB Index Enhancements

- Simple Network Management Protocol (SNMP) must be installed and enabled on the label switch routers (LSRs).
- You must enable ifMIB persistence by issuing the **snmp-server ifindex persist** command. Then issue the **snmp mib persist cbqos** command to enable CBQoS MIB index persistence.

Note If you issue the **snmp mib persist cbqos** command before the **snmp-server ifindex persist** command, you receive a prompt requesting you to enable ifIndex persistence first.

Restrictions for the QoS: CBQoS MIB Index Enhancements

If the internal hashing of configuration strings causes too many collisions, NVRAM storage may become tight. You can issue the **more nvram** command to display a new collision file called cbqos-mib to help you keep track of the size.

Information About the QoS: CBQoS MIB Index Enhancements

To use the QoS: CBQoS MIB Index Enhancements feature, you should understand the following concepts:

Feature Overview of the QoS: CBQoS MIB Index Enhancements

The cbQosConfigIndex, cbQosObjectsIndex, and cbQosPolicyIndex are volatile because when a networking device reboots, the index numbers may change. This happens because system rebooting can cause the order of the Modular QoS CLI (MQC) configuration to differ from the actual configuration order, which is user-driven and unpredictable. As a result, you must read the MIB frequently to extract statistical and configuration information. Therefore, once a reload has occurred, the MIB has to be repolled to reestablish the indexes to the data stored in the CBQoS MIB.

Traditionally, MIB persistence is handled by Cisco IOS APIs, which save the index and key information to NVRAM. The data is then retrieved and repopulated after reloading. However, this approach does not work well for the current implementation of the cbQosObjectsIndex because of the large amount of information that needs to be saved.

An index encoding scheme based on configuration entries instead of operational sequence is being implemented to provide persistent indexes on router reload so that MIB information retains the same set of object values each time that a networking device reboots.

Benefits of the QoS: CBQoS MIB Index Enhancements

Provide a Method to Produce MIB Indexes

These enhancements provide a repeatable method for generating MIB indexes so that they do not change between reboots.

Reduce Complexity for Network Management Applications

The complexity of configuring and correlating statistics objects is reduced, making it easier for network management applications to gather accurate information.

Maintain Compatibility with Previous MIBs

You do not need to make any changes to your Network Management Station (NMS) software since this feature is an infrastructure improvement that is backward compatible with older MIBs.

How to Configure the QoS: CBQoS MIB Index Enhancements

This section contains the following procedures:

- Enabling Cisco IOS MIB and CBQoS MIB Index Persistence, page 3 (required)
- Verifying CBQoS MIB Index Persistence, page 5 (optional)

Enabling Cisco IOS MIB and CBQoS MIB Index Persistence

Perform this task to enable Cisco IOS MIB and CBQoS MIB index persistence.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server ifindex persist**
4. **snmp mib persist** [**event** | **expression** | **circuit** | **cbqos**]
5. **end**
6. **write mib-data**

   or

   write

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server ifindex persist** **Example:** `Router(config)# snmp-server ifindex persist`	Enables Cisco IOS MIB index (ifIndex) persistence.
Step 4	**snmp mib persist** [**event** | **expression** | **circuit** | **cbqos**] **Example:** `Router(config)# snmp mib persist cbqos`	Enables MIB persistence. • The optional **event** keyword enables Event MIB persistence. • The optional **expression** keyword enables Expression MIB persistence. • The optional **circuit** keyword enables Circuit MIB persistence. • The optional **cbqos** keyword enables CBQoS MIB persistence. **Note** If you have not enabled Cisco IOS MIB index (ifIndex) persistence (Step 3), the following message appears when you issue the **snmp mib persist cbqos** command: `Enable 'snmp-server ifindex persist' for persist cbqos index`
Step 5	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.
Step 6	write mib-data or write **Example:** `Router# write mib-data` or `Router# write`	Saves CBQoS MIB data to NVRAM.

Verifying CBQoS MIB Index Persistence

Perform the following task to verify that CBQoS MIB index persistence has been enabled.

SUMMARY STEPS

1. **enable**
2. show running-config | include cbqos
3. exit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show running-config | include cbqos** **Example:** `Router# show running-config | include cbqos`	Displays the configuration information currently running on the router. **Note** The information should include snmp mib persist cbqos.
Step 3	**exit** **Example:** `Router# exit`	Returns to user EXEC mode.

Configuration Examples for the QoS: CBQoS MIB Index Enhancements

This section provides the following configuration examples:

- Enabling Cisco IOS MIB and CBQoS MIB Index Persistence: Example, page 6
- Verifying Cisco IOS MIB and CBQoS MIB Index Persistence: Examples, page 6

Enabling Cisco IOS MIB and CBQoS MIB Index Persistence: Example

The following example enables Cisco IOS MIB (ifIndex) and CBQoS MIB index persistence:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# snmp-server ifindex persist
Router(config)# snmp mib persist cbqos
```

Verifying Cisco IOS MIB and CBQoS MIB Index Persistence: Examples

The following examples verify that Cisco IOS MIB (ifIndex) and CBQoS MIB index persistence have been configured:

```
Router# show running-config | include cbqos
snmp mib persist cbqos

Router# show running-config | include persist
snmp-server ifindex persist
snmp mib persist cbqos
```

Additional References

The following sections provide references related to the QoS: CBQoS MIB Index Enhancements.

Related Documents

Related Topic	Document Title
SNMP commands	*Cisco IOS Network Management Command Reference*
SNMP configuration tasks, MIB persistence	"Configuring SNMP Support" module
Other documentation	For information on using SNMP MIB features, see the appropriate documentation for your network management system.

Standards

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

MIBs

MIB	MIBs Link
CISCO-CLASS-BASED-QOS-MIB, Revision 13 **Note** The CISCO-CLASS-BASED-QOS-MIB is actually two MIBs: the CISCO-CLASS-BASED-QOS-MIB and the CISCO-CLASS-BASED-QOS-CAPABILITY-MIB.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2233	The Interfaces Group MIB using SMIv2

Technical Assistance

Description	Link
The Cisco Technical Support & Documentation website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, tools, and technical documentation. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Quality of Service Solutions Command Reference* at http://www.cisco.com/en/US/docs/ios/qos/command/reference/qos_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

- **snmp mib persist**

Feature Information for the QoS: CBQoS MIB Index Enhancements

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for the QoS: CBQoS MIB Index Enhancements

Feature Name	Releases	Feature Information
QoS: CBQoS MIB Index Enhancements	12.4(4)T, 12.0(32)S, 12.2(31)SB2	The CBQoS MIB Index Enhancements feature introduces persistence across all CBQoS MIB indexes including cbQosConfigIndex, cbQosObjectsIndex, and cbQosPolicyIndex. In 12.4(4)T, this feature was introduced. In 12.0(32)S, this feature was integrated into the release. In 12.2(31)SB2, support for the Cisco 10000 Series routers and the Cisco 7304 router was introduced.

Glossary

LSR—label switch router. A Multiprotocol Label Switching (MPLS) node that can forward native Layer 3 packets. The LSR forwards a packet based on the value of a label attached to the packet.

MIB—Management Information Base. A database of network management information that is used and maintained by a network management protocol such as Simple Network Management Protocol (SNMP). The value of a MIB object can be changed or retrieved by using SNMP commands, usually through a network management system. MIB objects are organized in a tree structure that includes public (standard) and private (proprietary) branches.

MQC—Modular Quality of Service (QoS) Command-Line Interface (CLI). A way to specify a traffic class independently of QoS policies by defining a common command syntax and resulting set of QoS behaviors across platforms. This modelreplaces the previous one of defining unique syntax for each QoS feature and for each platform.

NMS—network management station. A powerful, well-equipped computer (typically an engineering workstation) that is used by a network administrator to communicate with other devices in the network. An NMS is typically used to manage network resources, gather statistics, and perform a variety of network administration and configuration tasks.

policy map—Any defined rule that determines the use of resources within the network. A QoS policy map identifies the traffic class to which it applies and the instructions for one or more actions to take on that traffic.

QoS—quality of service. A measure of performance for a transmission system that reflects its transmission quality and service availability. Quality of service focuses on achieving appropriate network performance for networked applications; it is superior to best effort performance.

SNMP—Simple Network Management Protocol. A management protocol used almost exclusively in TCP/IP networks. SNMP provides a means for monitoring and controlling network devices and for managing configurations, statistics collection, performance, and security.

Per-Session QoS

First Published: March 20, 2006
Last Updated: January 14, 2008

The Per-Session QoS feature is one of two features bundled with the QoS: Broadband Aggregation Enhancements—Phase 1 feature. The Per-Session QoS feature provides the ability to apply quality of service (QoS) features (such as traffic classification, shaping, queueing, and policing) on a per-session basis. The Per-Session QoS feature can be configured either using a RADIUS server or using the framework available on the Intelligent Service Gateway (ISG).

Note The Per-Session QoS feature can also be configured using a virtual template (for PPP sessions only). Using a virtual template is considered a "legacy" method but is still an available option for those familiar with virtual templates. For more information about using virtual templates to configure this feature, see the "Per-Session QoS and Virtual Templates" section on page 6.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your Cisco IOS software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Per-Session QoS" section on page 20.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Per-Session QoS

- The PPP or IP sessions are enabled.

Note This document uses the generic term PPP to cover all protocol types. Examples of protocols include PPP over Ethernet (PPPoE) and PPP over ATM (PPPoA). The specific protocol supported varies by platform. For example, the Cisco 7600 series router does not support PPPoA or PPP over Ethernet over ATM (PPPoEoA). For information about the Cisco 7600 series router, see the *Cisco 7600 Series Cisco IOS Configuration Guide* for the Cisco IOS release you are using.

- Layer 2 Tunneling Protocol (L2TP) resequencing is disabled.

Note This prerequisite does not apply to the Cisco 7600 series router. L2TP is not supported on the Cisco 7600 series router.

- Traffic classes and policy maps have been configured with the QoS feature (for example, traffic policing or traffic shaping) to be applied to the network traffic. Depending on the needs of your network, multiple traffic classes and policy maps may be required.

RADIUS-Server-Specific Prerequisites

Only if you are using a RADIUS server the following prerequisites apply:

- Authentication, authorization, and accounting (AAA) must be enabled.
- The RADIUS server must be configured.
- The service profile on the RADIUS server must be created.

Restrictions for Per-Session QoS

This feature does not support the following:

- L2TP sequencing.
- Packet dropping (packet discarding). That is, this feature does not allow you to discard packets using the **drop** command.

- The Multilink PPP (MLPPP) protocol. That is, multilink bundles are not supported in either a PPP Termination and Aggregation (PTA) configuration or an L2TP configuration.

Note MLPPP is supported on the Cisco 7600 series router.

- ATM interfaces (that is, PPPoA sessions) for Cisco IOS Release 12.2(33)SRC.

Restrictions for Per-Session QoS (Cisco 7600 Series Routers)

The following restrictions apply to the Cisco 7600 series router only.

QoS Features Supported

- Queueing features are not supported in the ingress (incoming) direction of a router in an IP session. This means that traffic shaping, priority queueing such as low latency queueing (LLQ), class-based weighted fair queueing (CBWFQ), and weighted random early detection (WRED) are not supported. Features that can be configured are traffic policing and traffic marking in either the class-default class or any of the user-defined classes, as shown in the following example:

```
policy-map sess_ingress
 class c1
  police 2000000
  set ip precedence 4
 class class-default
  police 5000000
  set ip precedence 1
```

Note This restriction does not apply at the subinterface level in the ingress direction. That is, LLQ and traffic shaping are supported in the ingress direction. CBWFQ and WRED are not supported. For more information, see the "IP Subscriber Awareness over Ethernet module.

Functionality Supported in Egress Policy Maps

- A policy map (in the egress direction) used in an IP session can have *only* packet marking enabled in the user-defined class. No other QoS features (for instance, traffic policing, LLQ, WRED, or traffic shaping) can be enabled. This means that the simplified configuration shown below would not be supported.

```
policy-map sess_egress
 class c1
  police/priority/bandwidth/wred/shape
```

The simplified configuration shown below would be supported.

```
policy-map sess_egress
 class c1
  set <name> <value>
```

However, all QoS features *can* be configured in the class-default class, as illustrated below.

```
policy-map sess_egress
 class class-default
  police/priority/bandwidth/wred/shape/set
```

- A hierarchical policy map (in the egress direction) on a IP session is supported, but the child policy map must be attached to the class class-default of the parent policy map as illustrated in the simplified configuration below.

```
policy-map sess_egress
 class class-default
  <Queueing feature like traffic shaping or bandwidth remaining ratio>
  service-policy child
```

Note None of the restrictions that apply to a "flat" policy map (that is, a policy map not in a hierarchical policy map structure) in the egress or outgoing direction on a session apply to the child policy map. A simplified configuration illustrating this point is shown below.

