



Cisco IOS Quality of Service Solutions Configuration Guide

Release 12.4

Corporate 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: 408 526-4100

Customer Order Number: DOC-7817485=
Text Part Number: 78-17485-01



THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

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

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

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.

Cisco IOS Quality of Service Solutions Configuration Guide
© 2005–2006 Cisco Systems, Inc. All rights reserved.



About Cisco IOS Software Documentation for Release 12.4 xlix

Documentation Objectives	xlix
Audience	xlix
Documentation Organization for Cisco IOS Release 12.4	I
Document Conventions	Ivi
Obtaining Documentation	Ivii
Cisco.com	Ivii
Product Documentation DVD	Iviii
Ordering Documentation	Iviii
Documentation Feedback	Iviii
Cisco Product Security Overview	lix
Reporting Security Problems in Cisco Products	lix
Obtaining Technical Assistance	lx
Cisco Technical Support & Documentation Website	lx
Submitting a Service Request	lx
Definitions of Service Request Severity	lxi
Obtaining Additional Publications and Information	lxi

Using Cisco IOS Software for Release 12.4 Ixiii

Understanding Command Modes	Ixiii
Getting Help	Ixiv
Example: How to Find Command Options	Ixv
Using the no and default Forms of Commands	Ixviii
Saving Configuration Changes	Ixviii
Filtering Output from the show and more Commands	Ixix
Finding Additional Feature Support Information	Ixix

Quality of Service Overview 1

What Is Quality of Service?	1
About QoS Architecture	2
Who Could Benefit from Using Cisco IOS QoS?	2
Why Deploy Cisco IOS QoS?	3
End-to-End QoS Models	3
Best-Effort Service	4

Integrated Service	4
Differentiated Service	4
Cisco IOS QoS Features	5
Classification	5
Congestion Management	6
What Is Congestion in Networks?	6
FIFO Queuing	7
PQ	7
FR PIPQ	7
CQ	7
WFQ and DWFO	7
CBWFQ and DCBWFQ	7
IP RTP Priority	8
Frame Relay IP RTP Priority	8
LLQ	8
DLLQ	8
LLQ for Frame Relay	8
Congestion Avoidance	9
WRED	9
DWRED	9
Flow-Based WRED	10
DiffServ Compliant WRED	10
Policing and Shaping	10
signalling	10
Link Efficiency Mechanisms	11
Multilink PPP	11
Frame Relay Fragmentation	12
Header Compression	12
QoS Solutions	12
IP to ATM CoS	12
QoS Features for Voice	13
Differentiated Services Implementations	13
QoS Bandwidth Estimation	13
Modular QoS Command-Line Interface	13
Security Device Manager	13
AutoQoS	13

PART 1: CLASSIFICATION

Classification Overview 17

- About IP Precedence 18
 - How the IP Precedence Bits Are Used to Classify Packets 18
 - Setting or Changing the IP Precedence Value 19
- Policy-Based Routing 20
 - How It Works 20
 - When Should You Use Policy-Based Routing? 21
- QoS Policy Propagation via Border Gateway Protocol 21
 - Restrictions 21
- Committed Access Rate 22
- Marking Network Traffic 22
- Network-Based Application Recognition 22
 - Classification of HTTP by URL, HOST, or MIME 25
 - Classification of Citrix ICA Traffic by Application Name 26
 - Protocol Discovery 27
 - Packet Description Language Module 27
 - Memory Management 27
 - Supported Protocols 28
 - Restrictions 32

Configuring Policy-Based Routing 35

- Policy-Based Routing Configuration Task List 35
 - Enabling PBR 35
 - Enabling Fast-Switched PBR 37
 - Enabling Local PBR 38
 - Enabling CEF-Switched PBR 38
- Policy-Based Routing Configuration Examples 38
 - Equal Access Example 38
 - Differing Next Hops Example 39

Configuring QoS Policy Propagation via Border Gateway Protocol 41

- Policy Propagation via BGP Configuration Task Overview 41
- Policy Propagation via BGP Configuration Task List 42
 - Configuring Policy Propagation Based on Community Lists 42
 - Configuring Policy Propagation Based on the Autonomous System Path Attribute 43
 - Configuring Policy Propagation Based on an Access List 44
 - Monitoring Policy Propagation via BGP 44

Policy Propagation via BGP Configuration Examples 45

Configuring Committed Access Rate 49

- Committed Access Rate Configuration Task List 49
 - Configuring CAR and DCAR for All IP Traffic 50
 - Configuring CAR and DCAR Policies 51
 - IP Precedence or MAC Address 52
 - IP Access List 52
 - Configuring a Class-Based DCAR Policy 52
 - Monitoring CAR and DCAR 53
- CAR and DCAR Configuration Examples 53
 - Subrate IP Services Example 54
 - Input and Output Rate Limiting on an Interface Example 54
 - Rate Limiting in an IXP Example 54
 - Rate Limiting by Access List Example 55

Marking Network Traffic 57

- Contents 57
- Prerequisites for Marking Network Traffic 57
- Restrictions for Marking Network Traffic 58
- Information About Marking Network Traffic 58
 - Purpose of Marking Network Traffic 58
 - Benefits of Marking Network Traffic 59
 - Two Methods for Marking Traffic Attributes 59
 - Method One: Using a set Command 60
 - Method Two: Using a Table Map 61
 - Traffic Marking Procedure Flowchart 62
 - MQC and Network Traffic Marking 64
 - Traffic Classification Compared with Traffic Marking 64
- How to Mark Network Traffic 65
 - Creating a Class Map for Marking Network Traffic 65
 - Creating a Table Map for Marking Network Traffic 66
 - Creating a Policy Map for Applying a QoS Feature to Network Traffic 67
 - Restrictions 67
 - What To Do Next 70
 - Attaching the Policy Map to an Interface 70
 - Configuring QoS When Using IPSec VPNs 72
- Configuration Examples for Marking Network Traffic 74
 - Creating a Class Map for Marking Network Traffic: Example 74
 - Creating a Table Map for Marking Network Traffic: Example 74

Creating a Policy Map for Applying a QoS Feature to Network Traffic: Examples	74
Attaching the Policy Map to an Interface: Example	77
Configuring QoS When Using IPSec VPNs: Example	77
Additional References	77
Related Documents	77
Standards	78
MIBs	78
RFCs	78
Technical Assistance	78
Glossary	79
Feature Information for Marking Network Traffic	79
Classifying Network Traffic	81
Contents	81
Prerequisites for Classifying Network Traffic	81
Information About Classifying Network Traffic	81
Purpose of Classifying Network Traffic	82
Benefits of Classifying Network Traffic	82
MQC and Network Traffic Classification	82
Network Traffic Classification match Commands and Match Criteria	83
Traffic Classification Compared with Traffic Marking	84
How to Classify Network Traffic	85
Creating a Class Map for Classifying Network Traffic	85
Creating a Policy Map for Applying a QoS Feature to Network Traffic	86
What To Do Next	88
Attaching the Policy Map to an Interface	88
Configuring QoS When Using IPSec VPNs	90
Configuration Examples for Classifying Network Traffic	92
Creating a Class Map for Classifying Network Traffic: Example	92
Creating a Policy Map for Applying a QoS Feature to Network Traffic: Example	92
Attaching the Policy Map to an Interface: Example	92
Configuring QoS When Using IPSec VPNs: Example	93
Additional References	93
Related Documents	93
Standards	93
MIBs	94
RFCs	94
Technical Assistance	94
Glossary	94

Feature Information for Classifying Network Traffic 95

Network-Based Application Recognition

Configuring Network-Based Application Recognition 99

- NBAR Configuration Task List 99
 - Configuring a Traffic Class 100
 - Configuring a Traffic Policy 101
 - Attaching a Traffic Policy to an Interface 101
 - Verifying Traffic Policy Configuration 101
 - Monitoring and Maintaining NBAR 102
- NBAR Configuration Example 102
 - Configuring a Traffic Class with NBAR Example 102

Network-Based Application Recognition and Distributed Network-Based Application Recognition 103

- Feature Overview 105
 - Benefits 106
 - NBAR Application Notes 107
 - Catalyst 6000 Family Switches without FlexWAN Modules Application Notes 107
 - Packet Description Language Module 109
 - Classification of HTTP by URL, Host, or MIME 109
 - Classification of Citrix ICA Traffic by Application Name 110
 - RTP Payload Type Classification 111
 - Classification of Custom Applications 112
 - Classification of Peer-to-Peer File-Sharing Applications 113
 - IP NBAR PDL Module Versioning 114
 - Supported Protocols 115
 - Restrictions 121
 - Memory Management 122
 - Related Features and Technologies 122
 - Related Documents 122
- Supported Platforms 122
- Supported Standards, MIBs, and RFCs 123
- Prerequisites 125
- Configuration Tasks 125
 - Enabling Protocol Discovery 126
 - Configuring a Traffic Class 126
 - Configuring a Traffic Policy 127
 - Attaching a Traffic Policy to an Interface 127

Downloading PDLMs	128
Verifying the Configuration	128
Troubleshooting Tips	128
Monitoring and Maintaining NBAR	129
Configuration Examples	129
Configuring a Traffic Policy with NBAR	129
Adding a PDLM	129
Command Reference	130
Glossary	130
Appendix	130
Sample Configuration	130
Network-Based Application Recognition Protocol Discovery Management Information Base	133
Contents	134
Prerequisites for NBAR Protocol Discovery MIBs	134
Restrictions for NBAR Protocol Discovery MIBs	134
Tables Supported by NBAR Protocol Discovery MIBs	134
cnpdSupportedProtocolsTable	135
cnpdStatusTable	135
cnpdAllStatsTable	136
cnpdTopNConfigTable	137
cnpdTopNStatsTable	138
cnpdThresholdConfigTable	139
cnpdThresholdHistoryTable	140
How to Use the NBAR Protocol Discovery MIB	140
Querying the Supported Protocols Table	141
Enabling and Disabling NBAR Protocol Discovery on an Interface	141
Searching the AllStats Table	142
Creating a Top-N Table	143
Setting Protocol Thresholds	145
Setting a Threshold for a Particular Application or Protocol	145
Setting the Any Protocol Threshold	146
Configuration Examples For NBAR Protocol Discovery MIBs	146
Querying Supported Protocols Table	147
Query Status Enabled Table	148
Enabling and Disabling Protocol Discovery	150
Disabling Protocol Discovery on an Interface	151
Searching the All Stats Table	153

Creating Top-N Tables Using the NBAR Protocol Discovery MIB	168
Create a new TopNConfig Entry	168
Change StatsSelect	169
Change SampleTime	169
Change RequestedSize	170
Set Status to createAndGo and Display Results	170
Configuring Thresholds	171
Specific Protocol Threshold	171
Any Protocol Threshold	172
Threshold Options	172
Any Protocol	172
SpecificProtocol (HTTP)	175
Additional References	187
Related Documents	187
Standards	187
MIBs	187
RFCs	188
Technical Assistance	188
Command Reference	188

PART 2: CONGESTION MANAGEMENT

Congestion Management Overview	191
Why Use Congestion Management?	192
Deciding Which Queueing Policy to Use	193
FIFO Queueing	195
Weighted Fair Queueing	195
Flow-Based Weighted Fair Queueing	196
Restrictions	198
WFQ and IP Precedence	198
WFQ and RSVP	199
WFQ and Frame Relay	199
Distributed Weighted Fair Queueing	199
Drop Policy	200
Restrictions	200
Class-Based Weighted Fair Queueing	201
CBWFQ Bandwidth Allocation	202
Why Use CBWFQ?	203
CBWFQ and RSVP	203

Restrictions	203
Distributed Class-Based Weighted Fair Queueing	203
RSVP Interaction with DCBWFQ	204
Benefits	204
Restrictions	205
Prerequisites	205
IP RTP Priority	205
IP RTP Priority Bandwidth Allocation	206
Restrictions	207
Frame Relay IP RTP Priority	208
Frame Relay PVC Interface Priority Queueing	208
Restrictions	209
Prerequisites	209
Low Latency Queueing	209
LLQ Bandwidth Allocation	210
LLQ and Committed Burst Size	212
LLQ and per-VC Hold Queue Support for ATM Adapters	212
Why Use LLQ?	212
Restrictions	213
Distributed Low Latency Queueing	213
Guaranteeing Bandwidth with the priority Command	214
Benefits	214
Restrictions	215
Prerequisites	216
Low Latency Queueing for Frame Relay	216
Restrictions	217
Prerequisites	217
How It Works	217
Custom Queueing	218
How It Works	218
Determining Byte Count Values for Queues	219
How the Byte Count Is Used	220
Determining the Byte Count	220
Window Size	221
Why Use CQ?	221
Restrictions	222
Priority Queueing	222
How It Works	222
How Packets Are Classified for Priority Queueing	223
Why Use Priority Queueing?	223