```
policy-map child
 class voip
  police 9000
  priority level 1
 class iptv
  police 4193000
  priority level 2
  set cos 4
 class gaming
  bandwidth 1000 (kbps)
 class class-default
  set cos 1
```

Fields Used for Classifying Traffic (Ingress and Egress Direction)

Traffic in both the ingress and egress direction can be classified (matched) on the basis of characteristics or attributes such as the following:

- Ip precedence value
- Differentiated services code point (DSCP) value
- Class of service (CoS) value and CoS-inner value (of a Layer 2 QinQ packet)
- Access control list (ACL) number
- VLAN and inner-VLAN numbers

Combinations of these characteristics or attributes are allowed with the following restrictions:

- A combination of the CoS-inner setting and ACL number is not supported.
- While the command-line interface (CLI) does allow a configuration that contains two **match cos** commands, the **match-any** keyword must be used with the **class-map** command to make such a configuration meaningful.
- The **match vlan** and **match vlan-inner** commands are supported at the main interface level only.

Fields Used for Marking Traffic (Ingress and Egress Direction)

Traffic in both the ingress and egress direction can be marked on the basis of characteristics or attributes such as the following:

- Ip precedence value
- DSCP value
- CoS value
- CoS-inner value (in the egress direction *only*)

If a **set** command is specified, note the following points:

- Specifying both **set cos 4** and **set cos 5** in the same traffic class causes the **show policy-map** command to display only **set cos 5** in the show command output.
- Specifying both the **set ip prec 5** command and the **set dscp cs6** command in the same class causes the **show policy-map** command to display only **set dscp cs6** in the **show** command output.

Information About Per-Session QoS

To configure the Per-Session QoS feature, you should understand the following concepts:

Benefits of Per-Session QoS

The ability to apply QoS features on a per-session basis helps the Internet service provider (ISP) to adhere to the Service Level Agreement (SLA) established for handling traffic. Applying QoS on a per-session basis provides a higher degree of granularity for managing traffic on the network.

Policy Maps and QoS Features

A policy map specifies the QoS feature to be applied to network traffic. Examples of QoS features that can be specified in a policy map include traffic classification, shaping, queueing, and policing, among others. Each QoS feature is configured using the appropriate QoS commands.

Policy maps (including hierarchical policy maps) are created using the Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC).

Hierarchical Policy Maps

Policy maps can be configured in a hierarchical structure. That is, policy maps can be configured in levels subordinate to one another. The policy map at the highest level is referred toas the “parent” policy map. A subordinate policy map is referred to as a “child” policy map.

A typical hierarchical policy map structure consists of a parent policy map and one child policy map. Configure the child policy map first; then configure the parent policy map. Both types of policy maps are configured in the same manner.

The parent policy map typically contains one class—the class called class-default. The child policy map can contain multiple classes.

Note Before configuring the policy map, create the traffic classes and specify the match criteria used to classify traffic. To create traffic classes and specify match criteria, use the MQC.

The following restrictions apply to hierarchical policy maps:

- Specify CBWFQ in the child policy map *only*. CBWFQ cannot be specified in the parent policy map.
- Traffic shaping can be specified in *either* the parent policy map *or* the child policy map.

  However, for this feature, you *must* specify traffic shaping in the parent policy map. Specifying traffic shaping in the child policy map is optional.

Note The restrictions related to policy maps and the Cisco 7600 series router are different from those listed above. For more information about the restrictions specific to the Cisco 7600 series router, see the "Restrictions for Per-Session QoS (Cisco 7600 Series Routers)" section on page 3.

Per-Session QoS and Virtual Templates

As mentioned earlier, you can configure the Per-Session QoS feature using a virtual template.

Note Using virtual templates to configure the Per-Session QoS feature applies to PPP sessions only.

A virtual template is a logical interface that is configured with generic configuration information for a specific purpose or with configuration information common to specific users, plus router-dependent information. The template takes the form of a list of Cisco IOS interface commands that are applied to virtual access interfaces, as needed.

A virtual template is configured (defined) on an interface. When a session is enabled (that is, when a packet arrives at the interface), the virtual template inherits the QoS features specified in the policy map for use during the session.

First, you configure the policy map (using the MQC) and then associate the policy map with the virtual template. For more information about policy maps and the MQC, see the "Applying QoS Features Using the MQC" module.

After configuring the policy maps (as many as needed) and associating the policy maps with the virtual template on the interface, you may want to verify the configuration. To verify the per-session QoS configuration, use the **show policy-map session** [**uid** *uid-number*] command. This command allows you to see whether the policy maps are configured the way that you intended.

Per-Session QoS and the ISG Framework

QoS features can be applied on a per-session basis using the ISG framework in a number of ways, including the following:

- Enabling the QoS feature when it is triggered by specific events configured in the ISG policy map (for instance, at the start of a session or at a predetermined expiration interval).
- Using the Change of Authorization (CoA).
- Using the Transparent Auto Logon (TAL).
- Downloading the service profile at the time of authentication.

This feature module documents the procedure for applying per-session QoS when it is triggered at the start of a session (the first method listed above). For information about the other methods listed, see the "ISG RADIUS Interface" chapter of the *Cisco IOS ISG RADIUS CoA Interface Guide*, Release 12.2SB.

How to Configure Per-Session QoS

The tasks for configuring per-session QoS vary according to the configuration method that you are using. You can choose to configure the feature either using a RADIUS server or using the ISG framework.

Choose one of the following:

- To configure the feature using a RADIUS server, see the "Configuring Per-Session QoS Using a RADIUS Server" section on page 7.
- To configure per-session QoS using the ISG framework, see the "Configuring Per-Session QoS Using the ISG Framework" section on page 10.

Note For information about configuring the feature using a virtual template, see the "Per-Session QoS and Virtual Templates" section on page 6.

Configuring Per-Session QoS Using a RADIUS Server

This section contains the following tasks:

Adding the Cisco QoS AV Pairs to the Service Profile on the RADIUS Server

To configure per-session QoS on the RADIUS server, you must add two Cisco QoS AV pairs to the service profile on the RADIUS server. To add the Cisco QoS AV pairs to the service profile, complete the following steps on the RADIUS server.

Cisco AV Pairs and VSAs

Cisco AV pairs are part of vendor-specific attributes (VSAs) that allow a policy map to be applied to the router. Cisco AV pairs are a combination of an attribute and a value. The purpose of the Cisco VSA (attribute 26) is to communicate vendor-specific information between the router and the RADIUS server. The Cisco VSA encapsulates vendor-specific attributes that allow vendors such as Cisco to support their own extended attributes.

For this configuration, one of two Cisco AV pairs can be used (formatted as shown below):

- lcp:interface-config=service-policy output/input <policy name>

  This Cisco AV pair is considered a "legacy" AV pair. It is of earlier origin but is still an available choice.

- ip:sub-qos-policy-in/out=<policy name>

  This Cisco AV pair takes advantage of more recent technology and is therecommended choice. This Cisco AV pair is the one shown in the configuration task and example.

The Cisco AV pair is added to the service profile on the RADIUS server. Each entry establishes an attribute that the user can access.

In a user file, the data to the left of the equal sign (=) is an attribute defined in the dictionary file, and the data to the right of the equal sign is the configuration data.

The Cisco AV pair identifies the policy map that was used to configure the specific QoS features. When the router requests the policy map name (specified in the Cisco AV pair), the policy map is pulled to the router from the RADIUS server when the session is established. The Cisco AV pair applies the appropriate policy map (and, therefore, the QoS feature) directly to the router from the RADIUS server.

Prerequisites

Before adding the Cisco QoS AV pairs to the service profile, you must create traffic classes and configure policy maps used to enable the QoS feature you want to use. To create traffic classes and policy maps, use the MQC. For more information about the MQC, see the "Applying QoS Features Using the MQC" module.

SUMMARY STEPS

1. **ip:sub-qos-policy-in/out=<policy name>**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`ip:sub-qos-policy-in/out=<policy name>` **Example:** `cisco-avpair="ip:sub-qos-policy-in=res_ingress"` `cisco-avpair="ip:sub-qos-policy-out=res_hsi_voip_parent1"`	Enter the Cisco QoS AV pair for policy maps on the RADIUS server in the service profile. When the router requests the service definition from the RADIUS server, the information in the service profile is used. • Add the Cisco QoS AV pairs to the service profile.

Defining an ISG Policy Map to Start the QoS Service on the RADIUS Server

Next, you need to define the ISG policy map to start the QoS service at the start of the session when the service profile is defined on the RADIUS server. To perform this task, complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map type control** *policy-map-name*
4. **class type control always event session-start**
5. *action-number* **service-policy type service name** *service-name*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map type control** *policy-map-name* **Example:** `Router(config)# policy-map type control TEST`	Creates or modifies a policy map that can be attached toone or more interfaces to specify a service policy. • Enter the **type** and **control** keywords and the name of the policy map. **Note** Using the **control** keyword enters control policy-map configuration mode.
Step 4	**class type control always event session-start** **Example:** `Router(config-control-policymap)# class type control always event session-start`	Specifies a control class (or event) for which actions may be configured in policy map. Enters control policy-map class control configuration mode.
Step 5	*action-number* **service-policy type service name** *service-name* **Example:** `Router(config-control-policymap-class-control)# 1 service-policy type service name QoS_Service`	Applies the specified service at the start of the session. • Enter the action number, the **name** keyword, and the name of the service.
Step 6	**end** **Example:** `Router(config-control-policymap-class-control)# end`	(Optional) Returns to privileged EXEC mode.

Reviewing Session Statistics and Verifying the Policy Map Configuration

The last task is to review the output of the **show subscriber session** command and/or the output of the **show policy-map session** command. These two show commands allow you to review the statistics of the session and verify the policy map configuration.

SUMMARY STEPS

1. **enable**
2. **show subscriber session uid** *uid-number*
3. **show policy-map session uid** *uid-number*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`show subscriber session uid` *uid-number* **Example:** `Router# show subscriber session uid 401`	Displays information about subscriber sessions on an ISG by the unique ID. • Enter the **uid** keyword and unique identifier.
Step 3	`show policy-map session uid` *uid-number* **Example:** `Router# show policy-map session uid 401`	Displays the information about the session identified by the unique ID. • Enter the **uid** keyword and unique identifier.
Step 4	`exit` **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

What to Do Next

Proceed to the "Configuration Examples for Per-Session QoS" section on page 14.

Configuring Per-Session QoS Using the ISG Framework

This section contains the following tasks:

Configuring a Local Service Profile

The first task is to configure and define a local service profile for use with the policy map. To configure a local service profile for use with the policy map, complete the following steps.

Prerequisites

Before configuring the local service profile, you must create traffic classes and configure policy maps used to enable the QoS feature that you want to use. To create traffic classes and policy maps, use the MQC. For more information about the MQC, see the"Applying QoS Features Using the MQC" module.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **policy-map type service** *policy-map-name*
4. **service-policy input** *policy-map-name*
5. **service-policy output** *policy-map-name*
6. **exit**
7. **aaa authorization subscriber-service default local group** *name*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map type service** *policy-map-name* **Example:** `Router(config)# policy-map type service QoS_Service`	Creates or modifies a policy map that can be attached toone or more interfaces to specify a service policy. • Enter the **type** and **service** keywords and the name of the policy map. **Note** Using the **service** keyword enters service policy-map configuration mode.
Step 4	**service-policy input** *policy-map-name* **Example:** `Router(config-service-policymap)# service-policy input res_ingress`	Attaches the specified policy map to the input interface or input VC. • Enter the name of the policy map.
Step 5	**service-policy output** *policy-map-name* **Example:** `Router(config-service-policymap)# service-policy output res_hsi_voip_iptv_parent1`	Attaches the specified policy map to the output interface or output VC. • Enter the name of the policy map.
Step 6	**exit** **Example:** `Router(config-service-policymap)# exit`	Returns to global configuration mode.

	Command or Action	Purpose
Step 7	**aaa authorization subscriber-service default local group** *name* **Example:** `Router(config)# aaa authorization subscriber-service default local group group1`	Specifies one or more authentication, authorization, and accounting (AAA) authorization methods for ISG to use in providing subscriber service. • Enter the **default** keyword, the **local** keyword, the **group** keyword, and the group name. **Note** The **local** keyword must be entered after the **default** keyword.
Step 8	**end** **Example:** `Router(config)# end`	(Optional) Returns to privileged EXEC mode.

Defining an ISG Policy Map to Start the QoS Service

Next, you need to define the ISG policy map to initiate the QoS service at the start of the session. To perform this task, complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map type control** *policy-map-name*
4. **class type control always event session-start**
5. *action-number* **service-policy type service name** *service-name*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map type control** *policy-map-name* **Example:** `Router(config)# policy-map type control TEST`	Creates or modifies a policy map that can be attached toone or more interfaces to specify a service policy. • Enter the **type** and **control** keywords and the name of the policy map. **Note** Using the **control** keyword enters control policy-map configuration mode.

	Command or Action	Purpose
Step 4	**class type control always event session-start** **Example:** `Router(config-control-policymap)# class type control always event session-start`	Specifies a control class (or event) for which actions may be configured in an ISG control policy. Enters control policy-map class control configuration mode.
Step 5	*action-number* **service-policy type service name** *service-name* **Example:** `Router(config-control-policymap-class-control)# 1 service-policy type service name QoS_Service`	Activates an ISG service. • Enter the action number, the **name** keyword, and the name of the service.
Step 6	**end** **Example:** `Router(config-control-policymap-class-control)# end`	(Optional) Returns to privileged EXEC mode.

Starting the Session and Verifying the Policy Map Configuration

The last task is tostart a session by sending traffic in the ingress (incoming) direction and then reviewing the output of the **show subscriber session** command and/or the output of the **show policy-map session** command.

SUMMARY STEPS

1. **enable**
2. **show subscriber session uid** *uid-number*
3. **show policy-map session uid** *uid-number*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show subscriber session uid** *uid-number* **Example:** `Router# show subscriber session uid 401`	(Optional) Displays information about subscriber sessions on an ISG by the unique ID. • Enter the **uid** keyword and unique identifier.

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 3	**show policy-map session uid** *uid-number* **Example:** `Router# show policy-map session uid 401`	(Optional) Displays information about the session identified by the unique ID. • Enter the **uid** keyword and unique identifier.
Step 4	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

Configuration Examples for Per-Session QoS

This section contains the following examples:

Adding the Cisco QoS AV Pairs to the Service Profile on the RADIUS Server: Example

The following is an example of a service profile in which the Cisco QoS AV pairs have been added. Cisco AV pairs are needed only if you are configuring the Per-Session QoS feature using a RADIUS server.

```
cisco-avpair = "ip:sub-qos-policy-in=res_ingress"
cisco-avpair = "ip:sub-qos-policy-out=res_hsi_voip_iptv_parent1”
```

Configuring a Local Service Profile: Example

The following is an example of a local service profile configuration. Configuring a local service profile is needed only if you are configuring the Per-Session QoS feature using the ISG framework.

```
Router> enable
Router# configure terminal
Router(config)# policy-map type service QoS_Service
Router(config-service-policymap)# service-policy input res_ingress
Router(config-service-policymap)# service-policy output res_hsi_voip_iptv_parent1
Router(config-service-policymap)# exit
Router(config)# aaa authorization subscriber-service default local group group1
Router(config)# end
```

Defining an ISG Policy Map to Start the QoS Service: Example

The following is an example an ISG policy map configured to initiate the QoS service at the start of a session.

```
Router> enable
Router# configure terminal
Router(config)# policy-map type control TEST
Router(config-control-policymap)# class type control always event session-start
Router(config-control-policymap-class-control)# 1 service-policy type service name
QoS_Service
Router(config-control-policymap-class-control)# end
```

Verifying the Per-Session QoS Configuration: Examples

The following is an example of the output of the **show subscriber session** command.

```
Router# show subscriber session uid 2

Unique Session ID: 2
Identifier:
SIP subscriber access type(s): IP
Current SIP options: Req Fwding/Req Fwded
Session Up-time: 00:00:20, Last Changed: 00:00:20

Policy information:
  Authentication status: unauthen
  Active services associated with session:
    name "QoS_Service", applied before account logon
  Rules, actions and conditions executed:
    subscriber rule-map TEST
      condition always event session-start
        1 service-policy type service name QoS_Service

Session inbound features:
 Feature: QoS Policy Map
  Input Policy Map: res_ingress

Session outbound features:
 Feature: QoS Policy Map
  Output Policy Map: res_hsi_voip_iptv_parent1

Configuration sources associated with this session:
Service: QoS_Service, Active Time = 00:00:22
Interface: GigabitEthernet3/1/3.100, Active Time = 00:00:22
```

The following is an example of the output of the **show policy-map session** command.

```
Router# show policy-map session uid 2

 SSS session identifier 2 -

  Service-policy input: res_ingress

  Counters last updated 00:00:00 ago
```

```
   Class-map: voip (match-all)
     126126 packets, 9585576 bytes
     30 second offered rate 1114000 bps, drop rate 1114000 bps
     Match: ip precedence 5
     police:
         cir 9000 bps, bc 1500 bytes
       conformed 40 packets, 3040 bytes; actions:
         transmit
       exceeded 126086 packets, 9582536 bytes; actions:
         drop
       conformed 0 bps, exceed 1114000 bps
     QoS Set
       cos 5
         Packets marked 126126

   Class-map: class-default (match-any)

     262772 packets, 133488176 bytes
     30 second offered rate 15550000 bps, drop rate 15502000 bps
     Match: any
     police:
         cir 2000000 bps, bc 62500 bytes
       conformed 784 packets, 398272 bytes; actions:
         transmit
       exceeded 261988 packets, 133089904 bytes; actions:
         drop
       conformed 44000 bps, exceed 15502000 bps
     QoS Set
       cos 1
         Packets marked 262772
SSS session identifier 2 -

 Service-policy output: res_hsi_voip_iptv_parent1

   Class-map: class-default (match-any)
     0 packets, 0 bytes
     30 second offered rate 0 bps, drop rate 0 bps
     Match: any
     Queueing
     queue limit 2000 packets
     (queue depth/total drops/no-buffer drops) 0/0/0
     (pkts output/bytes output) 0/0
     bandwidth remaining ratio 5
     bandwidth remaining 0%
     shape (average) cir 8000000, bc 32000, be 32000
     target shape rate 8000000

     Service-policy : hsi_voip_iptv

       queue stats for all priority classes:

         priority level 1

         queue limit 2 packets
         (queue depth/total drops/no-buffer drops) 0/0/0
         (pkts output/bytes output) 0/0

       queue stats for all priority classes:

         priority level 2
         queue limit 1048 packets
         (queue depth/total drops/no-buffer drops) 0/0/0
         (pkts output/bytes output) 0/0
```

```
Class-map: voip (match-all)
  0 packets, 0 bytes
  30 second offered rate 0 bps, drop rate 0 bps
  Match: ip precedence 5
  Priority: Strict, b/w exceed drops: 0

  Priority Level: 1
  police:
      cir 9000 bps, bc 1500 bytes
    conformed 0 packets, 0 bytes; actions:
      transmit

    exceeded 0 packets, 0 bytes; actions:
      drop
    conformed 0 bps, exceed 0 bps
  QoS Set
    cos 5
      Packets marked 0

Class-map: iptv (match-all)
  0 packets, 0 bytes
  30 second offered rate 0 bps, drop rate 0 bps
  Match: ip precedence 6
  Priority: Strict, b/w exceed drops: 0

  Priority Level: 2
  police:
      cir 4193000 bps, bc 131031 bytes
    conformed 0 packets, 0 bytes; actions:
      transmit
    exceeded 0 packets, 0 bytes; actions:
      drop
    conformed 0 bps, exceed 0 bps
  QoS Set

    cos 4
      Packets marked 0

Class-map: class-default (match-any)
  0 packets, 0 bytes
  30 second offered rate 0 bps, drop rate 0 bps
  Match: any

  queue limit 949 packets
  (queue depth/total drops/no-buffer drops) 0/0/0
  (pkts output/bytes output) 0/0
  QoS Set
    cos 1
      Packets marked 0
```

Additional References

The following sections provide references related to the Per-Session QoS feature.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
QoS features such as traffic classification and traffic policing	"Quality of Service Overview" module
Class maps, policy maps, hierarchical policy maps, and MQC	"Applying QoS Features Using the MQC" module
RADIUS servers and AAA	"Configuring Authentication" module
RADIUS accounting	"Configuring Accounting" module
ISG policies and session maintenance	"Configuring ISG Policies for Session Maintenance" module
Classification, policing, and marking on Layer 2 Tunneling Protocol (L2TP) access concentrator (LAC)	"QoS: Classification, Policing, and Marking on LAC" module
LLQ, traffic shaping, CBWFQ, and WRED support on a 7600 series router	"IP Subscriber Awareness over Ethernet" module

Standards

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

MIBs

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

RFCs

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

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

This feature uses no new or modified commands.

Feature Information for Per-Session QoS

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Per-Session QoS

Feature Name	Releases	Feature Information
Per-Session QoS	12.2(28)SB 12.2(33)SRC	The Per-Session QoS feature provides the ability to apply quality of service (QoS) features (such as traffic classification, shaping, queueing, and policing) on a per-session basis. In Release 12.2(28)SB, this feature was introduced on the Cisco 7200 series router. In Release 12.2(33)SRC, support was added for the Cisco 7600 series router.

Glossary

L2TP—Layer 2 Tunneling Protocol. An IETF standards track protocol defined in RFC 2661 that provides tunneling of PPP. Based upon the best features of L2F and PPTP, L2TP provides an industry-wide interoperable method of implementing a virtual private dialup network (VPDN).

LAC—L2TP access concentrator. A node that acts as one side of an L2TP tunnel endpoint and that is a peer to the L2TP network server (LNS). The LAC sits between an LNS and a remote system and forwards packets to and from each. Packets sent from the LAC to the LNS require tunneling with the L2TP protocol. The connection from the LAC to the remote system is either local or a PPP link.

LNS—L2TP network server. A node that acts as one side of an L2TP tunnel endpoint and that is a peer to the L2TP access concentrator (LAC). The LNS is the logical termination point of a PPP session that is being tunneled from the remote system by the LAC.

PPP—Point-to-Point Protocol. A protocol that provides router-to-router and host-to-network connections over synchronous and asynchronous circuits. PPP is designed to work with several network layer protocols, such as IP, Internetwork Packet Exchange (IPX), and AppleTalk Remote Access (ARA).

PPPoA—Point-to-Point Protocol over ATM. A feature that allows a PPP session to be initiated on a simple bridging ATM connected client. PPPoA provides the ability to connect a network of hosts over a simple bridging access device to a remote access concentrator or aggregation concentrator.

PPPoE—Point-to-Point Protocol over Ethernet. A feature that allows a PPP session to be initiated on a simple bridging Ethernet connected client. PPPoE provides the ability to connect a network of hosts over a simple bridging access device to a remote access concentrator or aggregation concentrator.

PTA—PPP Termination and Aggregation. A network architecture that indicates that after a PPP session is terminated, the network traffic is aggregated. For an ISP, the aggregated traffic either remains in the ISP network or routes to the Internet. For a wholesale provider, the aggregated IP traffic will be forwarded to different destinations or domains depending on the service selected.

QoS—quality of service. A measure of performance for a transmission system that reflects its transmission quality and service availability.

SLA—Service Level Agreement. A contract between wholesale service providers and retail service providers.

SSS—Subscriber Service Switch. A switch that provides flexibility on where and how many subscribers are connected to available services and how those services are defined. The primary focus of SSS is to direct PPP from one point to another using a Layer 2 subscriber policy. The policy will manage tunneling of PPP in a policy-based bridging fashion.

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

Configuring IP to ATM Class of Service

IP to ATM Class of Service Overview

This chapter provides a high-level overview of IP to ATM Class of Service (CoS), a feature suite that maps QoS characteristics between IP and ATM.

For information on how to configure IP to ATM CoS, see the "Configuring IP to ATM Class of Service" module.

About IP to ATM CoS

The IP to ATM CoS feature implements a solution for coarse-grained mapping of QoS characteristics between IP and ATM, using Cisco Enhanced ATM port adapters (PA-A3) on Cisco 7200 and Cisco 7500 series routers. (This category of coarse-grained QoS is often referred to as CoS). The resulting feature makes it possible to support differential services in network service provider environments.

IP to ATM CoS is designed to provide a true working solution to class-based services, without the investment of new ATM network infrastructures. Now networks can offer different service classes (sometimes termed *differential service classes*) across the entire WAN, not just the routed portion. Mission-critical applications can be given exceptional service during periods of high network usage and congestion. In addition, noncritical traffic can be restricted in its network usage, which ensures greater QoS for more important traffic and user types.

The IP to ATM CoS feature is supported on Cisco 2600, Cisco 3600, Cisco 7200, and Cisco 7500 series routers equipped with the following hardware:

- Cisco 2600 and Cisco 3600 series: ATM OC-3, T1 IMA, or E1 IMA port adapter
- Cisco 7200 series:
  - NPE-200 or higher (NPE-300 recommended for per-virtual circuit (VC) class-based weighted fair queueing (CBWFQ)
  - One of the following Enhanced ATM port adapters (PA-A3): T3, E3, DS3, or OC-3
- Cisco 7500 series:
  - VIP2-50
  - One of the following Enhanced ATM port adapters (PA-A3): T3, E3, DS3, or OC-3

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

IP to ATM CoS supports configuration of the following features:

- Single ATM VCs
- VC bundles
- Per-VC Low Latency Queueing (LLQ), WFQ, and CBWFQ

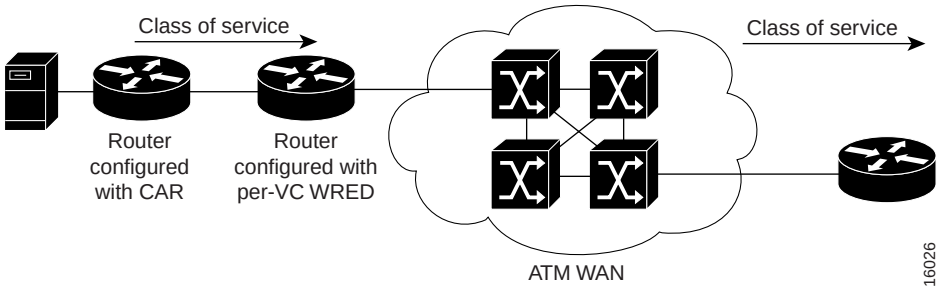
Single ATM VC Support

IP to ATM CoS support for a single ATM VC allows network managers to use existing features, such as committed access rate (CAR) or policy-based routing (PBR), to classify and mark different IP traffic by modifying the IP Precedence field in the IP version 4 (IPv4) packet header. Subsequently, Weighted Random Early Detection (WRED) or distributed WRED (DWRED) can be configured on a per-VC basis so that the IP traffic is subject to different drop probabilities (and therefore priorities) as IP traffic coming into a router competes for bandwidth on a particular VC.

Enhanced ATM port adapters (PA-A3) provide the ability to shape traffic on each VC according to the ATM service category and traffic parameters employed. When you use the IP to ATM CoS feature, congestion is managed entirely at the IP layer by WRED running on the routers at the edge of the ATM network.

Figure 1 illustrates the IP to ATM CoS support for a single ATM VC.

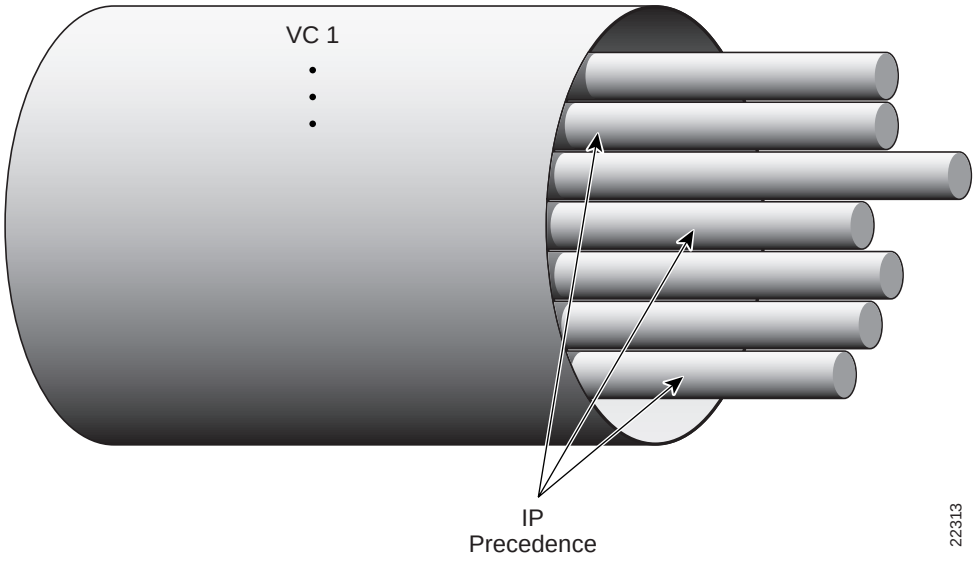
Figure 1 Single ATM Circuit Class



VC Bundle Support and Bundle Management

ATM VC bundle management allows you to configure multiple VCs that have different QoS characteristics between any pair of ATM-connected routers. As shown in Figure 2, these VCs are grouped in a bundle and are referred to as bundle members.

Figure 2 ATM VC Bundle

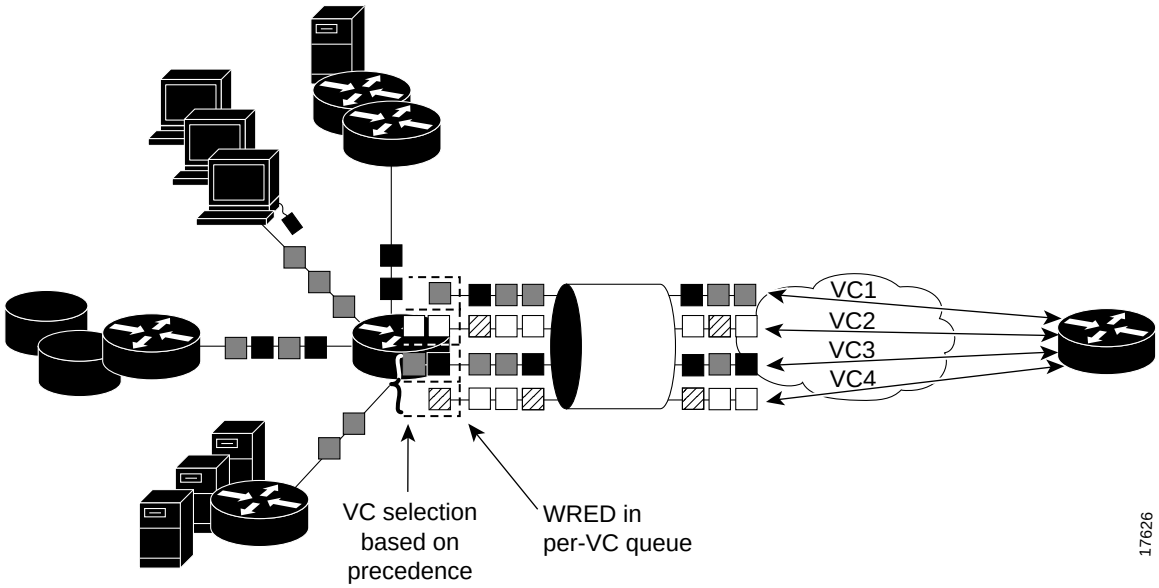


ATM VC bundle management allows you to define an ATM VC bundle and add VCs to it. Each VC of a bundle has its own ATM traffic class and ATM traffic parameters. You can apply attributes and characteristics to discrete VC bundle members or you can apply them collectively at the bundle level.

Using VC bundles, you can create differentiated service by flexibly distributing IP precedence levels over the different VC bundle members. You can map a single precedence level or a range of levels to each discrete VC in the bundle, thereby enabling individual VCs in the bundle to carry packets marked with different precedence levels. You can use WRED (or DWRED) to further differentiate service across traffic that has different IP precedences but that uses the same VC in a bundle.

To determine which VC in the bundle to use to forward a packet to its destination, the ATM VC bundle management software matches precedence levels between packets and VCs (see Figure 3). IP traffic is sent to the next hop address for the bundle because all VCs in a bundle share the same destination, but the VC used to carry a packet depends on the value set for that packet in the IP Precedence bits of the type of service (ToS) byte of its header. The ATM VC bundle management software matches the IP precedence of the packet to the IP Precedence value or range of values assigned to a VC, sending the packet out on the appropriate VC. Moreover, the ATM VC bundle management feature allows you to configure how traffic will be redirected when the VC the packet was matched to goes down. Figure 3 illustrates how the ATM VC bundle management software determines which permanent virtual circuit (PVC) bundle member to use to carry a packet and how WRED (or DWRED) is used to differentiate traffic on the same VC.

Figure 3 ATM VC Bundle PVC Selection for Packet Transfer



The support of multiple parallel ATM VCs allows you to create stronger service differentiation at the IP layer. For instance, you might want to provide IP traffic belonging to real-time CoS (such as Voice over IP traffic) on an ATM VC with strict constraints (constant bit rate (CBR) or variable bit rate real-time (VBR-rt), for example), while transporting traffic other than real-time traffic over a more elastic ATM available bit rate (ABR) PVC. Using a configuration such as this would allow you to fully utilize your network capacity. You could also elect to transport best-effort IP traffic over an unspecified bit rate (UBR) PVC—UBR is effectively the ATM version of best-effort service.

Per-VC LLQ, WFQ and CBWFQ Support

The IP to ATM CoS feature allows you to apply a policy map to a VC to specify a service policy for that VC so that all traffic sent on that VC is categorized according to the classes and their match criteria defined by the service policy. In other words, IP to ATM CoS takes the functionality defined for standard LLQ, WFQ, and CBWFQ and makes it available for application and use at the discrete VC level.

For conceptual information on LLQ, WFQ, and CBWFQ, see the "Congestion Management Overview" module.

IP to ATM CoS allows you to configure a single, standalone VC or individual VCs belonging to a bundle. You also can configure collectively all VCs belonging to a bundle. However, for per-VC LLQ, WFQ and CBWFQ, you can configure individual VCs only. That is, you can configure a standalone VC or a VC that belongs to a bundle, but you cannot use per-VC LLQ, WFQ and CBWFQ to configure a bundle of VCs collectively.

Per-VC LLQ, WFQ and CBWFQ allows you to differentiate the use of individual VCs within a bundle. For instance, you can apply one service policy to one VC belonging to a VC bundle and apply a different service policy to another VC belonging to the same bundle. You can also apply the same policy map to multiple VCs—whether standalone or bundle members—but each VC can have only one service policy. To concatenate service policies, you must create a third policy map and include in it all the classes that you want to use from policy maps you would have concatenated.

The following is a summary of how you configure a VC to use CBWFQ:

- You define traffic classes to specify the classification policy (class maps). This process determines how many types of packets are to be differentiated from one another.
- You configure policy maps containing classes that specify the policy for each traffic class.
- You attach a policy map to a VC that uses IP to ATM CoS to specify the service policy for the VC.

To apply flow-based WFQ on a per-VC basis, you configure WFQ in the predefined CBWFQ default class, which is called class-default, but you do not ascribe bandwidth to the default class. How to configure the default class to specify flow-based fair queueing is explained in the "Configuring IP to ATM Class of Service" module.

Why Use IP to ATM CoS?

Internet service classes can be identified and sorted within the router network. But as traffic traverses the wide-area ATM fabric, the relative ATM class definitions are not equivalent, and a traffic type may be treated differently in the ATM switching fabric than in the router network; mission-critical applications or data could be dropped during times of network congestion.

The IP to ATM CoS feature uses the Cisco Enhanced ATM port adapter (PA-A3) on Cisco 7500 and Cisco 7200 series routers to provide the ability to map IP CoS and ATM QoS, extending the capability previously available only for IP networks; differentiated services are preserved through the ATM network.

Benefits

Here are some benefits of using IP to ATM CoS:

- Ensures effective differential classes over IP and traditional ATM networks. For instance, the VC bundle management feature provides for differentiated QoS by allowing for the coexistence of multiple VCs with different QoS characteristics from the same source to the same destination.
- Uses existing ATM infrastructures.
- Implements solutions for coarse-grained mapping of QoS characteristics called CoS between IP and ATM.
- Employs a high-performance design benefiting from distributed processing on the Cisco 7500 series routers and Versatile Interface Processor (VIP).
- Uses the Cisco Enhanced ATM port adapter (PA-A3), which supports traffic shaping and has rich ATM Service Category support. This port adapter (PA) is supported on the Cisco 7500+VIP and Cisco 7200 series routers.
- Provides per-VC queueing on the PA, per-VC back pressure, and per-VC WRED VIP queueing, which bring stability to a network by ensuring that system packets such as Border Gateway Protocol (BGP) and Intermediate System-to-Intermediate System (IS-IS) are never dropped.
- Provides flexible management of the VC bundle on PVC failure.
- Provides CBWFQ functionality at the VC level.

IP to ATM CoS Features

IP to ATM CoS includes the following features:

- Per-VC queueing infrastructure. This feature enables queues to be maintained on a per-VC basis. Packets are queued and dequeued based on the back pressure from the PA. Use of a queue per VC prevents one or more congested VCs from affecting the traffic flow on other VCs that are not congested.
- Per-VC WRED (or DWRED). This feature applies the WRED algorithm independently to each per-VC queue. The WRED parameters are configurable on a per-VC basis so that congestion management can be configured as appropriate for each VC.
- Per-VC WRED (or DWRED) statistics. This feature maintains per-flow and per-VC statistics based on IP precedence.
- Per-VC LLQ, WFQ and CBWFQ. This feature allows you to apply CBWFQ functionality—normally applicable at the interface or subinterface levels only—to an individual VC configured for IP to ATM CoS. You can use this feature to apply either CBWFQ or flow-based WFQ on a per-VC basis.
- Per-VC traffic policing. This feature allows you to police traffic within a traffic policy, per-VC.

Congestion Avoidance

For each VC that is created on the Enhanced ATM port adapter (PA-A3), the PA allocates some of the buffers from its buffer pool to that VC in order to create a queue for that VC.

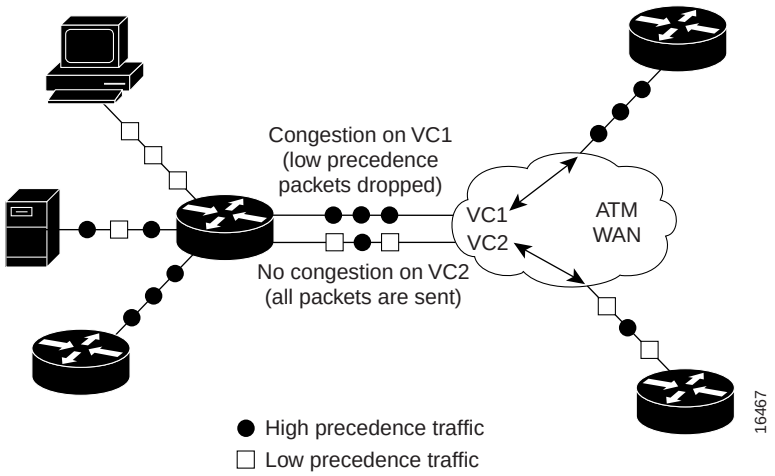
The use of per-VC queues ensures that a direct relationship exists between the outgoing ATM VC and the IP packets to be forwarded on that queue. This mechanism establishes a packet queue for each outgoing ATM VC. In this manner, should an ATM VC become congested, only the packet queue associated with that VC will begin to fill. If the queue overfills, then all other queues remain unaffected. Such a mechanism ensures that an individual VC cannot consume all of the resources of the router should only one of its outgoing VCs be congested or underprovisioned.

Queues for buffering more packets for a particular VC are created in the Layer 3 processor system and are mapped one-to-one to the per-VC queues on the PA. When the PA per-VC queues become congested, they signal back pressure to the Layer 3 processor; the Layer 3 processor can then continue to buffer packets for that VC in the corresponding Layer 3 queue. Furthermore, because the Layer 3 queues are accessible by the Layer 3 processor, a user can run flexible software scheduling algorithms on those queues.

When you transport data over ATM fabrics, it is essential that decisions to discard data (because of insufficient network resources or congestion) be made at the packet level. To do otherwise would be to send incomplete data packets into the ATM fabric, causing the packets to be discarded by either the ATM switched fabric (if it is equipped with early packet discard) or at the remote end where the packet will be reassembled and found to be incomplete.

To initiate effective congestion management techniques, IP to ATM CoS uses per-VC WRED (or DWRED). Per-VC WRED (or DWRED) selectively places TCP sessions in slow start mode to ensure higher aggregate throughput under congestion. Figure 4 shows low priority packets being dropped on VC1 because VC1 is congested. In this example, VC2 is not congested and all packets, regardless of priority, are sent.

Figure 4 Traffic Congestion with IP to ATM CoS and Per-VC WRED



Running the WRED algorithm independently on each per-VC queue provides differentiated QoS to traffic of different IP Precedence values.

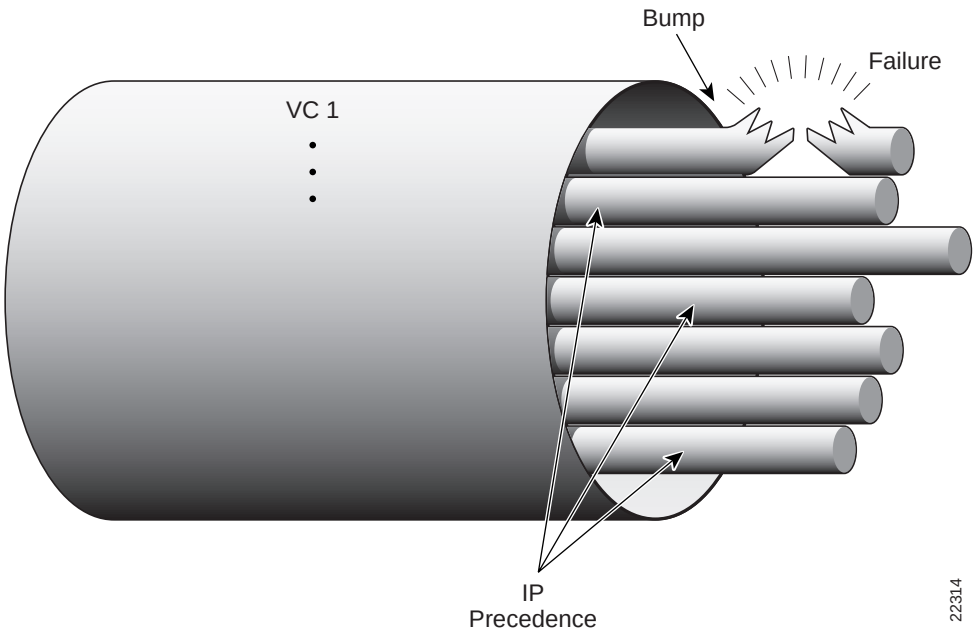
Bumping and ATM VC Bundles

The ATM VC bundle is designed to behave as a single routing link to the destination router while managing the integrity of its group of circuits. The integrity of each circuit is maintained through individual monitoring. Should a circuit fail, appropriate action is taken, in the form of circuit bumping or bundle disabling.

VC integrity is maintained through ATM Operation, Administration, and Maintenance (OAM) polling mechanisms. These mechanisms will determine whether a VC is unavailable or severely congested. Should an individual circuit become unavailable, then the device consults a preset series of rules to determine the course of action to take next. These rules are defined by the Internet service provider (ISP) through configuration parameters.

Figure 5 conceptualizes a failed VC bundle member whose failure calls into effect the configured bumping rules.

Figure 5 VC Bundle Member Circuit Failure Enacting Bumping Rules



In the event of failure, the router responds with one of two methods. The first method dynamically assigns the traffic bound on the failed VC to an alternative VC, which is termed *circuit bumping*. Bumped traffic is then shared on an existing in-service VC. Traffic typically would be bumped from a higher class to a lower one, although it need not be. For example, should the premium, or first class, data circuit become unavailable, then all premium users would share the second class or general circuit. Preference would then be given to the premium traffic within this shared circuit.

The second method is to declare all circuits of the bundle to be down. In effect, the device is declaring the routed bundle inactive and asking the routing layer to search for an alternate.

The determination of whether to bump or whether to declare the bundle inactive is predefined by the network provider when administering the network configuration.

Restrictions

The following restrictions apply for IP to ATM CoS:

- IP to ATM CoS supports only PVCs:
  - For PVC connections, it supports multipoint and point-to-point subinterfaces.
  - For PVC encapsulations, it supports only ATM adaptation layer (AAL5), Subnetwork Access Protocol (SNAP), and multiplex device (mux) interfaces.
- IP to ATM CoS does not allow point-to-multipoint VCs in the bundle. All VCs share the same source and destination (target) addresses.
- IP to ATM CoS does not work with the ATM Interface Processor (AIP) and the ATM port adapter (PA-A1).

Configuring IP to ATM Class of Service

This chapter describes the tasks for configuring the IP to ATM Class of Service (CoS), a feature suite that maps QoS characteristics between IP and ATM.

For complete conceptual information, see the "IP to ATM Class of Service Overview" module.

For a complete description of the IP to ATM CoS commands in this chapter, see the *Cisco IOS Quality of Service Solutions Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

IP to ATM CoS on a Single ATM VC Configuration Task List

To configure IP to ATM CoS for a single ATM virtual circuit (VC), perform the tasks described in the following sections. The tasks in the first two sections are required; the tasks in the remaining sections are optional.