Restrictions 223
 Bandwidth Management 224

Weighted Fair Queueing

Configuring Weighted Fair Queueing 227

- Flow-Based Weighted Fair Queueing Configuration Task List 227
 - Configuring WFQ 228
 - Monitoring Fair Queueing 229
- Distributed Weighted Fair Queueing Configuration Task List 229
 - Configuring Flow-Based DWFO 230
 - Configuring QoS-Group-Based DWFO 230
 - Configuring Type of Service-Based DWFO 231
 - Monitoring DWFO 231
- Class-Based Weighted Fair Queueing Configuration Task List 231
 - Defining Class Maps 232
 - Configuring Class Policy in the Policy Map 232
 - Configuring Class Policy Using Tail Drop 233
 - Configuring Class Policy Using WRED Packet Drop 234
 - Configuring the Class-Default Class Policy 234
 - Attaching the Service Policy and Enabling CBWFQ 236
 - Modifying the Bandwidth for an Existing Policy Map Class 237
 - Modifying the Queue Limit for an Existing Policy Map Class 237
 - Configuring the Bandwidth Limiting Factor 237
 - Deleting Classes 238
 - Deleting Policy Maps 238
 - Verifying Configuration of Policy Maps and Their Classes 238
- Distributed Class-Based Weighted Fair Queueing Configuration Task List 239
 - Modifying the Bandwidth for an Existing Traffic Class 239
 - Modifying the Queue Limit for an Existing Traffic Class 240
 - Monitoring and Maintaining DCBWFQ 240
- IP RTP Priority Configuration Task List 241
 - Configuring IP RTP Priority 241
 - Configuring the Bandwidth Limiting Factor 241
 - Verifying IP RTP Priority 242
 - Monitoring and Maintaining IP RTP Priority 242
- Frame Relay IP RTP Priority Configuration Task List 242
 - Configuring Frame Relay IP RTP Priority 242
 - Verifying Frame Relay IP RTP Priority 243

Monitoring and Maintaining Frame Relay IP RTP Priority	243
Frame Relay PVC Interface Priority Configuration Task List	243
Configuring PVC Priority in a Map Class	244
Enabling Frame Relay PIPQ and Setting Queue Limits	244
Assigning a Map Class to a PVC	244
Verifying Frame Relay PIPQ	245
Monitoring and Maintaining Frame Relay PIPQ	245
Low Latency Queueing Configuration Task List	245
Configuring LLQ	246
Configuring the Bandwidth Limiting Factor	246
Verifying LLQ	246
Monitoring and Maintaining LLQ	247
Distributed LLQ Configuration Task List	247
Configuring a Priority Queue for an Amount of Available Bandwidth	247
Configuring a Priority Queue for a Percentage of Available Bandwidth	248
Configuring a Transmission Ring Limit	248
Verifying Distributed LLQ	249
Verifying a Transmission Ring Limit	249
Monitoring and Maintaining Distributed LLQ	249
Low Latency Queueing for Frame Relay Configuration Task List	250
Defining Class Maps	250
Configuring Class Policy in the Policy Map	250
Configuring Class Policy for a LLQ Priority Queue	251
Configuring Class Policy Using a Specified Bandwidth and WRED Packet Drop	251
Configuring the Class-Default Class Policy	252
Attaching the Service Policy and Enabling LLQ for Frame Relay	253
Verifying Configuration of Policy Maps and Their Classes	253
Monitoring and Maintaining LLQ for Frame Relay	253
Configuring Burst Size in LLQ Configuration Task List	253
Configuring the LLQ Bandwidth	254
Configuring the LLQ Burst Size	254
Verifying the LLQ Burst Size	254
Per-VC Hold Queue Support for ATM Adapters Configuration Task List	254
Configuring the per-VC Hold Queue on an ATM Adapter	255
Verifying the Configuration of the per-VC Hold Queue on an ATM Adapter	255
Flow-Based WFQ Configuration Examples	255
DWFQ Configuration Examples	255
Flow-Based DWFQ Example	256
QoS-Group-Based DWFQ Example	256

ToS-Based DWFQ Example	257
CBWFQ Configuration Examples	257
Class Map Configuration Example	258
Policy Creation Example	258
Policy Attachment to Interfaces Example	258
CBWFQ Using WRED Packet Drop Example	259
Display Service Policy Map Content Examples	259
All Classes for a Specified Service Policy Map	259
All Classes for All Service Policy Maps	260
Specified Class for a Service Policy Map	260
All Classes for All Service Policy Maps on a Specified Interface	260
Distributed CBWFQ Configuration Examples	261
Traffic Class Configuration Example	261
Traffic Policy Creation Example	262
Traffic Policy Attachment to an Interface Example	262
IP RTP Priority Configuration Examples	262
CBWFQ Configuration Example	263
Virtual Template Configuration Example	263
Multilink Bundle Configuration Example	264
Debug Example	265
Frame Relay IP RTP Priority Configuration Examples	265
Strict Priority Service to Matching RTP Packets Example	265
Frame Relay PVC Interface PQ Configuration Examples	266
LLQ Configuration Examples	267
ATM PVC Configuration Example	267
Virtual Template Configuration Example	267
Multilink Bundle Configuration Example	268
Distributed LLQ Configuration Examples	269
Enabling PQ for an Amount of Available Bandwidth on an ATM Subinterface Example	269
Enabling PQ for a Percentage of Available Bandwidth on an ATM Subinterface Example	270
Limiting the Transmission Ring Limit on an ATM Interface Example	270
Limiting the Transmission Ring Limit on an ATM PVC Subinterface Example	270
LLQ for Frame Relay Configuration Examples	271
Burst Size in LLQ Configuration Examples	272
Per-VC Hold Queue Support for ATM Adapters Examples	272
Low Latency Queueing with Priority Percentage Support	273
Feature Overview	273
Changes to the bandwidth Command	273

Changes to the priority Command	274
How These Commands Calculate Bandwidth	274
Benefits	275
Restrictions	275
Supported Platforms	275
Supported Standards, MIBs, and RFCs	276
Configuration Tasks	276
Specifying the Bandwidth Percentage	276
Verifying the Bandwidth Percentage	277
Configuration Examples	277
Specifying the Bandwidth Percentage	277
Mixing the Units of Bandwidth for Non-Priority Traffic	278
Command Reference	278
Low Latency Queueing (LLQ) for IPSec Encryption Engines	279
Feature Overview	279
Benefits	280
Restrictions	280
Related Features and Technologies	280
Related Documents	281
Supported Platforms	281
Determining Platform Support Through Cisco Feature Navigator	281
Availability of Cisco IOS Software Images	282
Supported Standards, MIBs, and RFCs	282
Prerequisites	282
Configuration Tasks	283
Defining Class Maps	283
Configuring Class Policy in the Policy Map	283
Configuring Class Policy for a Priority Queue	284
Configuring Class Policy Using a Specified Bandwidth	284
Configuring the Class-Default Class Policy	285
Attaching the Service Policy	285
Verifying Configuration of Policy Maps and Their Classes	285
Monitoring and Maintaining LLQ for IPSec Encryption Engines	286
Configuration Examples	286
LLQ for IPSec Encryption Engines Example	286
Command Reference	287
Glossary	287

Configuring Custom Queueing	289
Custom Queueing Configuration Task List	289
Defining the Custom Queue List	290
Specifying the Maximum Size of the Custom Queues	290
Assigning Packets to Custom Queues	291
Monitoring Custom Queue Lists	291
Custom Queueing Configuration Examples	292
Custom Queue List Defined Example	292
Maximum Specified Size of the Custom Queues Examples	292
Packets Assigned to Custom Queues Examples	292
Protocol Type	292
Interface Type	293
Default Queue	293
Configuring Priority Queueing	295
Priority Queueing Configuration Task List	295
Defining the Priority List	295
Assigning Packets to Priority Queues	296
Specifying the Maximum Size of the Priority Queues	296
Assigning the Priority List to an Interface	297
Monitoring Priority Queueing Lists	297
Priority Queueing Configuration Examples	297
Priority Queueing Based on Protocol Type Example	297
Priority Queueing Based on Interface Example	298
Maximum Specified Size of the Priority Queue Example	298
Priority List Assigned to an Interface Example	298
Priority Queueing Using Multiple Rules Example	298

PART 3: CONGESTION AVOIDANCE

Congestion Avoidance Overview	301
Tail Drop	301
Weighted Random Early Detection	302
About Random Early Detection	302
How It Works	302
Packet Drop Probability	303
How TCP Handles Traffic Loss	303
How the Router Interacts with TCP	304
About WRED	305
Why Use WRED?	305

How It Works	305
Average Queue Size	306
Restrictions	307
Distributed Weighted Random Early Detection	307
How It Works	307
Average Queue Size	308
Packet-Drop Probability	308
Why Use DWRED?	309
Restrictions	310
Prerequisites	310
Flow-Based WRED	311
Why Use Flow-Based WRED?	311
How It Works	311
DiffServ Compliant WRED	312
How It Works	312
Usage Scenarios	313
Usage Points to Note	313

Weighted Random Early Detection

Configuring Weighted Random Early Detection	317
Weighted Random Early Detection Configuration Task List	318
Enabling WRED	318
Changing WRED Parameters	319
Monitoring WRED	319
DWRED Configuration Task List	320
Configuring DWRED in a Traffic Policy	320
Configuring DWRED to Use IP Precedence Values in a Traffic Policy	321
Monitoring and Maintaining DWRED	321
Flow-Based WRED Configuration Task List	322
Configuring Flow-Based WRED	322
DiffServ Compliant WRED Configuration Task List	322
Configuring WRED to Use the Differentiated Services Code Point Value	322
WRED at the Interface Level	323
WRED at the per-VC Level	323
WRED at the Class Level	323
Verifying the DSCP Value Configuration	324
WRED Configuration Examples	324
WRED Configuration Example	324

- Parameter-Setting DWRED Example 326
- Parameter-Setting WRED Example 327
- DWRED Configuration Examples 327
 - DWRED on an Interface Example 327
 - Modular QoS CLI Example 327
 - Configuring DWRED in Traffic Policy Example 328
- Flow-Based WRED Configuration Example 328
- DiffServ Compliant WRED Configuration Examples 330
 - WRED Configured to Use the DSCP Value Example 330
 - DSCP Value Configuration Verification Example 331
- WRED — Explicit Congestion Notification 333**
 - Feature Overview 333
 - How WRED Works 334
 - ECN Extends WRED Functionality 334
 - How Packets Are Treated When ECN Is Enabled 335
 - For More Information 335
 - Benefits 335
 - Related Documents 336
 - Supported Platforms 336
 - Supported Standards, MIBs, and RFCs 337
 - Prerequisites 337
 - Configuration Tasks 338
 - Configuring Explicit Congestion Notification 338
 - Verifying the Explicit Congestion Notification Configuration 338
 - Configuration Examples 338
 - Enabling ECN Example 339
 - Verifying the ECN Configuration Example 339
 - Command Reference 340

PART 4: POLICING AND SHAPING

- Policing and Shaping Overview 343**
 - What Is a Token Bucket? 344
 - Policing with CAR 345
 - How It Works 345
 - Matching Criteria 345
 - Rate Limits 346
 - Conform and Exceed Actions 348

Multiple Rate Policies	348
Restrictions	349
Traffic Policing	349
Benefits	350
Bandwidth Management Through Rate Limiting	350
Packet Marking Through IP Precedence, QoS Group, and DSCP Value Setting	350
Restrictions	350
Prerequisites	351
Traffic Shaping (Regulating Packet Flow)	351

Traffic Policing

Configuring Traffic Policing	355
Traffic Policing Configuration Task List	355
Configuring Traffic Policing	355
Verifying the Traffic Policing Configuration	356
Monitoring and Maintaining Traffic Policing	357
Traffic Policing Configuration Examples	357
Traffic Policy that Includes Traffic Policing Example	357
Verifying the Configuration Example	358
Traffic Policing	359
Feature Overview	359
Benefits	360
Restrictions	361
Related Features and Technologies	361
Related Documents	362
Supported Platforms	362
Supported Standards, MIBs, and RFCs	362
Prerequisites	363
Configuration Tasks	363
Configuring Traffic Policing	363
Verifying Traffic Policing	364
Troubleshooting Tips	364
Monitoring and Maintaining Traffic Policing	364
Configuration Examples	364
Configuring a Service Policy that Includes Traffic Policing	365
Command Reference	365
Glossary	366

Two-Rate Policer	367
Feature Overview	367
Benefits	368
Restrictions	369
Related Features and Technologies	370
Related Documents	370
Supported Platforms	370
Supported Standards, MIBs, and RFCs	371
Prerequisites	371
Configuration Tasks	371
Configuring the Two-Rate Policer	372
Verifying the Two-Rate Policer Configuration	373
Troubleshooting Tips	373
Monitoring and Maintaining the Two-Rate Policer	373
Configuration Examples	373
Limiting the Traffic Using a Policer Class Example	373
Command Reference	374
Policer Enhancement — Multiple Actions	375
Feature Overview	375
Benefits	376
Restrictions	376
Related Features and Technologies	377
Related Documents	377
Supported Platforms	377
Supported Standards, MIBs, and RFCs	378
Prerequisites	379
Configuration Tasks	379
Configuring Multiple Policer Actions	379
Verifying the Multiple Policer Actions Configuration	380
Troubleshooting Tips	380
Monitoring and Maintaining the Multiple Policer Actions	380
Configuration Examples	381
Multiple Actions in a Two-Rate Policer Example	381
Verifying the Multiple Policer Actions Example	381
Command Reference	382
Percentage-Based Policing and Shaping	383
Feature Overview	383

Benefits	384
Restrictions	384
Related Features and Technologies	384
Related Documents	384
Supported Platforms	385
Supported Standards, MIBs, and RFCs	385
Prerequisites	386
Configuration Tasks	386
Configuring Policing and Shaping Based on Bandwidth Percentage	386
Attaching the Policy Map to an Interface or a VC	387
Verifying the Policing and Shaping Bandwidth Percentage Setting	387
Troubleshooting Tips	388
Configuration Examples	388
Specifying Traffic Policing Based on a Bandwidth Percentage Example	388
Specifying Traffic Shaping Based on a Bandwidth Percentage Example	388
Verifying That CEF Is Enabled Example	389
Command Reference	390
Modular QoS CLI (MQC) Three-Level Hierarchical Policer	391
Contents	392
Restrictions for the Modular QoS CLI (MQC) Three-Level Hierarchical Policer	392
Information About the Modular QoS CLI (MQC) Three-Level Hierarchical Policer	393
Modular Quality of Service Command-Line Interface (MQC)	394
Packet Flow in the Modular QoS CLI (MQC) Three-Level Hierarchical Policer	394
Other Traffic Policing-Related Features	395
How to Configure the Modular QoS CLI (MQC) Three-Level Hierarchical Policer	395
Configuring Traffic Policing	396
Prerequisites	396
Attaching the Policy Map to an Interface	397
What to Do Next	398
Verifying the Configuration	399
Troubleshooting Tips	400
Configuration Examples for the Modular QoS CLI (MQC) Three-Level Hierarchical Policer	400
Configuring the Modular QoS CLI (MQC) Three-Level Hierarchical Policer Example	400
Additional References	403
Related Documents	403
Standards	403
MIBs	404
RFCs	404

Technical Assistance	404
Command Reference	405
ATM Policing by Service Category for SVC/SoftPVC	407
Feature Overview	407
Benefits	408
Related Features and Technologies	408
Related Documents	408
Supported Platforms	408
Supported Standards, MIBs, and RFCs	408
Configuration Tasks	409
Configuring ATM Policing by Service Category for SVC/SoftPVC	409
Verifying ATM Policing by Service Category for SVC/SoftPVC	409
Troubleshooting Tips	410
Monitoring and Maintaining ATM Policing by Service Category for SVC/SoftPVC	410
Example: Monitoring and Maintaining ATM Policing by Service Category for SVC/SoftPVC	411
Configuration Examples	411
Non-UBR Traffic Policing	412
Command Reference	412
Glossary	413
Modular QoS CLI (MQC) Unconditional Packet Discard	415
Feature Overview	415
Benefits	416
Restrictions	416
Related Features and Technologies	416
Related Documents	416
Supported Platforms	416
Supported Standards, MIBs, and RFCs	417
Configuration Tasks	418
Configuring the Class Map	418
Creating a Policy Map	419
Attaching the Policy Map to an Interface or a VC	419
Verifying the Discard Action Configuration in the Traffic Class	420
Configuration Examples	420
Configuring the Discard Action Configuration in a Traffic Class Example	420
Verifying the Discard Action Configuration in the Policy Map Example	421
Command Reference	421

Control Plane Policing	423
Contents	423
Prerequisites for Control Plane Policing	424
Restrictions for Control Plane Policing	424
Information About Control Plane Policing	425
Benefits of Control Plane Policing	425
Terms to Understand	425
Control Plane Security and Packet QoS Overview	427
Aggregate Control Plane Services	428
Distributed Control Plane Services	428
When Distributed CP Services Are Necessary	429
Output Rate-Limiting and Silent Mode Operation	430
How to Use the Control Plane Policing Feature	430
Defining Aggregate Control Plane Services	431
Prerequisites	431
Restrictions	431
Defining Distributed Control Plane Services	432
Prerequisites	432
Restrictions	432
Verifying Aggregate CP Services	433
Examples	434
Verifying Distributed CP Services	435
Examples	436
Configuration Examples for Control Plane Policing	437
Configuring Rate Limiting (Input) Telnet Traffic: Example	437
Configuring Rate Limiting (Output) Telnet Traffic: Example	437
Configuring Rate Limiting (Input) for Distributed CP Traffic: Example	438
Additional References	439
Related Documents	439
Standards	439
MIBs	439
RFCs	439
Technical Assistance	439
Command Reference	440

Packet Flow

Regulating Packet Flow Roadmap 443

Regulating Packet Flow Using Traffic Shaping 445

Contents 445

Information About Traffic Shaping 445

Benefits of Shaping Traffic on a Network 446

Cisco Traffic Shaping Mechanisms 446

Token Bucket and Traffic Shaping 447

Traffic Shaping and Rate of Transfer 448

How Traffic Shaping Regulates Traffic 448

Traffic Shaping versus Traffic Policing 450

Where to Go Next 450

Additional References 451

Related Documents 451

Standards 451

MIBs 452

RFCs 452

Technical Assistance 452

Regulating Packet Flow on a Per-Class Basis — Using Class-Based Traffic Shaping 453

Contents 453

Prerequisites for Configuring Class-Based Traffic Shaping 453

Restrictions for Configuring Class-Based Traffic Shaping 454

Information About Class-Based Traffic Shaping 454

Class-Based Traffic Shaping Functionality 454

Benefits of Class-Based Traffic Shaping 455

Hierarchical Policy Map Structure of Class-Based Traffic Shaping 455

How to Configure Class-Based Traffic Shaping 456

Configuring Class-Based Traffic Shaping in a Primary-Level (Parent) Policy Map 456

Prerequisites 457

What to Do Next 458

Configuring the Secondary-Level (Child) Policy Map 458

Configuration Examples for Class-Based Traffic Shaping 460

Class-Based Traffic Shaping Configuration: Example 460

Where to Go Next 461

Additional References 461

Related Documents 461

Standards	462
MIBs	462
RFCs	462
Technical Assistance	462
Feature Information for Class-Based Traffic Shaping	462
Regulating Packet Flow on a Per-Interface Basis — Using Generic Traffic Shaping	465
Contents	465
Prerequisites for Configuring Generic Traffic Shaping	465
Restrictions for Configuring Generic Traffic Shaping	466
Information About Configuring Generic Traffic Shaping	466
Generic Traffic Shaping Functionality	466
Adaptive Generic Traffic Shaping on Frame Relay Networks	467
Benefits of Generic Traffic Shaping	467
How to Configure Generic Traffic Shaping	467
Configuring Generic Traffic Shaping on an Interface	467
Configuring Generic Traffic Shaping Using an Access Control List	469
Access Control List Functionality	469
Configuring Adaptive Generic Traffic Shaping for Frame Relay Networks	470
Configuration Examples for Generic Traffic Shaping	472
Generic Traffic Shaping on an Interface Configuration: Example	472
Generic Traffic Shaping Using an Access Control List Configuration: Example	472
Adaptive Generic Traffic Shaping for a Frame Relay Network Configuration: Example	473
Where to Go Next	473
Additional References	473
Related Documents	473
Standards	474
MIBs	474
RFCs	474
Technical Assistance	474
Feature Information for Generic Traffic Shaping	474

PART 5: SIGNALLING

Signalling Overview	479
IP Precedence	479
Resource Reservation Protocol	480
How It Works	481
RSVP Support for Low Latency Queueing	481

- Restrictions 483
- Prerequisites 483
- RSVP Support for Frame Relay 483
 - RSVP Bandwidth Allocation and Modular QoS Command Line Interface (CLI) 484
 - Benefits 484
 - Restrictions 485
 - Prerequisites 485
- RSVP-ATM QoS Interworking 485
 - How It Works 486
 - An Example Scenario 487
- COPS for RSVP 488
 - How It Works 489
 - A Detailed Look at COPS for RSVP Functioning 490
- Subnetwork Bandwidth Manager 494

RSVP

- Configuring RSVP 499**
 - RSVP Reservation Types 500
 - Distinct Reservation 500
 - Shared Reservation 501
 - Planning for RSVP Configuration 501
 - RSVP Implementation Considerations 502
 - Frame Relay Internetwork Considerations 502
 - ATM Internetwork Considerations 503
 - Resource Reservation Protocol Configuration Task List 503
 - Enabling RSVP 503
 - Entering Senders in the RSVP Database 504
 - Entering Receivers in the RSVP Database 504
 - Specifying Multicast Destinations 505
 - Controlling Which RSVP Neighbor Can Offer a Reservation 505
 - Enabling RSVP to Attach to NetFlow 505
 - Setting the IP Precedence and ToS Values 506
 - Monitoring RSVP 506
 - RSVP Configuration for a Multicast Session Example 507
- Control Plane DSCP Support for RSVP 513**
 - Feature Overview 513
 - Benefits 514

Restrictions	514
Related Features and Technologies	514
Related Documents	515
Supported Platforms	515
Supported Standards, MIBs, and RFCs	515
Prerequisites	515
Configuration Tasks	515
Enabling RSVP on an Interface	516
Specifying the DSCP	516
Verifying Control Plane DSCP Support for RSVP Configuration	516
Monitoring and Maintaining Control Plane DSCP Support for RSVP	517
Configuration Examples	517
Command Reference	517
Glossary	518
RSVP Scalability Enhancements	519
Feature Overview	519
Benefits	520
Restrictions	520
Related Features and Technologies	521
Related Documents	521
Supported Platforms	521
Supported Standards, MIBs, and RFCs	521
Prerequisites	521
Configuration Tasks	522
Enabling RSVP on an Interface	522
Setting the Resource Provider	522
Disabling Data Packet Classification	523
Configuring Class and Policy Maps	523
Attaching a Policy Map to an Interface	524
Verifying RSVP Scalability Enhancements Configuration	524
Monitoring and Maintaining RSVP Scalability Enhancements	526
Configuration Examples	526
Configuring CBWFQ to Accommodate Reserved Traffic	526
Configuring the Resource Provider as None with Data Classification Turned Off	527
Command Reference	531
Glossary	532

RSVP Support for ATM/PVCs	533
Feature Overview	533
RSVP Bandwidth Allocation and Modular QoS Command Line Interface (CLI)	534
Admission Control	534
Data Packet Classification	534
Benefits	534
Restrictions	535
Related Features and Technologies	535
Related Documents	536
Supported Platforms	536
Supported Standards, MIBs, and RFCs	536
Prerequisites	536
Configuration Tasks	536
Creating a PVC	537
Defining ATM QoS Traffic Parameters for a PVC	537
Defining a Policy Map for WFQ	538
Applying a Policy Map to a PVC	538
Enabling RSVP on an Interface	538
Configuring a Path	538
Configuring a Reservation	538
Verifying RSVP Support for ATM/PVCs Configuration	539
Multipoint Configuration	539
Point-to-Point Configuration	541
Monitoring and Maintaining RSVP Support for ATM/PVCs	542
Configuration Examples	542
Point-to-Point Configuration	542
Multipoint Configuration	544
Command Reference	545
Glossary	546
RSVP Local Policy Support	549
Feature Overview	549
Benefits	550
Related Features and Technologies	550
Related Documents	550
Supported Platforms	551
Supported Standards, MIBs, and RFCs	551
Prerequisites	552

Configuration Tasks	552
Creating an RSVP Local Policy	552
Specifying Command Line Interface (CLI) Submodes	552
Verifying RSVP Local Policy Configuration	553
Monitoring and Maintaining RSVP Local Policy Support	554
Configuration Examples	554
RSVP Local Policy Support Example	554
Command Reference	555
Glossary	556
RSVP Refresh Reduction and Reliable Messaging	557
Contents	558
Prerequisites for RSVP Refresh Reduction and Reliable Messaging	558
Restrictions for RSVP Refresh Reduction and Reliable Messaging	558
Information About RSVP Refresh Reduction and Reliable Messaging	558
Feature Design of RSVP Refresh Reduction and Reliable Messaging	559
Types of Messages in RSVP Refresh Reduction and Reliable Messaging	559
Reliable Messages	560
Bundle Messages	560
Summary Refresh Messages	561
Benefits of RSVP Refresh Reduction and Reliable Messaging	561
How to Configure RSVP Refresh Reduction and Reliable Messaging	561
Enable RSVP on an Interface	561
Enable RSVP Refresh Reduction	562
Verify RSVP Refresh Reduction and Reliable Messaging	564
Configuration Examples for RSVP Refresh Reduction and Reliable Messaging	565
RSVP Refresh Reduction and Reliable Messaging Example	565
Additional References	567
Related Documents	567
Standards	567
MIBs	567
RFCs	568
Technical Assistance	568
Command Reference	569
Obsolete and Replaced Commands	570
Glossary	571
RSVP Support for RTP Header Compression, Phase 1	573
Contents	573