- Defining the WRED Parameter Group (Required)
- Configuring the WRED Parameter Group (Required)
- Displaying the WRED Parameters (Optional)
- Displaying the Queueing Statistics (Optional)

The IP to ATM CoS feature requires ATM permanent virtual circuit (PVC) management.

See the end of this chapter for the section "Single ATM VC with WRED Group and IP Precedence Example."

Defining the WRED Parameter Group

To define the Weighted Random Early Detection (WRED) parameter group, use the following command in global configuration mode:

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Command	Purpose
`Router(config)# random-detect-group group-name`	Defines the WRED or VIP-distributed WRED (DWRED) parameter group.

Configuring the WRED Parameter Group

To configure the exponential weight factor for the average queue size calculation for a WRED parameter group or to configure a WRED parameter group for a particular IP precedence, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# random-detect-group group-name`	Specifies the WRED or DWRED parameter group.
Step 2	`Router(config)# exponential-weighting-constant exponent`	Configures the exponential weight factor for the average queue size calculation for the specified WRED or DWRED parameter group.
	or	or
	`Router(config)# precedence precedence min-threshold max-threshold mark-probability-denominator`	Configures the specified WRED or DWRED parameter group for a particular IP precedence.

Displaying the WRED Parameters

To display the configured WRED parameters, use the following command in privileged EXEC mode:

Command	Purpose
`Router# show queueing random-detect [interface atm_subinterface [vc [[vpi/] vci]]]`	Displays the parameters of every VC with WRED or DWRED enabled on the specified ATM subinterface.

Displaying the Queueing Statistics

To display the queueing statistics of an interface, use the following command in privileged EXEC mode:

Command	Purpose
`Router# show queueing interface interface-number [vc [[vpi/] vci]]`	Displays the queueing statistics of a specific VC on an interface.

IP to ATM CoS on an ATM Bundle Configuration Task List

To configure IP to ATM CoS on an ATM bundle, perform the tasks in the following sections. The first four sections are required; the remaining sections are optional.

- Creating a VC Bundle (Required)
- Applying Bundle-Level Parameters (Required)
  - Configuring Bundle-Level Parameters
  - Configuring VC Class Parameters to Apply to a Bundle
  - Attaching a Class to a Bundle
- Committing a VC to a Bundle (Required)
- Applying Parameters to Individual VCs (Required)
  - Configuring a VC Bundle Member Directly
  - Configuring VC Class Parameters to Apply to a VC Bundle Member
  - Applying a VC Class to a Discrete VC Bundle Member
- Configuring a VC Not to Accept Bumped Traffic (Optional)
- Monitoring and Maintaining VC Bundles and Their VC Members (Optional)

The IP to ATM CoS feature requires ATM PVC management.

See the end of this chapter for the section "VC Bundle Configuration Using a VC Class Example."

Creating a VC Bundle

To create a bundle and enter bundle configuration mode in which you can assign attributes and parameters to the bundle and all of its member VCs, use the following command in subinterface configuration mode:

Command	Purpose
`Router(config-subif)# bundle` *`bundle-name`*	Creates the specified bundle and enters bundle configuration mode.

Applying Bundle-Level Parameters

Bundle-level parameters can be applied either by assigning VC classes or by directly applying them to the bundle.

Parameters applied through a VC class assigned to the bundle are superseded by those applied at the bundle level. Bundle-level parameters are superseded by parameters applied to an individual VC.

Configuring Bundle-Level Parameters

Configuring bundle-level parameters is optional if a class is attached to the bundle to configure it.

To configure parameters that apply to the bundle and all of its members, use the following commands in bundle configuration mode, as needed:

Command	Purpose
`Router(config-atm-bundle)# protocol protocol {protocol-address | inarp} [[no] broadcast]`	Configures a static map or enables Inverse Address Resolution Protocol (Inverse ARP) or Inverse ARP broadcasts for the bundle.
`Router(config-atm-bundle)# encapsulation aal-encap`	Configures the ATM adaptation layer (AAL) and encapsulation type for the bundle.
`Router(config-atm-bundle)# inarp minutes`	Configures the Inverse ARP time period for all VC bundle members.
`Router(config-atm-bundle)# broadcast`	Enables broadcast forwarding for all VC bundle members.
`Router(config-atm-bundle)# oam retry up-count down-count retry frequency`	Configures the VC bundle parameters related to operation, administration, and maintenance (OAM) management.
`Router(config-atm-bundle)# oam-bundle [manage] [frequency]`	Enables end-to-end F5 OAM loopback cell generation and OAM management for all VCs in the bundle.

Configuring VC Class Parameters to Apply to a Bundle

Use of a VC class allows you to configure a bundle applying multiple attributes to it at once because you apply the class itself to the bundle. Use of a class allows you to generalize a parameter across all VCs, after which (for some parameters) you can modify that parameter for individual VCs. (See the section "Applying Parameters to Individual VCs" for more information.)

To configure a VC class to contain commands that configure all VC members of a bundle when the class is applied to that bundle, use the following command in vc-class configuration mode. To enter vc-class configuration mode, use the **vc-class atm** command.

Command	Purpose
`Router(config-vc-class)# oam-bundle [manage] [frequency]`	Enables end-to-end F5 OAM loopback cell generation and OAM management for all VCs in the bundle.

In addition to this command, you can add the following commands to a VC class to be used to configure a bundle: **broadcast**, **encapsulation**, **inarp**, **oam retry**, and **protocol**. For information about these commands, see the *Cisco IOS Wide-Area Networking Command Reference.*

Attaching a Class to a Bundle

To attach a preconfigured VC class containing bundle-level configuration commands to a bundle, use the following command in bundle configuration mode:

Command	Purpose
`Router(config-atm-bundle)# class-bundle vc-class-name`	Configures a bundle with the bundle-level commands contained in the specified VC class.

Parameters set through bundle-level commands contained in the VC class are applied to the bundle and all of its VC members. Bundle-level parameters applied through commands configured directly on the bundle supersede those applied through a VC class.

Note that some bundle-level parameters applied through a VC class or directly to the bundle can be superseded by commands that you directly apply to individual VCs in bundle-vc configuration mode.

Committing a VC to a Bundle

To add a VC to an existing bundle and enter bundle-vc configuration mode, use the following command in bundle configuration mode:

Command	Purpose
`Router(config-atm-bundle)# pvc-bundle pvc-name [vpi/] [vci]`	Adds the specified VC to the bundle and enters bundle-vc configuration mode in order to configure the specified VC bundle member.

For information on how to first create the bundle and configure it, see the sections "Creating a VC Bundle" and "Applying Bundle-Level Parameters" earlier in this chapter.

Applying Parameters to Individual VCs

Parameters can be applied to individual VCs either by using VC classes or by directly applying them to the bundle members.

Parameters applied to an individual VC supersede bundle-level parameters. Parameters applied directly to a VC take precedence over the same parameters applied within a class to the VC at the bundle-vc configuration level.

Configuring a VC Bundle Member Directly

Configuring VC bundle members directly is optional if a VC class is attached to the bundle member.

To configure an individual VC bundle member directly, use the following commands in bundle-vc configuration mode, as needed:

Command	Purpose
`Router(config-if-atm-member)# ubr output-pcr [input-pcr]`	Configures the VC for unspecified bit rate (UBR) QoS and specifies the output peak cell rate (PCR) for it.
`Router(config-if-atm-member)# ubr+ output-pcr output-mcr [input-pcr] [input-mcr]`	Configures the VC for UBR QoS and specifies the output PCR and output minimum guaranteed cell rate for it.
`Router(config-if-atm-member)# vbr-nrt output-pcr output-scr output-mbs [input-pcr] [input-scr] [input-mbs]`	Configures the VC for variable bit rate nonreal-time (VBR-nrt) QoS and specifies the output PCR, output sustainable cell rate, and output maximum burst cell size for it.
`Router(config-if-atm-member)# precedence [other | range]`	Configures the precedence levels for the VC.

Command	Purpose
`Router(config-if-atm-member)# bump {implicit | explicit precedence-level | traffic}`	Configures the bumping rules for the VC.
`Router(config-if-atm-member)# protect {group | vc}`	Configures the VC to belong to the protected group of the bundle or to be an individually protected VC bundle member.

Parameters set directly for a VC at the bundle-vc configuration level take precedence over values for these parameters set for the VC at any other level, including application of a VC class at the bundle-vc configuration level.

Configuring VC Class Parameters to Apply to a VC Bundle Member

To configure a VC class to contain commands that configure a specific VC member of a bundle when the class is applied to it, use the following commands in vc-class configuration mode, as needed. To enter vc-class configuration mode, use the **vc-class atm** command in global configuration mode.

Command	Purpose
`Router(config-vc-class)# bump {implicit | explicit precedence-level | traffic}`	Specifies the bumping rules for the VC member to which the class is applied. These rules determine to which VC in the bundle traffic is directed when the carrier VC bundle member goes down.
`Router(config-vc-class)# precedence precedence min-threshold max-threshold mark-probability-denominator`	Defines precedence levels for the VC member to which the class is applied.
`Router(config-vc-class)# protect {group | vc}`	Configures the VC as a member of the protected group of the bundle or as an individually protected VC.

You can also add the following commands to a VC class to be used to configure a VC bundle member: **ubr**, **ubr+**, and **vbr-nrt**.

Use of a VC class allows you to configure a VC bundle member with multiple attributes at once because you can apply the class to the VC.

Note When a VC is a member of a VC bundle, the following commands cannot be used in vc-class mode to configure the VC: **encapsulation**, **protocol**, **inarp**, and **broadcast**. These commands are useful only at the bundle level, not the bundle member level.

To configure the way bumping is handled for individual VCs within a bundle, use the **bump** command in the bundle-vc configuration mode. For more information about the bumping rules, see the "IP to ATM Class of Service Overview" module in this book.

Configuration for an individual VC overrides the collective configuration applied to all VC bundle members through application of a VC class to the bundle.

Applying a VC Class to a Discrete VC Bundle Member

To attach a preconfigured VC class containing bundle-level configuration commands to a bundle member, use the following command in bundle-vc configuration mode:

Command	Purpose
`Router(config-if-atm-member)# class-vc vc-class -name`	Assigns a VC class to a VC bundle member.

Parameters that configure a VC that are contained in a VC class assigned to that VC are superseded by parameters that are directly configured for the VC through discrete commands entered in bundle-vc configuration mode.

Configuring a VC Not to Accept Bumped Traffic

To configure an individual VC bundle member not to accept traffic that otherwise might be directed to it if the original VC carrying the traffic goes down, use the following command in bundle-vc configuration mode:

Command	Purpose
`Router(config-if-atm-member)# no bump traffic`	Configures the VC not to accept any bumped traffic that would otherwise be redirected to it.

Monitoring and Maintaining VC Bundles and Their VC Members

To gather information on bundles so as to monitor them or to troubleshoot problems that pertain to their configuration or use, use the following commands in privileged EXEC mode, as needed:

Command	Purpose
`Router# show atm bundle bundle-name`	Displays the bundle attributes assigned to each bundle VC member and the current working status of the VC members.
`Router# show atm bundle bundle-name statistics [detail]`	Displays statistics or detailed statistics on the specified bundle.
`Router# show atm map`	Displays a list of all configured ATM static maps to remote hosts on an ATM network and on ATM bundle maps.
`Router# debug atm bundle errors`	Displays information on bundle errors.
`Router# debug atm bundle events`	Displays a record of bundle events.

Per-VC WFQ and CBWFQ Configuration Task List

To configure IP to ATM CoS for per-VC WFQ and CBWFQ, perform the tasks described in the following sections. The tasks in the first two sections are required; the tasks in the remaining sections are optional.

- Configuring Class-Based Weighted Fair Queueing (Required)
- Attaching a Service Policy and Enabling CBWFQ for a VC (Required)
- Configuring a VC to Use Flow-Based WFQ (Optional)
- Monitoring per-VC WFQ and CBWFQ (Optional)
- Enabling Logging of Error Messages to the Console (Optional)

The IP to ATM CoS feature requires ATM PVC management.

See the end of this chapter for the sections "Per-VC WFQ and CBWFQ on a Standalone VC Example" and "Per-VC WFQ and CBWFQ on Bundle-Member VCs Example."

Configuring Class-Based Weighted Fair Queueing

Before configuring CBWFQ for a VC, you must perform the following tasks using standard CBWFQ commands:

- Create one or more classes to be used to classify traffic sent across the VC
- Define a policy map containing the classes to be used as the service policy

Note You can configure class policies for as many classes as are defined on the router, up to the maximum of 64. However, the total amount of bandwidth allocated for all classes included in a policy map to be attached to a VC must not exceed 75 percent of the available bandwidth of the VC. The remaining 25 percent of available bandwidth is used for encapsulation, such as the ATM cell overhead (also referred to as ATM cell tax), routing and best-effort traffic, and other functions that assume overhead. For more information on bandwidth allocation, see the "Congestion Management Overview" module.

For information on how to configure CBWFQ and perform the tasks mentioned, see the chapter "Configuring Weighted Fair Queueing" in this book.

Attaching a Service Policy and Enabling CBWFQ for a VC

Because CBWFQ gives you minimum bandwidth guarantee, you can only apply CBWFQ to VCs having these classes of service: available bit rate (ABR) and variable bit rate (VBR). You cannot apply per-VC WFQ and CBWFQ to UBR and unspecified bit rate plus (UBR+) VCs because both of these service classes are best-effort classes that do not guarantee minimum bandwidth. When CBWFQ is enabled for a VC, all classes configured as part of the service policy are installed in the fair queueing system.

To attach a policy map to a standalone VC to be used as its service policy and to enable CBWFQ on that VC, use the following command in VC submode:

Command	Purpose
`Router(config-if-atm-vc)# service-policy output policy-map`	Enables CBWFQ and attaches the specified service policy map to the VC being created or modified.

To attach a policy map to an individual VC bundle member to be used as its service policy and to enable CBWFQ on that VC, use the following command in bundle-vc configuration mode:

Command	Purpose
`Router(config-if-atm-member)# service-policy output policy-map`	Enables CBWFQ and attaches the specified service policy map to the VC being created or modified.

Note The **service-policy output** and **random-detect-group** commands are mutually exclusive; you cannot apply a WRED group to a VC for which you have enabled CBWFQ through application of a service policy. Moreover, before you can configure one command, you must disable the other if it is configured.

Configuring a VC to Use Flow-Based WFQ

In addition to configuring CBWFQ at the VC level, the IP to ATM CoS feature allows you to configure flow-based WFQ at the VC level. Because flow-based WFQ gives you best-effort class of service—that is, it does not guarantee minimum bandwidth—you can configure per-VC WFQ for all types of CoS VCs: ABR, VBR, UBR, and UBR+.

Per-VC WFQ uses the class-default class. Therefore, to configure per-VC WFQ, you must first create a policy map and configure the class-default class. (You need not create the class-default class, which is predefined, but you must configure it.) For per-VC WFQ, the class-default class must be configured with the **fair-queue** policy-map class configuration command.

In addition to configuring the **fair-queue** policy-map class configuration command, you can configure the default class with either the **queue-limit** command or the **random-detect** command, but not both. Moreover, if you want the default class to use flow-based WFQ, you cannot configure the default class with the **bandwidth** policy-map class configuration command—to do so would disqualify the default class as flow-based WFQ, and therefore limit application of the service policy containing the class to ABR and VBR VCs.

To create a policy map and configure the class-default class, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# policy-map policy-map`	Specifies the name of the policy map to be created or modified.
Step 2	`Router(config-pmap)# class class-default default-class-name`	Specifies the default class so that you can configure or modify its policy.
Step 3	`Router(config-pmap-c)# fair-queue number-of-dynamic-queues`	Specifies the number of dynamic queues to be reserved for use by flow-based WFQ running on the default class.
Step 4	`Router(config-pmap-c)# queue-limit number-of-packets` or `Router(config-pmap-c)# random-detect`	Specifies the maximum number of packets that can be queued for the class. Enables WRED. The class policy will drop packets using WRED instead of tail drop.

For more information about creating a policy map and configuring the class-default class, see the "Configuring Weighted Fair Queueing" module in this book.

By default—that is, even if you do not configure the class-default class with the **fair-queue** policy-map class configuration command and you do not configure it with the **bandwidth** policy-map class configuration command—the default class is defined as flow-based WFQ.

Note that you can include other classes in the same policy map as the one that contains the flow-based WFQ class. Packets not otherwise matched are selected by the default class-default class match criteria.

To attach the policy map containing the class-default class to a standalone VC so that it becomes the service policy enabling WFQ for that VC, use the following command in VC submode:

Command	Purpose
`Router(config-if-atm-vc)# service-policy output policy-map`	Enables WFQ for the VC by attaching the specified policy map containing the class-default class to the VC being created or modified.

To attach the policy map containing the class-default class to an individual VC bundle member so that the policy map becomes the service policy enabling WFQ for that VC, use the following command in bundle-vc configuration mode:

Command	Purpose
`Router(config-if-atm-member)# service-policy output policy-map`	Enables WFQ for the VC bundle member by attaching the specified policy map containing the class-default class to the VC bundle member.

Monitoring per-VC WFQ and CBWFQ

To monitor per-VC WFQ and CBWFQ in your network, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router# show queue interface-name interface-number [vc [vpi/] vci]]`	Displays the contents of packets inside a queue for a particular interface or VC.
`Router# show queueing interface interface-number [vc [[vpi/] vci]]`	Displays the queueing statistics of a specific VC on an interface.

Enabling Logging of Error Messages to the Console

When you configure a VC in order to create or modify it, the router performs the task in interrupt mode. For this reason, the router cannot issue printf statements to inform you of error conditions, if errors occur. Rather, the router logs all error messages to the console. To accommodate these circumstances, you should enable logging of error messages to the console.

To enable logging of error messages to the console, use the following command in global configuration mode:

Command	Purpose
`Router(config)# logging console level`	Limits messages logged to the console based on severity.

For information on the **logging console** command, see the *Cisco IOS Configuration Fundamentals Command Reference*.

IP to ATM CoS Configuration Examples

The following sections provide IP to ATM CoS configuration examples:

- Single ATM VC with WRED Group and IP Precedence Example
- VC Bundle Configuration Using a VC Class Example
- Per-VC WFQ and CBWFQ on a Standalone VC Example
- Per-VC WFQ and CBWFQ on Bundle-Member VCs Example

For information on how to configure IP to ATM CoS, see the sections "IP to ATM CoS on a Single ATM VC Configuration Task List" and "IP to ATM CoS on an ATM Bundle Configuration Task List" in this chapter.

Single ATM VC with WRED Group and IP Precedence Example

The following example creates a PVC on an ATM interface and applies the WRED parameter group called sanjose to that PVC. Next, the IP Precedence values are configured for the WRED parameter group sanjose.