Prerequisites for RSVP Support for RTP Header Compression, Phase 1	574
Restrictions for RSVP Support for RTP Header Compression, Phase 1	574
Information About RSVP Support for RTP Header Compression, Phase 1	574
Feature Design of RSVP Support for RTP Header Compression, Phase 1	574
Predicting Compression within Admission Control	575
Benefits of RSVP Support for RTP Header Compression, Phase 1	575
How to Configure RSVP Support for RTP Header Compression, Phase 1	576
Configuring RSVP Admission-Control Compression	576
Verifying RSVP Support for RTP Header Compression, Phase 1 Configuration	577
Examples	578
Troubleshooting Tips	579
Configuration Examples for RSVP Support for RTP Header Compression, Phase 1	580
RSVP Support for RTP Header Compression, Phase 1 Example	580
Additional References	581
Related Documents	581
Standards	581
MIBs	581
RFCs	582
Technical Assistance	582
Command Reference	583
Glossary	584
RSVP Message Authentication	585
Contents	585
Prerequisites for RSVP Message Authentication	585
Restrictions for RSVP Message Authentication	586
Information About RSVP Message Authentication	586
Feature Design of RSVP Message Authentication	586
Special Considerations for RSVP Message Authentication	588
Benefits of RSVP Message Authentication	588
How to Configure RSVP Message Authentication	589
Enabling RSVP on an Interface	589
Configuring an RSVP Authentication Type	590
Configuring an RSVP Authentication Key	591
Enabling RSVP Key Encryption	592
Enabling RSVP Authentication Challenge	593
Configuring RSVP Authentication Lifetime	594
Configuring RSVP Authentication Window Size	595
Activating RSVP Authentication	596

Verifying RSVP Message Authentication	597
Examples	598
Troubleshooting Tips	599
Configuration Examples for RSVP Message Authentication	600
RSVP Message Authentication Example	600
Additional References	602
Related Documents	602
Standards	602
MIBs	602
RFCs	603
Technical Assistance	603
Command Reference	604
Glossary	605
Configuring RSVP Support for LLQ	607
RSVP Support for LLQ Configuration Task List	607
Configuring Flow Classification	608
Enabling RSVP and WFQ	608
Configuring a Burst Factor	608
Configuring a Path	608
Configuring a Reservation	609
Verifying RSVP Support for LLQ Configuration	609
Monitoring and Maintaining RSVP Support for LLQ	610
RSVP Support for LLQ Configuration Examples	610
Configuring RSVP Support for Frame Relay	613
RSVP Support for Frame Relay Configuration Task List	613
Enabling Frame Relay Encapsulation on an Interface	614
Configuring a Virtual Circuit	614
Enabling Frame Relay Traffic Shaping on an Interface	614
Enabling Enhanced Local Management Interface	614
Enabling RSVP on an Interface	615
Specifying a Traffic Shaping Map Class for an Interface	615
Defining a Map Class with WFQ and Traffic Shaping Parameters	615
Specifying the CIR	615
Specifying the Minimum CIR	616
Enabling WFQ	616
Enabling FRF.12	616
Configuring a Path	616
Configuring a Reservation	617

Verifying RSVP Support for Frame Relay	617
Multipoint Configuration	617
Point-to-Point Configuration	618
Monitoring and Maintaining RSVP Support for Frame Relay	619
RSVP Support for Frame Relay Configuration Examples	619
Multipoint Configuration Example	619
Point-to-Point Configuration Example	622
Configuring RSVP-ATM QoS Interworking	625
RSVP-ATM QoS Interworking Configuration Task List	625
Enabling RSVP and Limiting Reservable Bandwidth	626
Enabling Creation of SVCs for Reserved Flows	626
Limiting the Peak Rate Applied to the PCR for SVCs	628
Configuring per-VC DWRED	628
Monitoring RSVP-ATM Configuration for an Interface	629
RSVP-ATM QoS Interworking Configuration Examples	629
Configuring COPS for RSVP	635
COPS for RSVP Configuration Task List	635
Specifying COPS Servers and Enabling COPS for RSVP	636
Restricting RSVP Policy to Specific Access Control Lists	636
Rejecting Unmatched RSVP Messages	636
Confining Policy to PATH and RESV Messages	636
Retaining RSVP Information After Losing Connection with the COPS Server	637
Reporting the Results of Outsourcing and Configuration Decisions	637
Verifying the Configuration	637
COPS for RSVP Configuration Examples	637
COPS Server Specified Example	638
RSVP Behavior Customized Example	638
Verification of the COPS for RSVP Configuration Example	638
Configuring Subnetwork Bandwidth Manager	639
Subnetwork Bandwidth Manager Configuration Task List	639
Configuring an Interface as a Designated SBM Candidate	640
Configuring the NonResvSendLimit Object	640
Verifying Configuration of SBM State	641
Subnetwork Bandwidth Manager Candidate Configuration Example	642

PART 6: LINK EFFICIENCY MECHANISMS

Link Efficiency Mechanisms Overview 645

- Multilink PPP 645
- Frame Relay Fragmentation 645
- Header Compression 646

Reduction of Latency and Jitter for Real-Time Traffic Using Multilink PPP

Reducing Latency and Jitter Using Multilink PPP Roadmap 649

Reducing Latency and Jitter for Real-Time Traffic Using Multilink PPP 651

- Contents 651
- Information About Multilink 651
 - Restrictions for Multilink 652
 - Multilink Functionality 652
 - Multilink Interleaving 652
 - Multilink Fragmentation 652
 - Multilink Resequencing 654
 - Multilink Bundles and Their Network Links 654
 - Multiclass Multilink PPP 654
 - Distributed Multilink PPP 655
- Where to Go Next 655
- Additional References 656
 - Related Documents 656
 - Standards 656
 - MIBs 657
 - RFCs 657
 - Technical Assistance 657
- Glossary 657

Using Multilink PPP over ATM Links 659

- Contents 659
- Prerequisites for Using Multilink PPP over ATM Links 659
- Restrictions for Using Multilink PPP over ATM Links 660
- Information About Using Multilink PPP over ATM Links 660
 - MQC and Multilink PPP over ATM Links 660
 - Virtual Template Interfaces 660**
 - Multilink Group Interfaces 660

- How to Configure Multilink PPP over ATM Links **661**
 - Configuring Multilink PPP over ATM Links on a Virtual Template Interface **661**
 - Prerequisites **661**
 - Configuring Multilink PPP over ATM Links on a Multilink Group Interface **663**
 - Prerequisites **663**
 - What to Do Next **665**
 - Associating the Virtual Template Interface with the Multilink Group **665**
 - Associating the Virtual Template Interface with an ATM PVC **666**
 - Verifying the Multilink PPP over ATM Links Configuration **668**
- Configuration Examples for Using Multilink PPP over ATM Links **669**
 - Configuring Multilink PPP over ATM Links on a Virtual Template Interface: Example **669**
 - Configuring Multilink PPP over ATM Links on a Multilink Group Interface: Example **670**
 - Associating the Virtual Template Interface with the Multilink Group: Example **670**
 - Associating the Virtual Template Interface with an ATM PVC: Example **670**
 - Verifying the Multilink PPP over ATM Links Configuration: Example **670**
- Where to Go Next **671**
- Additional References **671**
 - Related Documents **671**
 - Standards **672**
 - MIBs **672**
 - RFCs **672**
 - Technical Assistance **672**
- Glossary **673**
- Feature Information for Using Multilink PPP over ATM Links **673**
- Using Multilink PPP over Dialer Interface Links 675**
 - Contents **675**
 - Prerequisites for Using Multilink PPP over Dialer Interface Links **675**
 - Restrictions for Using Multilink PPP over Dialer Interface Links **676**
 - Information About Using Multilink PPP over Dialer Interface Links **676**
 - Dialer Profiles **676**
 - MQC and Multilink PPP over Dialer Interface Links **677**
 - How to Configure Multilink PPP over Dialer Interface Links **677**
 - Configuring Multilink PPP over Dialer Interface Links **677**
 - Prerequisites **677**
 - Associating the Dialer Interface with a BRI **680**
 - Verifying the Multilink PPP over Dialer Interface Link Configuration **681**
 - Configuration Examples for Using Multilink PPP over Dialer Interface Links **682**
 - Configuring Multilink PPP over Dialer Interface Links: Example **682**

Associating the Dialer Interface with a BRI: Example	683
Verifying the Multilink PPP over Dialer Interface Link Configuration: Example	683
Where to Go Next	683
Additional References	684
Related Documents	684
Standards	684
MIBs	684
RFCs	685
Technical Assistance	685
Glossary	685
Feature Information for Using Multilink PPP over Dialer Interface Links	686
Using Multilink PPP over Frame Relay	689
Contents	689
Prerequisites for Using Multilink PPP over Frame Relay	690
Restrictions for Using Multilink PPP over Frame Relay	690
Information About Using Multilink PPP over Frame Relay	690
Frame Relay Traffic Shaping and Multilink PPP over Frame Relay	690
MQC and Multilink PPP over Frame Relay	691
Virtual Template Interfaces	691
Multilink Group Interfaces	691
How to Configure Multilink PPP over Frame Relay	691
Configuring Multilink PPP over Frame Relay on a Virtual Template Interface	692
Prerequisites	692
Configuring Multilink PPP over Frame Relay on a Multilink Group Interface	694
Prerequisites	694
What to Do Next	696
Associating the Virtual Template Interface with the Multilink Group	696
Associating the Virtual Template Interface with a Frame Relay PVC	697
Verifying the Multilink PPP over Frame Relay Configuration	698
Configuration Examples for Multilink PPP over Frame Relay	699
Configuring Multilink PPP over Frame Relay on a Virtual Template Interface: Example	700
Configuring Multilink PPP over Frame Relay on a Multilink Group Interface: Example	700
Associating the Virtual Template Interface with the Multilink Group: Example	700
Associating the Virtual Template Interface with a Frame Relay PVC: Example	701
Verifying the Multilink PPP over Frame Relay Configuration: Example	701
Where to Go Next	701
Additional References	702
Related Documents	702

Standards	702
MIBs	702
RFCs	703
Technical Assistance	703
Glossary	703
Feature Information for Using Multilink PPP over Frame Relay	704
Using Multilink PPP over Serial Interface Links	707
Contents	707
Prerequisites for Using Multilink PPP over Serial Interface Links	707
Restrictions for Using Multilink PPP over Serial Interface Links	708
Information About Using Multilink PPP over Serial Interface Links	708
MQC and Multilink PPP over Serial Interface Links	708
Multilink Group Interfaces	709
How to Configure Multilink PPP over Serial Interface Links	709
Configuring Multilink PPP over Serial Interface Links on a Multilink Group Interface	709
Prerequisites	709
Associating the Serial Interface with the Multilink Group	711
Verifying the Multilink PPP over Serial Interface Link Configuration	712
Configuration Examples for Using Multilink PPP over Serial Interface Links	713
Configuring Multilink PPP over Serial Interface Links on a Multilink Group Interface: Example	713
Associating the Serial Interface with the Multilink Group: Example	714
Verifying the Multilink PPP over Serial Interface Link Configuration: Example	714
Where to Go Next	714
Additional References	715
Related Documents	715
Standards	715
MIBs	715
RFCs	715
Technical Assistance	716
Glossary	716
Feature Information for Using Multilink PPP over Serial Interface Links	717

Header Compression

Header-Compression Features Roadmap	721
--	------------

Header Compression	723
---------------------------	------------

Contents	723
----------	-----

Information About Header Compression	723
Header Compression Defined	724
Types of Header Compression	724
RTP Functionality and Header Compression	724
How RTP Header Compression Works	724
Why Use RTP Header Compression	725
TCP Functionality and Header Compression	726
How TCP Header Compression Works	726
Why Use TCP Header Compression	727
Class-Based Header Compression Functionality	727
Why Use Class-Based Header Compression	728
Where to Go Next	728
Additional References	728
Related Documents	728
Standards	728
MIBs	729
RFCs	729
Technical Assistance	729
Glossary	730
Configuring RTP Header Compression	731
Contents	731
Prerequisites for Configuring RTP Header Compression	732
Information About Configuring RTP Header Compression	732
Configurable RTP Header-Compression Settings	732
RTP Header-Compression Keywords	732
Enhanced RTP Header Compression	734
RTP Header Compression over Satellite Links	734
Periodic Refreshes of a Compressed Packet Stream	734
Optional Disabling of Context-Status Messages	735
How to Configure RTP Header Compression	735
Enabling RTP Header Compression on an Interface	735
Enabling RTP Header Compression on an Interface That Uses Frame Relay Encapsulation	736
Restrictions	737
Enabling Enhanced RTP Header Compression	738
Prerequisites	738
Restrictions	739
Enabling RTP Header Compression over a Satellite Link	740
Specifying the Header-Compression Settings	741

Changing the Number of Header-Compression Connections	742
Implications of Changing the Number of Header-Compression Connections	743
Restrictions	743
Displaying Header-Compression Statistics	744
Configuration Examples for RTP Header Compression	745
Enabling RTP Header Compression on an Interface: Example	746
Enabling RTP Header Compression on an Interface That Uses Frame Relay Encapsulation: Example	746
Enabling Enhanced RTP Header Compression: Example	746
Enabling RTP Header Compression over a Satellite Link: Example	747
Specifying the Header-Compression Settings: Example	747
Changing the Number of Header-Compression Connections: Example	747
Displaying Header-Compression Statistics: Example	747
Additional References	748
Related Documents	748
Standards	748
MIBs	748
RFCs	748
Technical Assistance	749
Glossary	750
Feature Information for Configuring RTP Header Compression	751
Configuring TCP Header Compression	753
Contents	753
Prerequisites for Configuring TCP Header Compression	754
Information About Configuring TCP Header Compression	754
TCP Header-Compression Keywords	754
Maximum Compressed IP Header Size and TCP Header Compression	755
How to Configure TCP Header Compression	755
Enabling TCP Header Compression on an Interface	755
Enabling TCP Header Compression on an Interface That Uses Frame Relay Encapsulation	757
Restrictions	757
Changing the Maximum Size of the Compressed IP Header	758
Changing the Number of Header-Compression Connections	760
Implications of Changing the Number of Header-Compression Connections	760
Restrictions	760
Displaying Header-Compression Statistics	762
Configuration Examples for TCP Header Compression	763
Enabling TCP Header Compression on an Interface: Example	763

Enabling TCP Header Compression on an Interface That Uses Frame Relay Encapsulation: Example	763
Changing the Maximum Size of the Compressed IP Header: Example	763
Changing the Number of Header-Compression Connections: Example	764
Displaying Header-Compression Statistics: Example	764
Additional References	764
Related Documents	764
Standards	765
MIBs	765
RFCs	765
Technical Assistance	765
Glossary	766
Feature Information for Configuring TCP Header Compression	767
Configuring Class-Based RTP and TCP Header Compression	769
Contents	769
Prerequisites for Class-Based RTP and TCP Header Compression	770
Restrictions for Class-Based RTP and TCP Header Compression	770
Information About Class-Based RTP and TCP Header Compression	770
Class-Based Header Compression and the MQC	770
Benefits of Class-Based Header Compression	771
Header Compression on Local and Remote Routers	771
About Header-Compression Connections	771
How to Configure Class-Based RTP and TCP Header Compression	772
Enabling RTP or TCP Header Compression for a Class in a Policy Map	772
Attaching the Policy Map to an Interface	774
Restrictions	774
Verifying the Class-Based RTP and TCP Header Compression Configuration	775
Configuration Examples for Class-Based RTP and TCP Header Compression	776
Enabling RTP or TCP Header Compression for a Class in a Policy Map: Example	776
Attaching the Policy Map to an Interface: Example	777
Verifying the Class-Based RTP and TCP Header Compression Configuration: Example	777
Additional References	779
Related Documents	779
Standards	779
MIBs	779
RFCs	779
Technical Assistance	780
Glossary	781

Feature Information for Class-Based RTP and TCP Header Compression 781

PART 7: QUALITY OF SERVICE SOLUTIONS

IP to ATM Class of Service Overview 785

- About IP to ATM CoS 785
 - Single ATM VC Support 786
 - VC Bundle Support and Bundle Management 786
 - Per-VC LLQ, WFQ and CBWFQ Support 788
- Why Use IP to ATM CoS? 789
 - Benefits 789
- IP to ATM CoS Features 790
 - Congestion Avoidance 790
 - Bumping and ATM VC Bundles 791
 - Restrictions 792

IP to ATM Class of Service

Configuring IP to ATM Class of Service 795

- IP to ATM CoS on a Single ATM VC Configuration Task List 795
 - Defining the WRED Parameter Group 796
 - Configuring the WRED Parameter Group 796
 - Displaying the WRED Parameters 796
 - Displaying the Queueing Statistics 796
- IP to ATM CoS on an ATM Bundle Configuration Task List 797
 - Creating a VC Bundle 797
 - Applying Bundle-Level Parameters 797
 - Configuring Bundle-Level Parameters 798
 - Configuring VC Class Parameters to Apply to a Bundle 798
 - Attaching a Class to a Bundle 799
 - Committing a VC to a Bundle 799
 - Applying Parameters to Individual VCs 799
 - Configuring a VC Bundle Member Directly 800
 - Configuring VC Class Parameters to Apply to a VC Bundle Member 800
 - Applying a VC Class to a Discrete VC Bundle Member 801
 - Configuring a VC Not to Accept Bumped Traffic 801
 - Monitoring and Maintaining VC Bundles and Their VC Members 802
- Per-VC WFQ and CBWFQ Configuration Task List 802
 - Configuring Class-Based Weighted Fair Queueing 802

Attaching a Service Policy and Enabling CBWFQ for a VC	803
Configuring a VC to Use Flow-Based WFQ	803
Monitoring per-VC WFQ and CBWFQ	805
Enabling Logging of Error Messages to the Console	805
IP to ATM CoS Configuration Examples	805
Single ATM VC with WRED Group and IP Precedence Example	806
VC Bundle Configuration Using a VC Class Example	806
Bundle-Class Class	806
Control-Class Class	807
Premium-Class Class	807
Priority-Class Class	807
Basic-Class Class	807
new-york Bundle	808
san-francisco Bundle	808
los-angeles Bundle	809
Per-VC WFQ and CBWFQ on a Standalone VC Example	809
Per-VC WFQ and CBWFQ on Bundle-Member VCs Example	810
IP to ATM Class of Service Mapping for SVC Bundles	811
Feature Overview	811
Benefits	812
Restrictions	812
Related Features and Technologies	812
Related Documents	812
Supported Platforms	812
Supported Standards, MIBs, and RFCs	813
Prerequisites	813
Configuration Tasks	813
Creating an SVC Bundle	814
Configuring Bundle-Level Parameters	814
Attaching a Class to a Bundle	814
Configuring an SVC Bundle Member Directly	815
Monitoring IP to ATM Class of Service Mapping for SVC Bundles	815
Configuration Examples	816
IP to ATM Class of Service Mapping with Bundle Parameters Configured in Bundle Mode Example	816
IP to ATM Class of Service Mapping with Bundle Parameters Configured with the class-bundle Command Example	817
Command Reference	818

ATM PVC Bundle Enhancement — MPLS EXP-Based PVC Selection 819

- Feature Overview 819
 - VC Bundle Support and Bundle Management 820
 - Benefits 821
 - Restrictions 822
 - Related Features and Technologies 822
 - Related Documents 822
- Supported Platforms 822
- Supported Standards, MIBs, and RFCs 823
- Configuration Tasks 824
 - Enabling MPLS 824
 - Creating a VC Bundle 824
 - Applying Parameters to Bundles 825
 - Configuring Bundle-Level Parameters 825
 - What's Next? 825
 - Configuring a VC Bundle Member Directly 826
 - Configuring VC Class Parameters to Apply to a Bundle 826
 - Attaching a Class to a Bundle 827
 - Verifying the Configuration 827
- Configuration Examples 827
 - VC Bundle Configuration Using a VC Class Example 827
 - Bundle-Class Class 827
 - Control-Class Class 828
 - Premium-Class Class 828
 - Priority-Class Class 828
 - Basic-Class Class 829
 - new-york Bundle 829
 - san-francisco Bundle 830
 - los-angeles Bundle 830
- Command Reference 831

QoS Features for Voice

- QoS Features for Voice 835**
 - Cisco IOS QoS for Voice Features 835
 - For More Information 836

Voice and Quality of Service Features for ADSL and G.SHDSL on Cisco 1700, Cisco 2600, and Cisco 3600 Series Routers 839

- Feature Overview 840

Classification and Marking	842
Class-Based Packet Marking with Differentiated Services	842
Committed Access Rate	842
Dial-Peer DSCP and IP Precedence Marking	843
Local Policy Routing	843
Policy-Based Routing	843
Queueing and Scheduling	843
Class-Based Weighted Fair Queueing	843
Low Latency Queueing	843
Per-VC Queueing	844
Congestion Avoidance	844
Class-Based WRED with DSCP (egress)	844
Policing and Traffic Shaping	844
Class-Based Policing	844
VC Shaping for VBR-NRT	845
Link Latency	845
MLP with LFI – Bundling of VCs Across xDSL Interfaces	845
Tunable Transmission Ring	845
Other (IP QoS)	846
Access Control Lists	846
IP QoS Map to ATM CoS	846
Additional Supported Features	846
F5 OAM CC Segment Functionality	846
H.323 and Media Gateway Control Protocol	846
ILMI	847
Multiple PVC Support	847
RFC 1483 Routing	847
Benefits	847
Restrictions	847
Related Documents	848
Supported Platforms	850
Supported Standards, MIBs, and RFCs	851
Prerequisites	851
Configuration Tasks	852
Configuring the Error Duration for Digital Subscriber Line Access Multiplexers	852
Configuring the Tx Ring Limit	852
Verifying the TX Ring Limit	854
Configuration Examples	856
Differentiated Data Services over ADSL Example	856

- Verifying the Differentiated Data Services over ADSL Configuration 858
- VoIP and Data over ADSL Example 861
 - Verifying the VoIP and Data over ADSL Configuration 863
- Tx Ring-Limit Tuning over ADSL Example 865
- MLP with LFI over G.SHDSL Example 866
 - Verifying the MLP with LFI over G.SHDSL Configuration 869
- Command Reference 871

Implementing DiffServ for End-to-End Quality of Service Overview 873

- About Differentiated Services 873
 - DS Field Definition 874
 - Per-Hop Behaviors 874
 - Default PHB 875
 - Class-Selector PHB 875
 - Assured Forwarding PHB 875
 - Expedited Forwarding PHB 876
 - Benefits 876
- Differentiated Services Components 877
- Feature Sets 877
- Constructing Services Using DiffServ 878
 - Sample DiffServ Implementation 878
 - Sample Configurations 880
 - Troubleshooting Logs 886
- Class-Based Management 892
- What to Do Next 892

QoS: Classification, Policing, and Marking on LAC 893

- Contents 893
- Prerequisites for QoS: Classification, Policing, and Marking on LAC 894
- Restrictions for QoS: Classification, Policing, and Marking on LAC 894
- Information About QoS: Classification, Policing, and Marking on LAC 894
 - Benefits 895
 - QoS Policy Map 895
 - Upstream Traffic 895
 - Downstream Traffic 895
 - SSS Session 895
- How to Configure QoS: Classification, Policing, and Marking on LAC 896
 - Verifying a QoS Policy Map 896
- Configuration Examples for QoS: Classification, Policing, and Marking on LAC 896

Configuring the Routers: Example	897
Verifying the SSS Session: Example	899
Applying the QoS Policy Map: Example	900
Configuring the LAC: Example	900
Verifying the QoS Policy Map for Downstream Traffic: Example	900
Applying the QoS Policy Map to the Session: Example	901
Verifying the QoS Policy Map for Upstream Traffic: Example	902
Additional References	902
Related Documents	903
Standards	903
MIBs	903
RFCs	903
Technical Assistance	904
Command Reference	904
Glossary	905
QoS Bandwidth Estimation	907
Contents	907
Prerequisites for QoS Bandwidth Estimation	907
Restrictions for QoS Bandwidth Estimation	908
Information About QoS Bandwidth Estimation	908
Feature Overview of QoS Bandwidth Estimation	908
Applying Corvil Bandwidth	909
Benefits of QoS Bandwidth Estimation	910
How to Configure QoS Bandwidth Estimation	912
Generating a Bandwidth Estimate	912
Attaching the Policy Map to an Interface	914
Restrictions	914
Verifying the Configuration	915
Configuration Examples for QoS Bandwidth Estimation	916
Generating Bandwidth Estimates for QoS Targets: Example	916
Attaching the Policy Map to an Interface: Example	916
Verifying the Configuration: Example	916
Additional References	918
Related Documents	918
Standards	918
MIBs	918
RFCs	918
Technical Assistance	919

Command Reference 920

Glossary 921

PART 8: MODULAR QUALITY OF SERVICE COMMAND-LINE INTERFACE

Modular Quality of Service Command-Line Interface Overview 925

About the Modular QoS CLI 925

Supported MIB 927

Configuring the Modular Quality of Service Command-Line Interface 929

Modular QoS CLI Configuration Task List 929

Creating a Traffic Class 929

Creating a Traffic Policy 931

Attaching a Traffic Policy to an Interface 933

Verifying the Configuration 934

Modular QoS CLI Configuration Examples 934

Traffic Classes Defined Example 935

Traffic Policy Created Example 935

Traffic Policy Attached to an Interface Example 935

match not Command Example 936

Default Traffic Class Configuration Example 936

class-map match-any and class-map match-all Commands Example 936

Traffic Class as a Match Criterion (Nested Class Maps) Example 937

Nested Traffic Class for Maintenance Example 937

Nested Traffic Class to Combine match-any and match-all Characteristics in One Traffic Class Example 938