```
interface ATM1/1/0.46 multipoint
 ip address 200.126.186.2 255.255.255.0
 no ip mroute-cache
 shutdown
pvc cisco 46
 encapsulation aal5nlpid
 random-detect attach sanjose
!
random-detect-group sanjose
 precedence 0  200   1000 10
 precedence 1  300   1000 10
 precedence 2  400   1000 10
 precedence 3  500   1000 10
 precedence 4  600   1000 10
 precedence 5  700   1000 10
 precedence 6  800   1000 10
 precedence 7  900   1000 10
```

VC Bundle Configuration Using a VC Class Example

This example configures VC bundle management on a router that uses Intermediate System-to-Intermediate System (IS-IS) as its IP routing protocol.

Bundle-Class Class

At the outset, this configuration defines a VC class called bundle-class that includes commands that set VC parameters. When the class bundle-class is applied at the bundle level, these parameters are applied to all VCs that belong to the bundle. Note that any commands applied directly to an individual VC of a bundle in bundle-vc mode take precedence over commands applied globally at the bundle level. Taking into account hierarchy precedence rules, VCs belonging to any bundle to which the class bundle-class is applied will be characterized by these parameters: aal5snap encapsulation, broadcast on, use of Inverse Address Resolution Protocol (ARP) to resolve IP addresses, and operation, administration, and maintenance (OAM) enabled.

```
router isis
 net 49.0000.0000.0000.1111.00

vc-class atm bundle-class
 encapsulation aal5snap
 broadcast
 protocol ip inarp
 oam-bundle manage 3
 oam retry 4 3 10
```

The following sections of the configuration define VC classes that contain commands specifying parameters that can be applied to individual VCs in a bundle by assigning the class to that VC.

Control-Class Class

When the class called control-class is applied to a VC, the VC carries traffic whose IP Precedence level is 7. When the VC to which this class is assigned goes down, it takes the bundle down with it because this class makes the VC a protected one. The QoS type of a VC using this class is vbr-nrt.

```
vc-class atm control-class
 precedence 7
 protect vc
 vbr-nrt 10000 5000 32
```

Premium-Class Class

When the class called premium-class is applied to a VC, the VC carries traffic whose IP Precedence levels are 6 and 5. The VC does not allow other traffic to be bumped onto it. When the VC to which this class is applied goes down, its bumped traffic will be redirected to a VC whose IP Precedence level is 7. This class makes a VC a member of the protected group of the bundle. When all members of a protected group go down, the bundle goes down. The QoS type of a VC using this class is vbr-nrt.

```
vc-class atm premium-class
 precedence 6-5
 no bump traffic
 protect group
 bump explicitly 7
 vbr-nrt 20000 10000 32
```

Priority-Class Class

When the class called priority-class is applied to a VC, the VC is configured to carry traffic with IP Precedence in the 4-2 range. The VC uses the implicit bumping rule, it allows traffic to be bumped, and it belongs to the protected group of the bundle. The QoS type of a VC using this class is ubr+.

```
vc-class atm priority-class
 precedence 4-2
 protect group
 ubr+ 10000 3000
```

Basic-Class Class

When the class called basic-class is applied to a VC, the VC is configured through the **precedence other** command to carry traffic with IP Precedence levels not specified in the profile. The VC using this class belongs to the protected group of the bundle. The QoS type of a VC using this class is ubr.

```
vc-class atm basic-class
```

```
precedence other
protect group
ubr 10000
```

The following sets of commands configure three bundles that the router subinterface uses to connect to three of its neighbors. These bundles are called new-york, san-francisco, and los-angeles. Bundle new-york has four VC members, bundle san-francisco has four VC members, and bundle los-angeles has three VC members.

new-york Bundle

The first part of this example specifies the IP address of the subinterface, the router protocol—the router uses IS-IS as an IP routing protocol—and it creates the first bundle called new-york and enters bundle configuration mode:

```
interface atm 1/0.1 multipoint
 ip address 10.0.0.1 255.255.255.0
 ip router isis
 bundle new-york
```

From within bundle configuration mode, the next portion of the configuration uses two protocol commands to enable IP and Open Systems Interconnect (OSI) traffic flows in the bundle. The OSI routing packets will use the highest precedence VC in the bundle. The OSI data packets, if any, will use the lowest precedence VC in the bundle. If configured, other protocols, such as IPX or AppleTalk, will always use the lowest precedence VC in the bundle.

As the indentation levels of the preceding and following commands suggest, subordinate to bundle new-york is a command that configures its protocol and a command that applies the class called bundle-class to it.

```
  protocol ip 1.1.1.2 broadcast
  protocol clns 49.0000.0000.2222.00 broadcast
  class-bundle bundle-class
```

The class called bundle-class, which is applied to the bundle new-york, includes a **protocol ip inarp** command. According to inheritance rules, **protocol ip**, configured at the bundle level, takes precedence over **protocol ip inarp** specified in the class bundle-class.

The next set of commands beginning with **pvc-bundle ny-control 207**, which are further subordinate, add four VCs (called ny-control, ny-premium, ny-priority, and ny-basic) to the bundle new-york. A particular class—that is, one of the classes predefined in this configuration example—is applied to each VC to configure it with parameters specified by commands included in the class.

As is the case for this configuration, to configure individual VCs belonging to a bundle, the router must be in bundle mode for the mother bundle. For each VC belonging to the bundle, the subordinate mode is pvc-mode for the specific VC.

The following commands configure the individual VCs for the bundle new-york:

```
  pvc-bundle ny-control 207
   class-vc control-class
  pvc-bundle ny-premium 206
   class-vc premium-class
  pvc-bundle ny-priority 204
   class-vc priority-class
  pvc-bundle ny-basic 201
   class-vc basic-class
```

san-francisco Bundle

The following set of commands create and configure a bundle called san-francisco. At the bundle configuration level, the configuration commands included in the class bundle-class are ascribed to the bundle san-francisco and to the individual VCs that belong to the bundle. Then, the **pvc-bundle** command is executed for each individual VC to add it to the bundle. After a VC is added and bundle-vc configuration mode is entered, a particular, preconfigured class is assigned to the VC. The configuration commands comprising that class are used to configure the VC. Rules of hierarchy apply at this point. Command parameters contained in the applied class are superseded by the same parameters applied at the bundle configuration level, which are superseded by the same parameters applied directly to a VC.