Traffic Policy as a QoS Policy (Hierarchical Traffic Policies) Example 938

PART 9: SECURITY DEVICE MANAGER

Security Device Manager Overview 943

About the Security Device Manager 943

PART 10: AUTOQOS

AutoQoS — VoIP 947

Contents 947

Prerequisites for AutoQoS — VoIP 948

Restrictions for AutoQoS — VoIP 948

Information About AutoQoS — VoIP 949

Benefits of AutoQoS — VoIP	949
Design Considerations	950
Configurations for the Interface Configurations, Policy Maps, Class Maps, and ACLs	951
How to Configure the AutoQoS — VoIP Feature	951
Enabling the AutoQoS — VoIP Feature	951
Prerequisites for Using the auto qos Command	952
Restrictions for Using the auto qos Command	952
FAQs and Troubleshooting Tips	954
What to Do Next	955
Verifying the Configuration	955
Configuration Examples for AutoQoS — VoIP	956
Configuring the AutoQoS — VoIP Feature Examples	956
Verifying the AutoQoS — VoIP Feature Configuration Examples	958
Additional References	961
Related Documents	961
Standards	962
MIBs	962
RFCs	962
Technical Assistance	963
Command Reference	963
AutoQoS for the Enterprise	965
Contents	965
Prerequisites for the AutoQoS for the Enterprise Feature	965
Restrictions for the AutoQoS for the Enterprise Feature	966
Information About the AutoQoS for the Enterprise Feature	967
Benefits of the AutoQoS for the Enterprise Feature	967
Design Considerations	968
Configuration Phases	969
Auto-Discovery (Data Collection) Phase	971
AutoQoS Template Generation and Installation Phase	971
How to Configure the AutoQoS for the Enterprise Feature	975
Enabling the Auto-Discovery Phase	975
Prerequisites for Using the auto discovery qos Command	976
Restrictions for Using the auto discovery qos Command	976
What to Do Next	978
Enabling the AutoQoS Template Generation and Installation Phase	978
FAQs and Troubleshooting Tips	981
What to Do Next	981

- Verifying the Configuration 981
- Configuration Examples for the AutoQoS for the Enterprise Feature 983
 - Enabling the Auto-Discovery Phase: Example 983
 - Enabling the AutoQoS Template Generation Phase: Example 983
 - Verifying the AutoQoS for the Enterprise Configuration: Example 983
- Additional References 988
 - Related Documents 988
 - Standards 989
 - MIBs 989
 - RFCs 989
 - Technical Assistance 989
- Command Reference 990



About Cisco IOS Software Documentation for Release 12.4

This chapter describes the objectives, audience, organization, and conventions of Cisco IOS software documentation. It also provides sources for obtaining documentation, technical assistance, and additional publications and information from Cisco Systems. It contains the following sections:

- [Documentation Objectives, page xlix](#)
- [Audience, page xlix](#)
- [Documentation Organization for Cisco IOS Release 12.4, page 1](#)
- [Document Conventions, page lvi](#)
- [Obtaining Documentation, page lvii](#)
- [Documentation Feedback, page lviii](#)
- [Cisco Product Security Overview, page lix](#)
- [Obtaining Technical Assistance, page lx](#)
- [Obtaining Additional Publications and Information, page lxi](#)

Documentation Objectives

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

Audience

The Cisco IOS software documentation set is intended primarily 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 software commands necessary to perform particular tasks. The Cisco IOS software 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 software release.

Documentation Organization for Cisco IOS Release 12.4

The Cisco IOS Release 12.4 documentation set consists of the configuration guide and command reference pairs listed in [Table 1](#) and the supporting documents listed in [Table 2](#). The configuration guides and command references are organized by technology. For the configuration guides:

- Some technology documentation, such as that for DHCP, contains features introduced in Releases 12.2T and 12.3T and, in some cases, Release 12.2S. To assist you in finding a particular feature, a roadmap document is provided.
- Other technology documentation, such as that for OSPF, consists of a chapter and accompanying Release 12.2T and 12.3T feature documents.



Note

In some cases, information contained in Release 12.2T and 12.3T feature documents augments or supersedes content in the accompanying documentation. Therefore it is important to review all feature documents for a particular technology.

[Table 1](#) lists the Cisco IOS Release 12.4 configuration guides and command references.

Table 1 Cisco IOS Release 12.4 Configuration Guides and Command References

Configuration Guide and Command Reference Titles	Description
IP	
Cisco IOS IP Addressing Services Configuration Guide , Release 12.4 Cisco IOS IP Addressing Services Command Reference , Release 12.4	The configuration guide is a task-oriented guide to configuring IP addressing and services, including Network Address Translation (NAT), Domain Name System (DNS), and Dynamic Host Configuration Protocol (DHCP). The command reference provides detailed information about the commands used in the configuration guide.
Cisco IOS IP Application Services Configuration Guide , Release 12.4 Cisco IOS IP Application Services Command Reference , Release 12.4	The configuration guide is a task-oriented guide to configuring IP application services, including IP access lists, Web Cache Communication Protocol (WCCP), Gateway Load Balancing Protocol (GLBP), Server Load Balancing (SLB), Hot Standby Router Protocol (HSRP), and Virtual Router Redundancy Protocol (VRRP). The command reference provides detailed information about the commands used in the configuration guide.
Cisco IOS IP Mobility Configuration Guide , Release 12.4 Cisco IOS IP Mobility Command Reference , Release 12.4	The configuration guide is a task-oriented guide to configuring Mobile IP and Cisco Mobile Networks. The command reference provides detailed information about the commands used in the configuration guide.
Cisco IOS IP Multicast Configuration Guide , Release 12.4 Cisco IOS IP Multicast Command Reference , Release 12.4	The configuration guide is a task-oriented guide to configuring IP multicast, including Protocol Independent Multicast (PIM), Internet Group Management Protocol (IGMP), Distance Vector Multicast Routing Protocol (DVMRP), and Multicast Source Discovery Protocol (MSDP). The command reference provides detailed information about the commands used in the configuration guide.
Cisco IOS IP Routing Protocols Configuration Guide , Release 12.4 Cisco IOS IP Routing Protocols Command Reference , Release 12.4	The configuration guide is a task-oriented guide to configuring IP routing protocols, including Border Gateway Protocol (BGP), Intermediate System-to-Intermediate System (IS-IS), and Open Shortest Path First (OSPF). The command reference provides detailed information about the commands used in the configuration guide.

Table 1 Cisco IOS Release 12.4 Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Description
Cisco IOS IP Switching Configuration Guide , Release 12.4 Cisco IOS IP Switching Command Reference , Release 12.4	The configuration guide is a task-oriented guide to configuring IP switching features, including Cisco Express Forwarding, fast switching, and Multicast Distributed Switching (MDS). The command reference provides detailed information about the commands used in the configuration guide.
Cisco IOS IPv6 Configuration Guide , Release 12.4 Cisco IOS IPv6 Command Reference , Release 12.4	The configuration guide is a task-oriented guide to configuring IP version 6 (IPv6), including IPv6 broadband access, IPv6 data-link layer, IPv6 multicast routing, IPv6 quality of service (QoS), IPv6 routing, IPv6 services and management, and IPv6 tunnel services. The command reference provides detailed information about the commands used in the configuration guide.
Cisco IOS Optimized Edge Routing Configuration Guide , Release 12.4 Cisco IOS Optimized Edge Routing Command Reference , Release 12.4	The configuration guide is a task-oriented guide to configuring Optimized Edge Routing (OER) features, including OER prefix learning, OER prefix monitoring, OER operational modes, and OER policy configuration. The command reference provides detailed information about the commands used in the configuration guide.
Security and VPN	
Cisco IOS Security Configuration Guide , Release 12.4 Cisco IOS Security Command Reference , Release 12.4	The configuration guide is a task-oriented guide to configuring various aspects of security, including terminal access security, network access security, accounting, traffic filters, router access, and network data encryption with router authentication. The command reference provides detailed information about the commands used in the configuration guide.
QoS	
Cisco IOS Quality of Service Solutions Configuration Guide , Release 12.4 Cisco IOS Quality of Service Solutions Command Reference , Release 12.4	The configuration guide is a task-oriented guide to configuring quality of service (QoS) features, including traffic classification and marking, traffic policing and shaping, congestion management, congestion avoidance, and signalling. The command reference provides detailed information about the commands used in the configuration guide.
LAN Switching	
Cisco IOS LAN Switching Configuration Guide , Release 12.4 Cisco IOS LAN Switching Command Reference , Release 12.4	The configuration guide is a task-oriented guide to local-area network (LAN) switching features, including configuring routing between virtual LANs (VLANs) using Inter-Switch Link (ISL) encapsulation, IEEE 802.10 encapsulation, and IEEE 802.1Q encapsulation. The command reference provides detailed information about the commands used in the configuration guide.
Multiprotocol Label Switching (MPLS)	
Cisco IOS Multiprotocol Label Switching Configuration Guide , Release 12.4 Cisco IOS Multiprotocol Label Switching Command Reference , Release 12.4	The configuration guide is a task-oriented guide to configuring Multiprotocol Label Switching (MPLS), including MPLS Label Distribution Protocol, MPLS traffic engineering, and MPLS Virtual Private Networks (VPNs). The command reference provides detailed information about the commands used in the configuration guide.
Network Management	
Cisco IOS IP SLAs Configuration Guide , Release 12.4 Cisco IOS IP SLAs Command Reference , Release 12.4	The configuration guide is a task-oriented guide to configuring the Cisco IOS IP Service Level Assurances (IP SLAs) feature. The command reference provides detailed information about the commands used in the configuration guide.

Table 1 Cisco IOS Release 12.4 Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Description
<p><i>Cisco IOS NetFlow Configuration Guide</i>, Release 12.4</p> <p><i>Cisco IOS NetFlow Command Reference</i>, Release 12.4</p>	<p>The configuration guide is a task-oriented guide to NetFlow features, including configuring NetFlow to analyze network traffic data, configuring NetFlow aggregation caches and export features, and configuring Simple Network Management Protocol (SNMP) and NetFlow MIB features. The command reference provides detailed information about the commands used in the configuration guide.</p>
<p><i>Cisco IOS Network Management Configuration Guide</i>, Release 12.4</p> <p><i>Cisco IOS Network Management Command Reference</i>, Release 12.4</p>	<p>The configuration guide is a task-oriented guide to network management features, including performing basic system management, performing troubleshooting and fault management, configuring Cisco Discovery Protocol, configuring Cisco Networking Services (CNS), configuring DistributedDirector, and configuring Simple Network Management Protocol (SNMP). The command reference provides detailed information about the commands used in the configuration guide.</p>
Voice	
<p><i>Cisco IOS Voice Configuration Library</i>, Release 12.4</p> <p><i>Cisco IOS Voice Command Reference</i>, Release 12.4</p>	<p>The configuration library is a task-oriented collection of configuration guides, application guides, a troubleshooting guide, feature documents, a library preface, a voice glossary, and more. It also covers Cisco IOS support for voice call control protocols, interoperability, physical and virtual interface management, and troubleshooting. In addition, the library includes documentation for IP telephony applications. The command reference provides detailed information about the commands used in the configuration library.</p>
Wireless/Mobility	
<p><i>Cisco IOS Mobile Wireless Gateway GPRS Support Node Configuration Guide</i>, Release 12.4</p> <p><i>Cisco IOS Mobile Wireless Gateway GPRS Support Node Command Reference</i>, Release 12.4</p>	<p>The configuration guide is a task-oriented guide to understanding and configuring a Cisco IOS Gateway GPRS Support Node (GGSN) in a 2.5G General Packet Radio Service (GPRS) and 3G Universal Mobile Telecommunication System (UMTS) network. The command reference provides detailed information about the commands used in the configuration guide.</p>
<p><i>Cisco IOS Mobile Wireless Home Agent Configuration Guide</i>, Release 12.4</p> <p><i>Cisco IOS Mobile Wireless Home Agent Command Reference</i>, Release 12.4</p>	<p>The configuration guide is a task-oriented guide to understanding and configuring the Cisco Mobile Wireless Home Agent, which is an anchor point for mobile terminals for which Mobile IP or Proxy Mobile IP services are provided. The command reference provides detailed information about the commands used in the configuration guide.</p>
<p><i>Cisco IOS Mobile Wireless Packet Data Serving Node Configuration Guide</i>, Release 12.4</p> <p><i>Cisco IOS Mobile Wireless Packet Data Serving Node Command Reference</i>, Release 12.4</p>	<p>The configuration guide is a task-oriented guide to understanding and configuring the Cisco Packet Data Serving Node (PDSN), a wireless gateway between the mobile infrastructure and standard IP networks that enables packet data services in a Code Division Multiple Access (CDMA) environment. The command reference provides detailed information about the commands used in the configuration guide.</p>