```
bundle san-francisco
 protocol clns 49.0000.0000.0000.333.00 broadcast
 inarp 1
 class-bundle bundle-class
 pvc-bundle sf-control 307
  class-vc control-class
 pvc-bundle sf-premium 306
  class-vc premium-class
 pvc-bundle sf-priority 304
  class-vc priority-class
 pvc-bundle sf-basic 301
  class-vc basic-class
```

los-angeles Bundle

The following set of commands create and configure a bundle called los-angeles. At the bundle configuration level, the configuration commands included in the class bundle-class are ascribed to the bundle los-angeles and to the individual VCs that belong to the bundle. Then, the **pvc-bundle** command is executed for each individual VC to add it to the bundle. After a VC is added and bundle-vc configuration mode is entered, precedence is set for the VC and the VC is either configured as a member of a protected group (protect group) or as an individually protected VC. A particular class is then assigned to each VC to further characterize it. Rules of hierarchy apply. Parameters of commands applied directly and discretely to a VC take precedence over the same parameters applied within a class to the VC at the bundle-vc configuration level, which take precedence over the same parameters applied to the entire bundle at the bundle configuration level.

```
bundle los-angeles
 protocol ip 1.1.1.4 broadcast
 protocol clns 49.0000.0000.4444.00 broadcast
 inarp 1
 class-bundle bundle-class
 pvc-bundle la-high 407
  precedence 7-5
  protect vc
  class-vc premium-class
 pvc-bundle la-mid 404
  precedence 4-2
  protect group
  class-vc priority-class
 pvc-bundle la-low 401
  precedence other
  protect group
  class-vc basic-class
```

Per-VC WFQ and CBWFQ on a Standalone VC Example

The following example creates two class maps and defines their match criteria. For the first map class, called class1, the numbered access control list (ACL) 101 is used as the match criterion. For the second map class called class2, the numbered ACL 102 is used as the match criterion.

Next, the example includes these classes in a policy map called policy1. For class1, the policy includes a minimum bandwidth allocation request of 500 kbps and maximum packet count limit of 30 for the queue reserved for the class. For class2, the policy specifies only the minimum bandwidth allocation request of 1000 kbps, so the default queue limit of 64 packets is assumed. Note that the sum of the bandwidth requests for the two classes comprising policy1 is 75 percent of the total amount of bandwidth (2000 kbps) for the PVC called cisco to which the policy map is attached.

The example attaches the policy map called policy1 to the PVC called cisco. Once the policy map policy1 is attached to PVC cisco, its classes constitute the CBWFQ service policy for that PVC. Packets sent on this PVC will be checked for matching criteria against ACLs 101 and 102 and classified accordingly.

Because the **class-default** command is not explicitly configured for this policy map, all traffic that does not meet the match criteria of the two classes comprising the service policy is handled by the predefined class-default class, which provides best-effort flow-based WFQ.

```
class-map class1
 match access-group 101

class-map class2
 match access-group 102

policy-map policy1
 class class1
  bandwidth 500
  queue-limit 30

 class class2
  bandwidth 1000

interface ATM1/1/0.46 multipoint
 ip address 200.126.186.2 255.255.255.0
 pvc cisco 46
  vbr-nrt 2000 2000
  encap aal5snap
  service policy output policy1
```

Per-VC WFQ and CBWFQ on Bundle-Member VCs Example

The following example shows a PVC bundle called san-francisco with members for which per-VC WFQ and CBWFQ are enabled and service policies configured. The example assumes that the classes included in the following policy maps have been defined and that the policy maps have been created: policy1, policy2, and policy4. For each PVC, the IP to ATM CoS **pvc-bundle** command is used to specify the PVC to which the specified policy map is to be attached.

Note that PVC 0/34 and 0/31 have the same policy map attached to them, policy2. Although you can assign the same policy map to multiple VCs, each VC can have only one policy map attached at an output PVC.

```
bundle san-francisco
 protocol ip 1.0.2.20 broadcast
 encapsulation aal5snap
```

```
pvc-bundle 0/35
 service policy output policy1
 vbr-nrt 5000 3000 500
 precedence 4-7
pvc-bundle 0/34
 service policy output policy2
 vbr-nrt 5000 3000 500
 precedence 2-3
pvc-bundle 0/33
 vbr-nrt 4000 3000 500
 precedence 2-3
 service policy output policy4
pvc-bundle 0/31
 service policy output policy2
```

Service Groups

First Published: November 2, 2009
Last Updated: November 2, 2009

The Service Group feature allows network administrators to create service groups, add members (such as service instances) to those service groups, and apply service policies (also known an policy maps) to those newly created groups. The service policies (policy maps) contain the aggregate features (such as traffic policing and queueing) to be applied to the groups in compliance with the Service-Level Agreement (SLA) negotiated between the service provider and the subscribers.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "Feature Information for Service Groups" section on page 13.

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

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Restrictions for Service Groups

For Cisco IOS Release12.2(33)SRE, the following restrictions apply:

- This feature is supported only on the Cisco IOS 7600 series router.
- Layer 3 subinterfaces are not supported.

Note For additional restrictions for the Cisco 7600 series routers and the line cards used on the router, see both the *Cisco 7600-ES20 Ethernet Line Card Configuration Guide* and the *Cisco 7600 Series Ethernet Services + Line Card Configuration Guide*.

Information About Service Groups

To configure the Service Groups feature, you should understand the following concepts.

- Service Instance Definition, page 2
- Benefits of Service Groups, page 2
- Service Groups, QoS Policy Maps, and Automatic Load-Balancing, page 2

Service Instance Definition

A service instance is a configuration object (container) that holds all management and control plane attributes and parameters that apply to that service instance on a per-port basis. Different service instances that correspond to the same EVC must share the same name. Service instances are associated with a global EVC object through their shared name.

For more information about service instances, see the *Cisco IOS Carrier Ethernet Configuration Guide*.

Benefits of Service Groups

This feature allows you to create service groups and apply aggregate features to those service groups. For Cisco IOS Release 12.2(33)SRE on a Cisco 7600 series router, a QoS policy map is the only feature that can be applied to service groups.

Service Groups, QoS Policy Maps, and Automatic Load-Balancing

For Cisco IOS Release 12.2(33)SRE on Cisco 7600 series router, only QoS service policies (policy maps) on service groups or group members are supported. A QoS policy map may be configured on service groups (or on individual service group members) on an interface or a port-channel. On a port-channel, the service group feature enables you to implement load-balancing by distributing the multiple service instances among the different member links.

When a member link goes down, automatic load-balancing is triggered on the port-channel. Load-balancing redistributes the EVCs to the remaining member links on the port-channel, while maintaining the original QoS policy maps. For example, consider that there are two QoS policy maps, QoS1 and QoS2, implemented on two service groups, SG1 and SG2, respectively. Service group SG1 is connected to the network through link M1; service group SG2 is connected to the network through link

M2. If the M2 link goes down, automatic load-balancing is initiated and all the EVCs of service group SG2 are redistributed to the M1 link. Even though the service group SG2 is now moved under the link M1, service group SG2 maintains its original QoS policy map, QoS2; when service groups are redistributed, the QoS policy maps are not affected.

Note It is possible to manually load-balance service groups and service instances across member links of a port-channel using a feature called "User-Network Interface (UNI) Link Aggregation Group (LAG) Advanced Load-Balancing". For more information, see the "Configuring Layer 2 Features" chapter in the *Cisco 7600 Series Ethernet Services + Line Card Configuration Guide*.

How to Configure Service Groups

This section contains the following tasks:

- Creating a Service Group, page 3 (Required)
- Adding or Deleting Service Group Members, page 4 (Deleting members is optional; adding members is required)
- Deleting a Service Group, page 6 (Optional)
- Verifying the Service Group Configuration, page 7 (Optional)

Creating a Service Group

To create and configure a service group, complete the following steps.

Prerequisites

In this procedure, you need to specify the name of a QoS policy to be attached to the service group. The QoS policy must already exist. To create the QoS policy, use the Modular Quality of Service Command-Line Interface (CLI) (MQC). For more information about the MQC, see the "Applying QoS Features Using the MQC" module.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **service-group** *service-group-identifier*
4. **description** *descriptive-text*
5. **service-policy** {**input** | **output**} *policy-map-name*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**service-group** *service-group-identifier* **Example:** `Router(config)# service-group 20`	Creates a service group and enters service-group configuration mode. • Enter the service group number. The number of service groups that can be created varies by router.
Step 4	**description** *descriptive-text* **Example:** `Router(config-service-group)# description subscriber account number 105AB1`	(Optional) Creates a description of the service group. • Enter a description (for example, additional information about the group) of the service group. Descriptions can be a maximum of 240 characters.
Step 5	**service-policy** {**input** | **output**} *policy-map-name* **Example:** `Router(config-service-group)# service-policy input policy1`	(Optional) Attaches a policy map to the service group, in either the ingress (input) or egress (output) direction. • Enter either the **input** or **output** keyword and the name of the previously created policy map.
Step 6	**end** **Example:** `Router(config-service-group)# end`	(Optional) Returns to privileged EXEC mode.

Adding or Deleting Service Group Members

To add members to a service group, or to delete members from a service group, complete the following steps.

Restrictions

The following restrictions apply to service group members:

- In Cisco IOS Release 12.2(33)SRE on the Cisco 7600 series router, a member can join only one service group at a time.
- On the Cisco 7600 series router, all members of a service group must reside on the same physical or port-channel interface.
- A member of a service group cannot be individually assigned to a load-balance link of a port-channel. The entire service group must be assigned to the load-balance link.
- A service group cannot have members that are assigned to multiple load-balance links on a port-channel.

- The Cisco 7600 series router does not allow service instances to join the same group from multiple interfaces. On the Cisco 7600 series router, group members must come from the same interface, as shown the sample configuration below:

```
interface GigabitEthernet 2/0/0
 service instance 1 ethernet
  group 32
  Service-policy output policy3
 service instance 2 ethernet
  group 32
 service instance 3 ethernet
  group 37
interface GigabitEthernet 2/0/1
 service instance 1 ethernet
  group 32  |<--Disallowed because this group has members in g2/0/0 already |
```

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*

   or

   interface port-channel *port-channel-number*
4. **service instance** *service-instance-number* **ethernet**
5. **group** *service-group-identifier*
6. **no group** *service-group-identifier*
7. **exit**
8. Repeat steps 4, 5, and 6 (as applicable) to add or delete any additional service group members.
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface GigabitEthernet 1/0/0`	Configures an interface and enters interface configuration mode. • Enter the interface type and the interface number.

	Command or Action	Purpose
	Or	
	interface port-channel *port-channel-number* **Example:** `Router(config)# interface port-channel 50`	(Optional) Configures a port-channel and enters interface configuration mode. • Enter the port-channel number.
Step 4	**service instance** *service-instance-number* **ethernet** **Example:** `Router(config-if)# service instance 200 ethernet`	Specifies the service-instance to be added or deleted from a service group. Enters service configuration mode. • Enter the service-instance number.
Step 5	**group** *service-group-identifier* **Example:** `Router(config-if-srv)# group 20`	Number of the service group to which the member specified in Step 4 will be added. • Enter the service group number.
Step 6	**no group** *service-group-identifier* **Example:** `Router(config-if-srv)# no group 30`	(Optional) Number of the service group from which the member specified in Step 4 will be deleted. • Enter the service group number.
Step 7	**exit** **Example:** `Router(config-if-srv)# exit`	(Optional) Returns to interface configuration mode.
Step 8	(Optional) Repeat Step 4, Step 5, or Step 6 (as applicable) for each group member (in this case, service instances) that you want to add or delete.	
Step 9	**end** **Example:** `Router(config-if-srv)# end`	(Optional) Returns to privileged EXEC mode.

Deleting a Service Group

To delete service group, complete the following steps. When you delete a service group, all members of the service group are automatically removed from the service group.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no service-group** *service-group-identifier*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**no service-group** *service-group-identifier* **Example:** `Router(config)# no service-group 20`	Removes a service group and removes all members from the service group. • Enter the service group number to be deleted.
Step 4	**end** **Example:** `Router(config)# end`	(Optional) Exits global configuration mode.

Verifying the Service Group Configuration

To verify, debug, and troubleshoot the service group configuration, use one or more of commands listed below.

SUMMARY STEPS

1. **enable**
2. **show running-config service-group**
3. **show service-group** {*service-group-identifier* | **all**}
4. **show service-group interface** *type number*
5. **show service-group stats**
6. **show service-group state**
7. **show service-group traffic-stats**
8. **show policy-map interface** *type number* **service group**
9. **show ethernet service instance** [**detail**]
10. **clear service-group traffic-stats**
11. **debug service-group** {**all** | **error** | **feature** | **group** | **interface** | **ipc** | **member** | **qos** | **stats**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show running-config service-group** **Example:** `Router# show running-config service-group`	(Optional) Displays the running service-group configuration.
Step 3	**show service-group** {*service-group-identifier* | **all**} **Example:** `Router# show service-group all`	(Optional) Displays service-group configuration information for one or all service groups.
Step 4	**show service-group interface** *type number* **Example:** `Router# show service-group interface gigabitethernet 3/1`	(Optional) Displays service-group membership information by interface. • Enter the interface type and number. The other keywords and arguments are optional.
Step 5	**show service-group stats** **Example:** `Router# show service-group stats`	(Optional) Displays service-group statistical information.
Step 6	**show service-group state** **Example:** `Router# show service-group state`	(Optional) Displays state information about service-groups.
Step 7	**show service-group traffic-stats** **Example:** `Router# show service-group traffic-stats`	(Optional) Displays the traffic statistics for all the members of a service group. The information displayed is the combined total of the traffic statistics for all members.
Step 8	**show policy-map interface** *type number* **service group** **Example:** `Router# show policy-map interface gigabitEthernet 9/5 service group`	(Optional) Displays the policy-map information for service groups that have members attached to the specified interface. • Enter the interface type and number.
Step 9	**show ethernet service instance** [**detail**] **Example:** `Router# show ethernet service instance detail`	(Optional) Displays information about the service instances. **Note** To display the service group number, use the **detail** keyword.

	Command or Action	Purpose
Step 10	`clear service-group traffic-stats` **Example:** `Router# clear service-group traffic-stats`	(Optional) Clears the traffic statistics for the service group. **Note** Clearing the traffic statistics for the service group does not clear the traffic statistics for the group members. To clear the traffic statistics for group members, use the **clear ethernet service instance** command. For more information about the **clear ethernet service instance** command, see the *Cisco IOS Carrier Ethernet Command Reference*.
Step 11	`debug service-group {all | error | feature | group | interface | ipc | member | qos | stats}` **Example:** `Router# debug service-group qos`	(Optional) Debugs service-group events and errors.

Configuration Examples for Service Groups

This section provides the following configuration examples:

Creating a Service Group: Example

In the following example, service group 20 has been created.

```
Router> enable
Router# configure terminal
Router(config)# service-group 20
Router(config-service-group)# description account number 105AB1
Router(config-service-group)# service-policy input policy1
Router(config-service-group)# end
```

Adding Members to a Service Group: Example

In the following example, service instance 200 will be added to service group 20.

```
Router> enable
Router# configure terminal
Router(config)# interface GigabitEthernet 1/0/0
Router(config-if)# service instance 200 ethernet
Router(config-if-srv)# group 20
Router(config-if-srv)# end
```

Deleting Members from a Service Group: Example

In the following example, service instance 300 will be deleted from service group 30 on a port-channel.

```
Router> enable
Router# configure terminal
Router(config)# interface port-channel 50
Router(config-if)# service instance ethernet 300
Router(config-if-srv)# no group 30
Router(config-if-srv)# end
```

Deleting a Service Group: Example

In the following example, service group 20 will be deleted.

```
Router> enable
Router# configure terminal
Router(config)# no service-group 20
Router(config)# end
```

Verifying the Service Group Configuration: Example

This section contains sample output of the **show policy-map interface service group** command. The **show policy-map interface service group** command displays the policy-map information for service groups that have members attached to an interface.

Note This command is one of several that you can use to verify the service-group configuration. For additional commands that can be used, see the "Verifying the Service Group Configuration" section on page 7.

In the following example, service group 1 is specified. Service group 1 contains two policy maps (service policies), policy1 and policy2. Traffic policing is enabled in the evc policy map. Traffic queueing is enabled in the isg policy map.

```
Router# show policy-map interface gigabitEthernet 9/5 service group 1

 GigabitEthernet9/5: Service Group 1

  Service-policy input: policy1

    Class-map: class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any
      police:
          cir 200000 bps, bc 6250 bytes
        conformed 0 packets, 0 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          drop
        conformed 0000 bps, exceed 0000 bps

  Service-policy output: policy2

  Counters last updated 00:00:34 ago
```

```
Class-map: class-default (match-any)
  0 packets, 0 bytes
  5 minute offered rate 0000 bps, drop rate 0000 bps
  Match: any
  Queueing
  queue limit 131072 packets
  (queue depth/total drops/no-buffer drops) 0/0/0
  (pkts output/bytes output) 0/0
  bandwidth remaining ratio 2
```

Additional References

The following sections provide references related to the Service Groups feature.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples.	*Cisco IOS Quality of Service Solutions Command Reference*
Debug commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples.	*Cisco IOS Debug Command Reference*
MQC, policy maps	"Applying QoS Features Using the MQC" module
Cisco IOS 7600 series routers	*Cisco 7600-ES20 Ethernet Line Card Configuration Guide* *Cisco 7600 Series Ethernet Services + Line Card Configuration Guide*.
Service instance configuration information and concepts	*Cisco IOS Carrier Ethernet Configuration Guide*
Service instance commands	*Cisco IOS Carrier Ethernet Command Reference*
Manually load-balancing service groups and service instances across member links of a port-channel	"Configuring Layer 2 Features" chapter of the *Cisco 7600 Series Ethernet Services + Line Card Configuration Guide*.

Standards

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

MIBs

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

RFCs

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

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Service Groups

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Service Groups

Feature Name	Releases	Feature Information
Service Groups	12.2(33)SRE	The Service Groups feature allows network administrators to create service groups, add members (such as service instances) to those service groups, and apply service policies (also known an policy maps) tothose newly created groups. In Release 12.2(33)SRE, this feature was introduced on the Cisco 7600 series router. The following commands were introduced or modified: **clear service-group traffic-stats**, **debug service-group**, **description (service group)**, **group (service group)**, **service-group**, **service instance ethernet**, **service-policy (service group)**, **show policy-map interface service group**, **show running-config service-group**, **show service-group**, **show service-group interface**, **show service-group state**, **show service-group stats**, **show service-group traffic-stats**.

Modular Quality of Service Command-Line Interface

Applying QoS Features Using the MQC

First Published: April 30, 2007
Last Updated: June 3, 2009

This module contains the concepts about applying QoS features using the Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC) and the tasks for configuring the MQC. The MQC allows you to define a traffic class, create a traffic policy (policy map), and attach the traffic policy to an interface. The traffic policy contains the QoS feature that will be applied to the traffic class.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Applying QoS Features Using the MQC" section on page 20.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Restrictions for Applying QoS Features Using the MQC

IPX Packets

The MQC does not support Internetwork Packet Exchange (IPX) packets.

Number of QoS Class Maps Supported

The number of QoS class maps supported in a single policy map varies by release, as follows:

- For Cisco IOS XE Release 2.1, and Cisco IOS XE Release 2.2, the MQC supports a maximum of 8 class maps in a single policy map.
- For Cisco IOS Release 12.4T, Cisco IOS Release 12.2 SR, and Cisco IOS XE Release 2.3, the MQC supports a maximum of 256 class maps in a single policy map.

Number of QoS Policy Maps Supported

The number of QoS class maps supported on a router varies by release, as follows:

- For Cisco IOS XE Release 2.1, and Cisco IOS XE Release 2.2, the MQC supports no more than 1000 policy maps in the incoming (ingress) direction, outgoing (egress) direction, or a combination of both on a router.
- For Cisco IOS Release 12.4T, Cisco IOS Release 12.2 SR, and Cisco IOS XE Release 2.3, the MQC supports no more than 4000 policy maps in the incoming (ingress) direction, outgoing (egress) direction, or a combination of both on a router.

QoS Policy Maps and Sessions

When sessions are created and QoS policy maps are attached in both the ingress and egress directions, only 2000 sessions are supported. Sessions exceeding this limit can still be created, but the QoS policy maps will not be applied to the session.

Information About Applying QoS Features Using the MQC

Before applying QoS features using the MQC, you should be familiar with the following concepts:

- The MQC Structure, page 3
- Elements of a Traffic Class, page 3
- Elements of a Traffic Policy, page 5
- Nested Traffic Classes, page 7
- Benefits of Applying QoS Features Using the MQC, page 7

The MQC Structure

The MQC structure allows you to define a traffic class, create a traffic policy, and attach the traffic policy to an interface.

The MQC structure consists of the following three high-level steps.

Step 1 Define a traffic class by using the **class-map** command. A traffic class is used to classify traffic.

Step 2 Create a traffic policy by using the **policy-map** command. (The terms *traffic policy* and *policy map* are often synonymous.) A traffic policy (policy map) contains a traffic class and one or more QoS features that will be applied to the traffic class. The QoS features in the traffic policy determine how to treat the classified traffic.

Step 3 Attach the traffic policy (policy map) to the interface by using the **service-policy** command.

Elements of a Traffic Class

A traffic class contains three major elements: a traffic class name, a series of **match** commands, and, if more than one **match** command is used in the traffic class, instructions on how to evaluate these **match** commands.

The **match** commands are used for classifying packets. Packets are checked to determine whether they meet the criteria specified in the **match** commands; if a packet meets the specified criteria, that packet is considered a member of the class. Packets that fail to meet the matching criteria are classified as members of the default traffic class.

Available match Commands

Table 1 lists some of the available **match** commands that can be used with the MQC. The available **match** commands vary by Cisco IOS release and platform. For more information about the commands and command syntax, see the command reference for the Cisco IOS release and platform that you are using.

Table 1 match Commands That Can Be Used with the MQC

Command	Purpose
match access-group	Configures the match criteria for a class map on the basis of the specified access control list (ACL).
match any	Configures the match criteria for a class map to be successful match criteria for all packets.
match class-map	Specifies the name of a traffic class to be used as a matching criterion (for nesting traffic classes [nested class maps] within one another).
match cos	Matches a packet based on a Layer 2 class of service (CoS) marking.
match destination-address mac	Uses the destination MAC address as a match criterion.
match discard-class	Matches packets of a certain discard class.

Table 1 match Commands That Can Be Used with the MQC (continued)

Command	Purpose
match [ip] dscp	Identifies a specific IP differentiated service code point (DSCP) value as a match criterion. Up to eight DSCP values can be included in one match statement.
match field	Configures the match criteria for a class map on the basis of the fields defined in the protocol header description files (PHDFs).
match fr-dlci	Specifies the Frame Relay data-link connection identifier (DLCI) number as a match criterion in a class map.
match input-interface	Configures a class map to use the specified input interface as a match criterion.
match ip rtp	Configures a class map to use the Real-Time Transport Protocol (RTP) port as the match criterion.
match mpls experimental	Configures a class map to use the specified value of the Multiprotocol Label Switching (MPLS) experimental (EXP) field as a match criterion.
match mpls experimental topmost	Matches the MPLS EXP value in the topmost label.
match not	Specifies the single match criterion value to use as an unsuccessful match criterion. **Note** The **match not** command, rather than identifying the specific match parameter to use as a match criterion, is used to specify a match criterion that prevents a packet from being classified as a member of the class. For instance, if the **match not qos-group 6** command is issued while you configure the traffic class, QoS group 6 becomes the only QoS group value that is not considered a successful match criterion. All other QoS group values would be successful match criteria.
match packet length	Specifies the Layer 3 packet length in the IP header as a match criterion in a class map.
match port-type	Matches traffic on the basis of the port type for a class map.
match [ip] precedence	Identifies IP precedence values as match criteria.
match protocol	Configures the match criteria for a class map on the basis of the specified protocol. **Note** There is a separate **match protocol** (NBAR) command used to configure Network-Based Application Recognition (NBAR) to match traffic by a protocol type known to NBAR.
match protocol citrix	Configures NBAR to match Citrix traffic.
match protocol fasttrack	Configures NBAR to match FastTrack peer-to-peer traffic.
match protocol gnutella	Configures NBAR to match Gnutella peer-to-peer traffic.
match protocol http	Configures NBAR to match Hypertext Transfer Protocol (HTTP) traffic by URL, host, Multipurpose Internet Mail Extension (MIME) type, or fields in HTTP packet headers.

Table 1 match Commands That Can Be Used with the MQC (continued)

Command	Purpose
match protocol rtp	Configures NBAR to match Real-Time Transport Protocol (RTP) traffic.
match qos-group	Identifies a specific QoS group value as a match criterion.
match source-address mac	Uses the source MAC address as a match criterion.
match start	Configures the match criteria for a class map on the basis of the datagram header (Layer 2) or the network header (Layer 3).
match tag	Specifies tag type as a match criterion.

Multiple match Commands in One Traffic Class

If the traffic class contains more than one **match** command, you need to specify how to evaluate the **match** commands. You specify this by using either the **match-any** or **match-all** keywords of the **class-map** command. Note the following points about the **match-any** and **match-all** keywords:

- If you specify the **match-any** keyword, the traffic being evaluated by the traffic class must match *one* of the specified criteria.
- If you specify the **match-all** keyword, the traffic being evaluated by the traffic class must match *all* of the specified criteria.
- If you do not specify either keyword, the traffic being evaluated by the traffic class must match *all* of the specified criteria (that is, the behavior of the **match-all** keyword is used).

Elements of a Traffic Policy

A traffic policy contains three elements: a traffic policy name, a traffic class (specified with the **class** command), and the command used to enable the QoS feature.

The traffic policy (policy map) applies the enabled QoS feature to the traffic class once you attach the policy map to the interface (by using the **service-policy** command).

Note 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 will be used.

Commands Used to Enable QoS Features

The commands used to enable QoS features vary by Cisco IOS release and platform. Table 2 lists some of the available commands and the QoS features that they enable. For complete command syntax, see the command reference for the Cisco IOS release and platform that you are using.

For more information about a specific QoS feature that you want to enable, see the appropriate module of the *Cisco IOS Quality of Service Solutions Configuration Guide*.

Table 2 Commands Used to Enable QoS Features

Command	Purpose
bandwidth	Enables Class-Based Weighted Fair Queuing (CBWFQ).
fair-queue	Specifies the number of queues to be reserved for a traffic class.

Table 2 Commands Used to Enable QoS Features (continued)

Command	Purpose
drop	Discards the packets in the specified traffic class.
identity policy	Creates an identity policy.
police	Configures traffic policing.
police (control-plane)	Configures traffic policing for traffic that is destined for the control plane.
police (EtherSwitch)	Defines a policer for classified traffic.
police (percent)	Configures traffic policing on the basis of a percentage of bandwidth available on an interface.
police (two rates)	Configures traffic policing using two rates, the committed information rate (CIR) and the peak information rate (PIR).
police rate pdp	Configures Packet Data Protocol (PDP) traffic policing using the police rate. **Note** This command is intended for use on the Gateway General Packet Radio Service (GPRS) Support Node (GGSN).
priority	Gives priority to a class of traffic belonging to a policy map.
queue-limit	Specifies or modifies the maximum number of packets the queue can hold for a class configured in a policy map.
random-detect	Enables Weighted Random Early Detection (WRED) or distributed WRED (DWRED).
random-detect discard-class	Configures the WRED parameters for a discard-class value for a class in a policy map.
random-detect discard-class-based	Configures WRED on the basis of the discard class value of a packet.
random-detect ecn	Enables explicit congestion notification (ECN).
random-detect exponential-weighting-constant	Configures the exponential weight factor for the average queue size calculation for the queue reserved for a class.
random-detect precedence	Configure the WRED parameters for a particular IP Precedence for a class policy in a policy map.
service-policy	Specifies the name of a traffic policy used as a matching criterion (for nesting traffic policies [hierarchical traffic policies] within one another).
set atm-clp	Sets the cell loss priority (CLP) bit when a policy map is configured.
set cos	Sets the Layer 2 class of service (CoS) value of an outgoing packet.
set discard-class	Marks a packet with a discard-class value.
set [**ip**] **dscp**	Marks a packet by setting the differentiated services code point (DSCP) value in the type of service (ToS) byte.
set fr-de	Changes the discard eligible (DE) bit setting in the address field of a Frame Relay frame to 1 for all traffic leaving an interface.

Table 2 Commands Used to Enable QoS Features (continued)

Command	Purpose
set mpls experimental	Designates the value to which the MPLS bits are set if the packets match the specified policy map.
set precedence	Sets the precedence value in the packet header.
set qos-group	Sets a QoS group identifier (ID) that can be used later to classify packets.
shape	Shapes traffic to the indicated bit rate according to the algorithm specified.
shape adaptive	Configures a Frame Relay interface or a point-to-point subinterface to estimate the available bandwidth by backward explicit congestion notification (BECN) integration while traffic shaping is enabled.
shape fecn-adapt	Configures a Frame Relay interface to reflect received forward explicit congestion notification (FECN) bits as backward explicit congestion notification (BECN) bits in Q.922 test response messages.

Nested Traffic Classes

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

Similarly, the MQC allows multiple traffic classes (nested traffic classes, which are also called nested class maps) to be configured as a single traffic class. This nesting can be achieved with the use of the **match class-map** command. The only method of combining match-any and match-all characteristics within a single traffic class is with the **match class-map** command.

For an example of a nested traffic class configuration, see the "Traffic Class as a Match Criterion (Nested Traffic Classes): Example" section on page 16.

Benefits of Applying QoS Features Using the MQC

The MQC structure allows you to create the traffic policy (policy map) once and then apply it to as many traffic classes as needed. You can also attach the traffic policies to as many interfaces as needed.

How to Apply QoS Features Using the MQC

To apply QoS features using the MQC, perform the following tasks.

- Creating a Traffic Class (required)
- Creating a Traffic Policy (required)
- Attaching a Traffic Policy to an Interface (required)
- Verifying the Traffic Class and Traffic Policy Information (optional)

Creating a Traffic Class

To create a traffic class, use the **class-map** command to specify the traffic class name. Then use one or more **match** commands to specify the appropriate match criteria. Packets matching the criteria that you specify are placed in the traffic class.

To create the traffic class, complete the following steps.

Note The **match cos** command is shown in Step 4. The **match cos** command is simply an example of one of the **match** commands that you can use. For information about the other available **match** commands, see Table 1 on page 3.

The match-all and match-any Keywords of the class-map Command

One of the commands used when you create a traffic class is the **class-map** command. The command syntax for the **class-map** command includes two keywords: **match-all** and **match-any**. The **match-all** and **match-any** keywords need to be specified only if more than one match criterion is configured in the traffic class. Note the following points about these keywords:

- The **match-all** keyword is used when *all* of the match criteria in the traffic class must be met in order for a packet to be placed in the specified traffic class.
- The **match-any** keyword is used when only *one* of the match criterion in the traffic class must be met in order for a packet to be placed in the specified traffic class.
- If neither the **match-all** keyword nor **match-any** keyword is specified, the traffic class will behave in a manner consistent with **match-all** keyword.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** [**match-all** | **match-any**] *class-map-name*
4. **match cos** *cos-number*
5. Enter additional **match** commands, if applicable; otherwise, continue with Step 6.
6. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**class-map** [**match-all** | **match-any**] *class-map-name* **Example:** `Router(config)# class-map match-any class1`	Creates a class to be used with a class map and enters class-map configuration mode. The class map is used for matching packets to the specified class. • Enter the class name. **Note** The **match-all** keyword specifies that all match criteria must be met. The **match-any** keyword specifies that one of the match criterion must be met. Use these keywords only if you will be specifying more than one **match** command.
Step 4	**match cos** *cos-number* **Example:** `Router(config-cmap)# match cos 2`	Matches a packet on the basis of a Layer 2 class of service (CoS) number. • Enter the CoS number. **Note** The **match cos** command is simply an example of one of the **match** commands you can use. For information about the other **match** commands that are available, see Table 1 on page 3.
Step 5	Enter additional match commands, if applicable; otherwise, continue with Step 6.	—
Step 6	**end** **Example:** `Router(config-cmap)# end`	(Optional) Exits class-map configuration mode and returns to privileged EXEC mode.

Creating a Traffic Policy

To create a traffic policy (or policy map) and enable one or more QoS features, perform the following steps.

Note The **bandwidth** command is shown in Step 5. The **bandwidth** command is simply an example of one of the commands that you can use in a policy map to enable a QoS feature (in this case, CBWFQ). For information about other available commands, see Table 2 on page 5.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** {*class-name* | **class-default**}
5. **bandwidth** {*bandwidth-kbps* | **percent** *percent*}
6. Enter the commands for any additional QoS feature that you want to enable, if applicable; otherwise, continue with Step 7.
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map-name* **Example:** `Router(config)# policy-map policy1`	Creates or specifies the name of the traffic policy and enters policy-map configuration mode. • Enter the policy map name.
Step 4	**class** {*class-name* | **class-default**} **Example:** `Router(config-pmap)# class class1`	Specifies the name of a traffic class and enters policy-map class configuration mode. • Enter the class name created in the "Creating a Traffic Class" section on page 8). **Note** This step associates the traffic class with the traffic policy.
Step 5	**bandwidth** {*bandwidth-kbps* | **percent** *percent*} **Example:** `Router(config-pmap-c)# bandwidth 3000`	(Optional) Specifies a minimum bandwidth guarantee to a traffic class in periods of congestion. A minimum bandwidth guarantee can be specified in kbps or by a percentage of the overall available bandwidth. **Note** The **bandwidth** command enables CBWFQ. The **bandwidth** command is simply an example of one of the commands that you can use in a policy map to enable a QoS feature. For information about the other commands available, see Table 2 on page 5.
Step 6	Enter the commands for any additional QoS feature that you want to enable, if applicable; otherwise, continue with Step 7.	—
Step 7	**end** **Example:** `Router(config-pmap-c)# end`	(Optional) Exits policy-map class configuration mode and returns to privileged EXEC mode.

Attaching a Traffic Policy to an Interface

The traffic policy (policy map) applies the enabled QoS feature to the traffic class once you attach the policy map to the interface (by using the **service-policy** command).

Depending on the platform and Cisco IOS release that you are using, a traffic policy can be attached to an ATM permanent virtual circuit (PVC) subinterface, a Frame Relay data-link connection identifier (DLCI), or another type of interface.

To attach a traffic policy to an interface, perform the following steps.

The input and output Keywords of the service-policy Command

The QoS feature configured in the traffic policy can be applied to packets entering the interface or to packets leaving the interface. Therefore, when you use the **service-policy** command, you need to specify the direction by using the **input** or **output** keyword.

For instance, the **service-policy output class1** command would apply the feature in the traffic policy to the interface. All packets leaving the interface are evaluated according to the criteria specified in the traffic policy named class1.

Restrictions

Multiple Traffic Policies

Multiple traffic policies on tunnel interfaces and physical interfaces are not supported if the interfaces are associated with each other. For instance, if a traffic policy is attached to a tunnel interface while another traffic policy is attached to a physical interface—with which the tunnel interface is associated—only the traffic policy on the tunnel interface works properly.

Bandwidth Allocated to Priority Traffic

The amount of bandwidth allocated to the priority traffic cannot exceed the amount of bandwidth available on the interface. If the traffic policy is configured such that the amount of bandwidth allocated to the priority traffic exceeds the amount of bandwidth available on the interface, the traffic policy will be suspended. Previously, the policy map would have been rejected. Now that it is only suspended, you have the option of modifying the traffic policy accordingly and then reattaching the traffic policy to the interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-type interface-number*
4. **service-policy** {**input** | **output**} *policy-map-name*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-type interface-number* **Example:** `Router(config)# interface serial0`	Configures an interface type and enters interface configuration mode. • Enter the interface type and interface number.
Step 4	**service-policy** {**input** | **output**} *policy-map-name* **Example:** `Router(config-if)# service-policy input policy1`	Attaches a policy map to an interface. • Enter either the **input** or **output** keyword and the policy map name.
Step 5	**end** **Example:** `Router (config-if)# end`	(Optional) Exits interface configuration mode and returns to privileged EXEC mode.

Verifying the Traffic Class and Traffic Policy Information

To display and verify the information about a traffic class or traffic policy, perform the following steps.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show class-map** **Example:** `Router# show class-map`	(Optional) Displays all class maps and their matching criteria.
Step 3	**show policy-map** *policy-map-name* **class** *class-name* **Example:** `Router# show policy-map policy1 class class1`	(Optional) Displays the configuration for the specified class of the specified policy map. • Enter the policy map name and the class name.
Step 4	**show policy-map** **Example:** `Router# show policy-map`	(Optional) Displays the configuration of all classes for all existing policy maps.
Step 5	**show policy-map interface** *interface-type interface-number* **Example:** `Router# show policy-map interface serial0`	(Optional) Displays the statistics and the configurations of the input and output policies that are attached to an interface. • Enter the interface type and number.
Step 6	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

Configuration Examples for Applying QoS Features Using the MQC

This section provides the following Modular QoS CLI configuration examples:

- Creating a Traffic Class: Example
- Creating a Traffic Policy: Example
- Attaching a Traffic Policy to an Interface: Example
- match not Command: Example
- Default Traffic Class Configuration: Example
- class-map match-any and class-map match-all Commands: Example
- Traffic Class as a Match Criterion (Nested Traffic Classes): Example
- Traffic Policy as a QoS Policy (Hierarchical Traffic Policies): Example

Creating a Traffic Class: Example

In the following example, two traffic classes are created and their match criteria are defined. For the first traffic class called class1, access control list (ACL) 101 is used as the match criterion. For the second traffic class called class2, ACL 102 is used as the match criterion. Packets are checked against the contents of these ACLs to determine if they belong to the class.