Table 1 Cisco IOS Release 12.4 Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Description
<i>Cisco IOS Mobile Wireless Radio Access Networking Configuration Guide</i> , Release 12.4 <i>Cisco IOS Mobile Wireless Radio Access Networking Command Reference</i> , Release 12.4	The configuration guide is a task-oriented guide to understanding and configuring Cisco IOS Radio Access Network products. The command reference provides detailed information about the commands used in the configuration guide.
Long Reach Ethernet (LRE) and Digital Subscriber Line (xDSL)	
<i>Cisco IOS Broadband and DSL Configuration Guide</i> , Release 12.4 <i>Cisco IOS Broadband and DSL Command Reference</i> , Release 12.4	The configuration guide is a task-oriented guide to configuring broadband access aggregation and digital subscriber line features. The command reference provides detailed information about the commands used in the configuration guide.
<i>Cisco IOS Service Selection Gateway Configuration Guide</i> , Release 12.4 <i>Cisco IOS Service Selection Gateway Command Reference</i> , Release 12.4	The configuration guide is a task-oriented guide to configuring Service Selection Gateway (SSG) features, including subscriber authentication, service access, and accounting. The command reference provides detailed information about the commands used in the configuration guide.
Dial—Access	
<i>Cisco IOS Dial Technologies Configuration Guide</i> , Release 12.4 <i>Cisco IOS Dial Technologies Command Reference</i> , Release 12.4	The configuration guide is a task-oriented guide to configuring lines, modems, and ISDN services. This guide also contains information about configuring dialup solutions, including solutions for remote sites dialing in to a central office, Internet service providers (ISPs), ISP customers at home offices, enterprise WAN system administrators implementing dial-on-demand routing, and other corporate environments. The command reference provides detailed information about the commands used in the configuration guide.
<i>Cisco IOS VPDN Configuration Guide</i> , Release 12.4 <i>Cisco IOS VPDN Command Reference</i> , Release 12.4	The configuration guide is a task-oriented guide to configuring Virtual Private Dialup Networks (VPDNs), including information about Layer 2 tunneling protocols, client-initiated VPDN tunneling, NAS-initiated VPDN tunneling, and multihop VPDN. The command reference provides detailed information about the commands used in the configuration guide.
Asynchronous Transfer Mode (ATM)	
<i>Cisco IOS Asynchronous Transfer Mode Configuration Guide</i> , Release 12.4 <i>Cisco IOS Asynchronous Transfer Mode Command Reference</i> , Release 12.4	The configuration guide is a task-oriented guide to configuring Asynchronous Transfer Mode (ATM), including WAN ATM, LAN ATM, and multiprotocol over ATM (MPOA). The command reference provides detailed information about the commands used in the configuration guide.
WAN	
<i>Cisco IOS Wide-Area Networking Configuration Guide</i> , Release 12.4 <i>Cisco IOS Wide-Area Networking Command Reference</i> , Release 12.4	The configuration guide is a task-oriented guide to configuring wide-area network (WAN) features, including Layer 2 Tunneling Protocol Version 3 (L2TPv3); Frame Relay; Link Access Procedure, Balanced (LAPB); and X.25. The command reference provides detailed information about the commands used in the configuration guide.

Table 1 Cisco IOS Release 12.4 Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Description
System Management	
<p>Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.4</p> <p>Cisco IOS Configuration Fundamentals Command Reference, Release 12.4</p>	<p>The configuration guide is a task-oriented guide to using Cisco IOS software to configure and maintain Cisco routers and access servers, including information about using the Cisco IOS command-line interface (CLI), loading and maintaining system images, using the Cisco IOS file system, using the Cisco IOS Web browser user interface (UI), and configuring basic file transfer services. The command reference provides detailed information about the commands used in the configuration guide.</p>
<p>Cisco IOS Interface and Hardware Component Configuration Guide, Release 12.4</p> <p>Cisco IOS Interface and Hardware Component Command Reference, Release 12.4</p>	<p>The configuration guide is a task-oriented guide to configuring and managing interfaces and hardware components, including dial shelves, LAN interfaces, logical interfaces, serial interfaces, and virtual interfaces. The command reference provides detailed information about the commands used in the configuration guide.</p>
IBM Technologies	
<p>Cisco IOS Bridging and IBM Networking Configuration Guide, Release 12.4</p> <p>Cisco IOS Bridging Command Reference, Release 12.4</p> <p>Cisco IOS IBM Networking Command Reference, Release 12.4</p>	<p>The configuration guide is a task-oriented guide to configuring:</p> <ul style="list-style-type: none"> • Bridging features, including transparent and source-route transparent (SRT) bridging, source-route bridging (SRB), Token Ring Inter-Switch Link (TRISL), and Token Ring Route Switch Module (TRRSM). • IBM network features, including data-link switching plus (DLSw+), serial tunnel (STUN), and block serial tunnel (BSTUN); Logical Link Control, type 2 (LLC2), and Synchronous Data Link Control (SDLC); IBM Network Media Translation, including SDLC Logical Link Control (SDLLC) and Qualified Logical Link Control (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. <p>The two command references provide detailed information about the commands used in the configuration guide.</p>
Additional and Legacy Protocols	
<p>Cisco IOS AppleTalk Configuration Guide, Release 12.4</p> <p>Cisco IOS AppleTalk Command Reference, Release 12.4</p>	<p>The configuration guide is a task-oriented guide to configuring the AppleTalk protocol. The command reference provides detailed information about the commands used in the configuration guide.</p>
<p>Cisco IOS DECnet Configuration Guide, Release 12.4</p> <p>Cisco IOS DECnet Command Reference, Release 12.4</p>	<p>The configuration guide is a task-oriented guide to configuring the DECnet protocol. The command reference provides detailed information about the commands used in the configuration guide.</p>
<p>Cisco IOS ISO CLNS Configuration Guide, Release 12.4</p> <p>Cisco IOS ISO CLNS Command Reference, Release 12.4</p>	<p>The configuration guide is a task-oriented guide to configuring International Organization for Standardization (ISO) Connectionless Network Service (CLNS). The command reference provides detailed information about the commands used in the configuration guide.</p>

Table 1 Cisco IOS Release 12.4 Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Description
<i>Cisco IOS Novell IPX Configuration Guide</i> , Release 12.4 <i>Cisco IOS Novell IPX Command Reference</i> , Release 12.4	The configuration guide is a task-oriented guide to configuring the Novell Internetwork Packet Exchange (IPX) protocol. The command reference provides detailed information about the commands used in the configuration guide.
<i>Cisco IOS Terminal Services Configuration Guide</i> , Release 12.4 <i>Cisco IOS Terminal Services Command Reference</i> , Release 12.4	The configuration guide is a task-oriented guide to configuring terminal services, including DEC, local-area transport (LAT), and X.25 packet assembler/disassembler (PAD). The command reference provides detailed information about the commands used in the configuration guide.

Table 2 lists the documents and resources that support the Cisco IOS Release 12.4 software configuration guides and command references.

Table 2 Cisco IOS Release 12.4 Supporting Documents and Resources

Document Title	Description
<i>Cisco IOS Master Commands List</i> , Release 12.4	An alphabetical listing of all the commands documented in the Cisco IOS Release 12.4 command references.
<i>Cisco IOS New, Modified, Replaced, and Removed Commands</i> , Release 12.4	A listing of all the new, modified, replaced and removed commands since Cisco IOS Release 12.3, grouped by Release 12.3T maintenance release and ordered alphabetically within each group.
<i>Cisco IOS New and Modified Commands</i> , Release 12.3	A listing of all the new, modified, and replaced commands since Cisco IOS Release 12.2, grouped by Release 12.2T maintenance release and ordered alphabetically within each group.
<i>Cisco IOS System Messages, Volume 1 of 2</i> <i>Cisco IOS System Messages, Volume 2 of 2</i>	Listings and descriptions of Cisco IOS system messages. Not all system messages indicate problems with your system. Some are purely informational, and others may help diagnose problems with communications lines, internal hardware, or the system software.
<i>Cisco IOS Debug Command Reference</i> , Release 12.4	An alphabetical listing of the debug commands and their descriptions. Documentation for each command includes a brief description of its use, command syntax, and usage guidelines.
<i>Release Notes</i> , Release 12.4	A description of general release information, including information about supported platforms, feature sets, platform-specific notes, and Cisco IOS software defects.
<i>Internetworking Terms and Acronyms</i>	Compilation and definitions of the terms and acronyms used in the internetworking industry.

Table 2 Cisco IOS Release 12.4 Supporting Documents and Resources (continued)

Document Title	Description
RFCs	RFCs are standards documents maintained by the Internet Engineering Task Force (IETF). Cisco IOS software documentation references supported RFCs when applicable. The full text of referenced RFCs may be obtained at the following URL: http://www.rfc-editor.org/
MIBs	MIBs are used for network monitoring. 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

Document Conventions

Within Cisco IOS software documentation, the term *router* is generally used to refer to a variety of Cisco products (for example, routers, access servers, and switches). Routers, access servers, and other networking devices that support Cisco IOS software are shown interchangeably within examples. These products are used only for illustrative purposes; that is, an example that shows one product does not necessarily indicate that other products are not supported.

The Cisco IOS documentation set uses the following conventions:

Convention	Description
^ or Ctrl	The ^ and Ctrl symbols represent the Control key. For example, the key combination ^D or Ctrl-D means hold down the Control key while you press the D key. Keys are indicated in capital letters but are not case sensitive.
<i>string</i>	A string is a nonquoted set of characters shown in italics. For example, when setting an SNMP community string to <i>public</i> , do not use quotation marks around the string or the string will include the quotation marks.

Command syntax descriptions use the following conventions:

Convention	Description
bold	Bold text indicates commands and keywords that you enter literally as shown.
<i>italics</i>	Italic text indicates arguments for which you supply values.
[x]	Square brackets enclose an optional element (keyword or argument).
	A vertical line indicates a choice within an optional or required set of keywords or arguments.
[x y]	Square brackets enclosing keywords or arguments separated by a vertical line indicate an optional choice.
{x y}	Braces enclosing keywords or arguments separated by a vertical line indicate a required choice.

Nested sets of square brackets or braces indicate optional or required choices within optional or required elements. For example:

Convention	Description
[x {y z}]	Braces and a vertical line within square brackets indicate a required choice within an optional element.

Examples use the following conventions:

Convention	Description
screen	Examples of information displayed on the screen are set in Courier font.
bold screen	Examples of text that you must enter are set in Courier bold font.
< >	Angle brackets enclose text that is not printed to the screen, such as passwords, and are used in contexts in which the italic document convention is not available, such as ASCII text.
!	An exclamation point at the beginning of a line indicates a comment line. (Exclamation points are also displayed by the Cisco IOS software for certain processes.)
[]	Square brackets enclose default responses to system prompts.

The following conventions are used to attract the attention of the reader:



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 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.

Obtaining Documentation

Cisco documentation and additional literature are available on Cisco.com. Cisco also provides several ways to obtain technical assistance and other technical resources. These sections explain how to obtain technical information from Cisco Systems.

Cisco.com

You can access the most current Cisco documentation and technical support at this URL:

<http://www.cisco.com/techsupport>

You can access the Cisco website at this URL:

<http://www.cisco.com>

You can access international Cisco websites at this URL:

http://www.cisco.com/public/countries_languages.shtml

Product Documentation DVD

Cisco documentation and additional literature are available in the Product Documentation DVD package, which may have shipped with your product. The Product Documentation DVD is updated regularly and may be more current than printed documentation.

The Product Documentation DVD is a comprehensive library of technical product documentation on portable media. The DVD enables you to access multiple versions of hardware and software installation, configuration, and command guides for Cisco products and to view technical documentation in HTML. With the DVD, you have access to the same documentation that is found on the Cisco website without being connected to the Internet. Certain products also have .pdf versions of the documentation available.

The Product Documentation DVD is available as a single unit or as a subscription. Registered Cisco.com users (Cisco direct customers) can order a Product Documentation DVD (product number DOC-DOCDVD=) from Cisco Marketplace at this URL:

<http://www.cisco.com/go/marketplace/>

Ordering Documentation

Beginning June 30, 2005, registered Cisco.com users may order Cisco documentation at the Product Documentation Store in the Cisco Marketplace at this URL:

<http://www.cisco.com/go/marketplace/>

Nonregistered Cisco.com users can order technical documentation from 8:00 a.m. to 5:00 p.m. (0800 to 1700) PDT by calling 1 866 463-3487 in the United States and Canada, or elsewhere by calling 011 408 519-5055. You can also order documentation by e-mail at tech-doc-store-mkpl@external.cisco.com or by fax at 1 408 519-5001 in the United States and Canada, or elsewhere at 011 408 519-5001.

Documentation Feedback

You can rate and provide feedback about Cisco technical documents by completing the online feedback form that appears with the technical documents on Cisco.com.

You can send comments about Cisco documentation to bug-doc@cisco.com.

You can submit comments by using the response card (if present) behind the front cover of your document or by writing to the following address:

Cisco Systems
Attn: Customer Document Ordering
170 West Tasman Drive
San Jose, CA 95134-9883

We appreciate your comments.

Cisco Product Security Overview

Cisco provides a free online Security Vulnerability Policy portal at this URL:

http://www.cisco.com/en/US/products/products_security_vulnerability_policy.html

From this site, you can perform these tasks:

- Report security vulnerabilities in Cisco products.
- Obtain assistance with security incidents that involve Cisco products.
- Register to receive security information from Cisco.