```
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
```

Creating a Traffic Policy: Example

In the following example, a traffic policy called policy1 is defined. The traffic policy contains the QoS features to be applied to two classes—class1 and class2. The match criteria for these classes were previously defined (as described in the "Creating a Traffic Class: Example").

For class1, the policy includes a bandwidth allocation request and a maximum packet count limit for the queue reserved for the class. For class2, the policy specifies only a bandwidth allocation request.

```
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
```

Attaching a Traffic Policy to an Interface: Example

The following example shows how to attach an existing traffic policy to an interface. After you define a traffic policy with the **policy-map** command, you can attach it to one or more interfaces by using the **service-policy** command in interface configuration mode. Although you can assign the same traffic policy to multiple interfaces, each interface can have only one traffic policy attached in the input direction and only one traffic policy attached in the output direction.

```
Router(config)# interface ethernet1/1
Router(config-if)# service-policy output policy1
Router(config-if)# exit

Router(config)# interface fastethernet1/0/0
Router(config-if)# service-policy output policy1
Router(config-if)# exit
```

match not Command: Example

The **match not** command is used to specify a specific QoS policy value that is not used as a match criterion. When using the **match not** command, all other values of that QoS policy become successful match criteria.

For instance, if the **match not qos-group 4** command is issued in class-map configuration mode, the specified class will accept all QoS group values except 4 as successful match criteria.

In the following traffic class, all protocols except IP are considered successful match criteria:

```
Router(config)# class-map noip
Router(config-cmap)# match not protocol ip
Router(config-cmap)# exit
```

Default Traffic Class Configuration: Example

Unclassified traffic (traffic that does not meet the match criteria specified in the traffic classes) is treated as belonging to the default traffic class.

If you do not configure a default class, packets are still treated as members of the default class. However, by default, the default class has no QoS features enabled. Therefore, packets belonging to a default class have no QoS functionality. These packets are placed into a first-in, first-out (FIFO) queue managed by tail drop. Tail drop is a means of avoiding congestion that treats all traffic equally and does not differentiate between classes of service. Queues fill during periods of congestion. When the output queue is full and tail drop is in effect, packets are dropped until the congestion is eliminated and the queue is no longer full.

The following example configures a traffic policy for the default class of the traffic policy called policy1. The default class (which is always called class-default) has these characteristics: 10 queues for traffic that does not meet the match criteria of other classes whose policy is defined by the traffic policy policy1, and a maximum of 20 packets per queue before tail drop is enacted to handle additional queued packets.

```
Router(config)# policy-map policy1
Router(config-pmap)# class class-default
Router(config-pmap-c)# fair-queue 10
Router(config-pmap-c)# queue-limit 20
```

class-map match-any and class-map match-all Commands: Example

This example illustrates the difference between the **class-map match-any** command and the **class-map match-all** command. The **match-any** and **match-all** keywords determine how packets are evaluated when multiple match criteria exist. Packets must either meet all of the match criteria (**match-all**) or meet one of the match criterion (**match-any**) to be considered a member of the traffic class.

The following example shows a traffic class configured with the **class-map match-all** command:

```
Router(config)# class-map match-all cisco1
Router(config-cmap)# match protocol ip
Router(config-cmap)# match qos-group 4
Router(config-cmap)# match access-group 101
```

If a packet arrives on a router with the traffic class called cisco1 configured on the interface, the packet is evaluated to determine if it matches the IP protocol, QoS group 4, *and* access group 101. If all three of these match criteria are met, the packet is classified as a member of the traffic class cisco1.

The following example shows a traffic class configured with the **class-map match-any** command:

```
Router(config)# class-map match-any cisco2
Router(config-cmap)# match protocol ip
Router(config-cmap)# match qos-group 4
Router(config-cmap)# match access-group 101
```

In the traffic class called cisco2, the match criteria are evaluated consecutively until a successful match criterion is located. The packet is first evaluated to the determine whether the IP protocol can be used as a match criterion. If the IP protocol can be used as a match criterion, the packet is matched to traffic class cisco2. If the IP protocol is not a successful match criterion, then QoS group 4 is evaluated as a match criterion. Each criterion is evaluated to see if the packet matches that criterion. Once a successful match occurs, the packet is classified as a member of traffic class cisco2. If the packet matches none of the specified criteria, the packet is classified as a member of the default traffic class (class default-class).

Note that the **class-map match-all** command requires that *all* of the match criteria be met in order for the packet to be considered a member of the specified traffic class (a logical AND operator). In the first example, protocol IP AND QoS group 4 AND access group 101 must be successful match criteria. However, only one match criterion must be met in order for the packet in the **class-map match-any** command to be classified as a memberof the traffic class (a logical OR operator). In the second example, protocol IP OR QoS group 4 OR access group 101 must be successful match criterion.

Traffic Class as a Match Criterion (Nested Traffic Classes): Example

There are two reasons to use the **match class-map** command. One reason is maintenance; if a large traffic class currently exists, using the traffic class match criterion is simply easier than retyping the same traffic class configuration. The more common reason for the **match class-map** command is to allow users to use match-any and match-all statements in the same traffic class. If you want to combine match-all and match-any characteristics in a traffic policy, create a traffic class using one match criterion evaluation instruction (either match-any or match-all) and then use this traffic class as a match criterion in a traffic class that uses a different match criterion type.

Here is a possible scenario: Suppose A, B, C, and D were all separate match criterion, and you wanted traffic matching A, B, or C and D (A or B or [C and D]) to be classified as belonging to the traffic class. Without the nested traffic class, traffic would either have to match all 4 of the match criterion (A and B and C and D) or match any of the match criterion (A or B or C or D) to be considered part of the traffic class. You would not be able to combine “and” (match-all) and “or” (match-any) statements within the traffic class, and you would therefore be unable to configure the desired configuration.

The solution: Create one traffic class using match-all for C and D (which we will call criterion E), and then create a new match-any traffic class using A, B, and E. The new traffic class would have the correct evaluation sequence (A or B or E, which would also be A or B or [C and D]). The desired traffic class configuration has been achieved.

The only method of mixing match-all and match-any statements in a traffic class is through the use of the traffic class match criterion.

Nested Traffic Class for Maintenance: Example

In the following example, the traffic class called class1 has the same characteristics as the traffic class called class2, with the exception that traffic class class1 has added a destination address as a match criterion. Rather than configuring traffic class class1 line by line, you can enter the **match class-map class2** command. This command allows all of the characteristics in the traffic class called class2 to be included in the traffic class called class1, and you can simply add the new destination address match criterion without reconfiguring the entire traffic class.

```
Router(config)# class-map match-any class2
Router(config-cmap)# match protocol ip
Router(config-cmap)# match qos-group 3
Router(config-cmap)# match access-group 2
Router(config-cmap)# exit

Router(config)# class-map match-all class1
Router(config-cmap)# match class-map class2
Router(config-cmap)# match destination-address mac 00.00.00.00.00.00
Router(config-cmap)# exit
```

Nested Traffic Class to Combine match-any and match-all Characteristics in One Traffic Class: Example

The only method of including both match-any and match-all characteristics in a single traffic class is to use the **match class-map** command. To combine match-any and match-all characteristics into a single class, a traffic class created with the match-any instruction must use a class configured with the match-all instruction as a match criterion (through the **match class-map** command) or vice versa.

The following example shows how to combine the characteristics of two traffic classes, one with match-any and one with match-all characteristics, into one traffic class with the **match class-map** command. The result requires a packet to match one of the following three match criteria to be considered a member of traffic class class4: IP protocol *and* QoS group 4, destination MAC address 00.00.00.00.00.00, or access group 2.

In this example, only the traffic class called class4 is used with the traffic policy called policy1.

```
Router(config)# class-map match-all class3
Router(config-cmap)# match protocol ip
Router(config-cmap)# match qos-group 4
Router(config-cmap)# exit

Router(config)# class-map match-any class4
Router(config-cmap)# match class-map class3
Router(config-cmap)# match destination-address mac 00.00.00.00.00.00
Router(config-cmap)# match access-group 2
Router(config-cmap)# exit

Router(config)# policy-map policy1
Router(config-pmap)# class class4
Router(config-pmap-c)# police 8100 1500 2504 conform-action transmit exceed-action
set-qos-transmit 4
Router(config-pmap-c)# end
```

Traffic Policy as a QoS Policy (Hierarchical Traffic Policies): Example

A traffic policy can be nested within a QoS policy when the **service-policy** command is used in policy-map class configuration mode. A traffic policy that contains a nested traffic policy is called a hierarchical traffic policy.

A hierarchical traffic policy contains a child policy and a parent policy. The child policy is the previously defined traffic policy that is being associated with the new traffic policy through the use of the **service-policy** command. The new traffic policy using the preexisting traffic policy is the parent policy. In the example in this section, the traffic policy called child is the child policy and traffic policy called parent is the parent policy.

Hierarchical traffic policies can be attached to subinterfaces, Frame Relay PVCs, and ATM PVCs. A hierarchical traffic policy is particularly beneficial when configuring VIP-based distributed FRF.12 (and higher) PVCs. When hierarchical traffic policies are used, a single traffic policy (with a child and a parent policy) can be used to shape and prioritize PVC traffic. In the following example, the child policy is responsible for prioritizing traffic and the parent policy is responsible for shaping traffic. In this configuration, the parent policy allows packets to be sent from the interface, and the child policy determines the order in which the packets are sent.

```
Router(config)# policy-map child
Router(config-pmap)# class voice
Router(config-pmap-c)# priority 50

Router(config)# policy-map parent
Router(config-pmap)# class class-default
Router(config-pmap-c)# shape average 10000000
Router(config-pmap-c)# service-policy child
```

With the exception that the values associated with the **priority** and **shape** commands can be modified, the example is the required configuration for PVCs using FRF.12 (or higher). The value used with the **shape** command is provisioned from the committed information rate (CIR) value from the service provider. For more information about FRF.12 (or higher) PVCs, see the "FRF .20 Support" module.

Additional References

The following sections provide references related to the applying QoS features using the MQC.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
Packet classification	"Classifying Network Traffic" module
FRF PVCs	"FRF .20 Support" module

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Applying QoS Features Using the MQC

Table 3 lists the release history for this feature..

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 3 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 3 Feature Information for Applying QoS Features Using the MQC

Feature Name	Releases	Feature Information
Modular QoS CLI (MQC) Unconditional Packet Discard	12.2(13)T	The Modular QoS CLI (MQC) Unconditional Packet Discard feature allows you to classify traffic matching certain criteria and then configure the system to unconditionally discard any packets matching that criteria.
Class-Based Frame Relay Discard Eligible (DE)-Bit Matching and Marking	12.2(2)T	The Class-Based Frame Relay Discard Eligible (DE)-Bit Matching and Marking feature enhances the MQC to support Frame Relay DE bit matching and marking. Packets with FR DE bit set can be matched to a class and the appropriate QoS feature or treatment be applied.
Modular QoS CLI (MQC)	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

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

Security Device Manager

Security Device Manager Overview

This chapter provides a high-level overview of the Cisco Security Device Manager.

About the Security Device Manager

The Cisco Router and Security Device Manager (SDM) provides an intuitive, graphical user interface for configuring and monitoring advanced IP-based QoS functionality within Cisco routers, and is used to ease QoS configuration and monitoring for a single device.

Additionally, the Cisco SDM provides integrated management of Cisco IOS features like wide-area network (WAN) access, dynamic routing, IPSec virtual private networks (VPNs), firewalls, and intrusion prevention.

For more information about the Cisco SDM, please visit http://www.cisco.com/go/sdm.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

© 2007 Cisco Systems, Inc. All rights reserved.