A current list of security advisories and notices for Cisco products is available at this URL:

<http://www.cisco.com/go/psirt>

If you prefer to see advisories and notices as they are updated in real time, you can access a Product Security Incident Response Team Really Simple Syndication (PSIRT RSS) feed from this URL:

http://www.cisco.com/en/US/products/products_psirt_rss_feed.html

Reporting Security Problems in Cisco Products

Cisco is committed to delivering secure products. We test our products internally before we release them, and we strive to correct all vulnerabilities quickly. If you think that you might have identified a vulnerability in a Cisco product, contact PSIRT:

- Emergencies—security-alert@cisco.com

An emergency is either a condition in which a system is under active attack or a condition for which a severe and urgent security vulnerability should be reported. All other conditions are considered nonemergencies.

- Nonemergencies—psirt@cisco.com

In an emergency, you can also reach PSIRT by telephone:

- 1 877 228-7302
- 1 408 525-6532



Tip

We encourage you to use Pretty Good Privacy (PGP) or a compatible product to encrypt any sensitive information that you send to Cisco. PSIRT can work from encrypted information that is compatible with PGP versions 2.x through 8.x.

Never use a revoked or an expired encryption key. The correct public key to use in your correspondence with PSIRT is the one linked in the Contact Summary section of the Security Vulnerability Policy page at this URL:

http://www.cisco.com/en/US/products/products_security_vulnerability_policy.html

The link on this page has the current PGP key ID in use.

Obtaining Technical Assistance

Cisco Technical Support provides 24-hour-a-day award-winning technical assistance. The Cisco Technical Support & Documentation website on Cisco.com features extensive online support resources. In addition, if you have a valid Cisco service contract, Cisco Technical Assistance Center (TAC) engineers provide telephone support. If you do not have a valid Cisco service contract, contact your reseller.

Cisco Technical Support & Documentation Website

The Cisco Technical Support & Documentation website provides online documents and tools for troubleshooting and resolving technical issues with Cisco products and technologies. The website is available 24 hours a day, at this URL:

<http://www.cisco.com/techsupport>

Access to all tools on the Cisco Technical Support & Documentation website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a user ID or password, you can register at this URL:

<http://tools.cisco.com/RPF/register/register.do>



Note

Use the Cisco Product Identification (CPI) tool to locate your product serial number before submitting a web or phone request for service. You can access the CPI tool from the Cisco Technical Support & Documentation website by clicking the **Tools & Resources** link. Choose **Cisco Product Identification Tool** from the Alphabetical Index drop-down list, or click the **Cisco Product Identification Tool** link under Alerts & RMAs. The CPI tool offers three search options: by product ID or model name; by tree view; or for certain products, by copying and pasting **show** command output. Search results show an illustration of your product with the serial number label location highlighted. Locate the serial number label on your product and record the information before placing a service call.

Submitting a Service Request

Using the online TAC Service Request Tool is the fastest way to open S3 and S4 service requests. (S3 and S4 service requests are those in which your network is minimally impaired or for which you require product information.) After you describe your situation, the TAC Service Request Tool provides recommended solutions. If your issue is not resolved using the recommended resources, your service request is assigned to a Cisco engineer. The TAC Service Request Tool is located at this URL:

<http://www.cisco.com/techsupport/servicerequest>

For S1 or S2 service requests or if you do not have Internet access, contact the Cisco TAC by telephone. (S1 or S2 service requests are those in which your production network is down or severely degraded.) Cisco engineers are assigned immediately to S1 and S2 service requests to help keep your business operations running smoothly.

To open a service request by telephone, use one of the following numbers:

Asia-Pacific: +61 2 8446 7411 (Australia: 1 800 805 227)

EMEA: +32 2 704 55 55

USA: 1 800 553-2447

For a complete list of Cisco TAC contacts, go to this URL:

<http://www.cisco.com/techsupport/contacts>

Definitions of Service Request Severity

To ensure that all service requests are reported in a standard format, Cisco has established severity definitions.

Severity 1 (S1)—Your network is “down,” or there is a critical impact to your business operations. You and Cisco will commit all necessary resources around the clock to resolve the situation.

Severity 2 (S2)—Operation of an existing network is severely degraded, or significant aspects of your business operation are negatively affected by inadequate performance of Cisco products. You and Cisco will commit full-time resources during normal business hours to resolve the situation.

Severity 3 (S3)—Operational performance of your network is impaired, but most business operations remain functional. You and Cisco will commit resources during normal business hours to restore service to satisfactory levels.

Severity 4 (S4)—You require information or assistance with Cisco product capabilities, installation, or configuration. There is little or no effect on your business operations.

Obtaining Additional Publications and Information

Information about Cisco products, technologies, and network solutions is available from various online and printed sources.

- Cisco Marketplace provides a variety of Cisco books, reference guides, documentation, and logo merchandise. Visit Cisco Marketplace, the company store, at this URL:
<http://www.cisco.com/go/marketplace/>
- *Cisco Press* publishes a wide range of general networking, training and certification titles. Both new and experienced users will benefit from these publications. For current Cisco Press titles and other information, go to Cisco Press at this URL:
<http://www.ciscopress.com>
- *Packet* magazine is the Cisco Systems technical user magazine for maximizing Internet and networking investments. Each quarter, Packet delivers coverage of the latest industry trends, technology breakthroughs, and Cisco products and solutions, as well as network deployment and troubleshooting tips, configuration examples, customer case studies, certification and training information, and links to scores of in-depth online resources. You can access Packet magazine at this URL:
<http://www.cisco.com/packet>
- *iQ Magazine* is the quarterly publication from Cisco Systems designed to help growing companies learn how they can use technology to increase revenue, streamline their business, and expand services. The publication identifies the challenges facing these companies and the technologies to help solve them, using real-world case studies and business strategies to help readers make sound technology investment decisions. You can access iQ Magazine at this URL:

<http://www.cisco.com/go/iqmagazine>

or view the digital edition at this URL:

<http://ciscoiq.texterity.com/ciscoiq/sample/>

- *Internet Protocol Journal* is a quarterly journal published by Cisco Systems for engineering professionals involved in designing, developing, and operating public and private internets and intranets. You can access the *Internet Protocol Journal* at this URL:
<http://www.cisco.com/ipj>
- Networking products offered by Cisco Systems, as well as customer support services, can be obtained at this URL:
<http://www.cisco.com/en/US/products/index.html>
- Networking Professionals Connection is an interactive website for networking professionals to share questions, suggestions, and information about networking products and technologies with Cisco experts and other networking professionals. Join a discussion at this URL:
<http://www.cisco.com/discuss/networking>
- World-class networking training is available from Cisco. You can view current offerings at this URL:
<http://www.cisco.com/en/US/learning/index.html>



Using Cisco IOS Software for Release 12.4

This chapter provides tips for understanding and configuring Cisco IOS software using the command-line interface (CLI). It contains the following sections:

- [Understanding Command Modes, page lxiii](#)
- [Getting Help, page lxiv](#)
- [Using the no and default Forms of Commands, page lxviii](#)
- [Saving Configuration Changes, page lxviii](#)
- [Filtering Output from the show and more Commands, page lxix](#)
- [Finding Additional Feature Support Information, page lxix](#)

For an overview of Cisco IOS software configuration, see the [Cisco IOS Configuration Fundamentals Configuration Guide](#).

For information on the conventions used in the Cisco IOS software documentation set, see the “[About Cisco IOS Software Documentation for Release 12.4](#)” chapter.

Understanding Command Modes

You use the CLI to access Cisco IOS software. Because the CLI is divided into many different modes, the commands available to you at any given time depend on the mode that you are currently in. Entering a question mark (?) at the CLI prompt allows you to obtain a list of commands available for each command mode.

When you log in to a Cisco device, the device is initially in user EXEC mode. User EXEC mode contains only a limited subset of commands. To have access to all commands, you must enter privileged EXEC mode by entering the **enable** command and a password (when required). From privileged EXEC mode you have access to both user EXEC and privileged EXEC commands. Most EXEC commands are used independently to observe status or to perform a specific function. For example, **show** commands are used to display important status information, and **clear** commands allow you to reset counters or interfaces. The EXEC commands are not saved when the software reboots.

Configuration modes allow you to make changes to the running configuration. If you later save the running configuration to the startup configuration, these changed commands are stored when the software is rebooted. To enter specific configuration modes, you must start at global configuration mode. From global configuration mode, you can enter interface configuration mode and a variety of other modes, such as protocol-specific modes.

ROM monitor mode is a separate mode used when the Cisco IOS 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.

Table 1 describes how to access and exit various common command modes of the Cisco IOS software. It also shows examples of the prompts displayed for each mode.

Table 1 Accessing and Exiting Command Modes

Command Mode	Access Method	Prompt	Exit Method
User EXEC	Log in.	Router>	Use the logout command.
Privileged EXEC	From user EXEC mode, use the enable command.	Router#	To return to user EXEC mode, use the disable command.
Global configuration	From privileged EXEC mode, use the configure terminal command.	Router(config)#	To return to privileged EXEC mode from global configuration mode, use the exit or end command.
Interface configuration	From global configuration mode, specify an interface using an interface command.	Router(config-if)#	To return to global configuration mode, use the exit command. To return to privileged EXEC mode, use the end command.
ROM monitor	From privileged EXEC mode, use the reload command. Press the Break key during the first 60 seconds while the system is booting.	>	To exit ROM monitor mode, use the continue command.

For more information on command modes, see the “Using the Cisco IOS Command-Line Interface” chapter in the *Cisco IOS Configuration Fundamentals Configuration Guide*.

Getting Help

Entering a question mark (?) at the CLI prompt displays a list of commands available for each command mode. You can also get a list of keywords and arguments associated with any command by using the context-sensitive help feature.

To get help specific to a command mode, a command, a keyword, or an argument, use one of the following commands:

Command	Purpose
help	Provides a brief description of the help system in any command mode.
<i>abbreviated-command-entry?</i>	Provides a list of commands that begin with a particular character string. (No space between command and question mark.)
<i>abbreviated-command-entry<Tab></i>	Completes a partial command name.

Command	Purpose
?	Lists all commands available for a particular command mode.
<i>command</i> ?	Lists the keywords or arguments that you must enter next on the command line. (Space between command and question mark.)

Example: How to Find Command Options

This section provides an example of how to display syntax for a command. The syntax can consist of optional or required keywords and arguments. To display keywords and arguments for a command, enter a question mark (?) at the configuration prompt or after entering part of a command followed by a space. The Cisco IOS software displays a list and brief description of available keywords and arguments. For example, if you were in global configuration mode and wanted to see all the keywords or arguments for the **arap** command, you would type **arap ?**.

The <cr> symbol in command help output stands for “carriage return.” On older keyboards, the carriage return key is the Return key. On most modern keyboards, the carriage return key is the Enter key. The <cr> symbol at the end of command help output indicates that you have the option to press **Enter** to complete the command and that the arguments and keywords in the list preceding the <cr> symbol are optional. The <cr> symbol by itself indicates that no more arguments or keywords are available and that you must press **Enter** to complete the command.

[Table 2](#) shows examples of how you can use the question mark (?) to assist you in entering commands. The table steps you through configuring an IP address on a serial interface on a Cisco 7206 router that is running Cisco IOS Release 12.0(3).

Table 2 How to Find Command Options

Command	Comment
Router> enable Password: <password> Router#	Enter the enable command and password to access privileged EXEC commands. You are in privileged EXEC mode when the prompt changes to Router#.
Router# configure terminal Enter configuration commands, one per line. End with CNTL/Z. Router(config)#	Enter the configure terminal privileged EXEC command to enter global configuration mode. You are in global configuration mode when the prompt changes to Router(config)#.

Table 2 How to Find Command Options (continued)

Command	Comment
<pre>Router(config)# interface serial ? <0-6> Serial interface number Router(config)# interface serial 4 ? / Router(config)# interface serial 4/ ? <0-3> Serial interface number Router(config)# interface serial 4/0 ? <cr> Router(config)# interface serial 4/0 Router(config-if)#</pre>	<p>Enter interface configuration mode by specifying the serial interface that you want to configure using the interface serial global configuration command.</p> <p>Enter ? to display what you must enter next on the command line. In this example, you must enter the serial interface slot number and port number, separated by a forward slash.</p> <p>When the <cr> symbol is displayed, you can press Enter to complete the command.</p> <p>You are in interface configuration mode when the prompt changes to Router(config-if)#.</p>
<pre>Router(config-if)# ? Interface configuration commands: . . . ip Interface Internet Protocol config commands keepalive Enable keepalive lan-name LAN Name command llc2 LLC2 Interface Subcommands load-interval Specify interval for load calculation for an interface locaddr-priority Assign a priority group logging Configure logging for interface loopback Configure internal loopback on an interface mac-address Manually set interface MAC address mls mls router sub/interface commands mpoa MPOA interface configuration commands mtu Set the interface Maximum Transmission Unit (MTU) netbios Use a defined NETBIOS access list or enable name-caching no Negate a command or set its defaults nrzi-encoding Enable use of NRZI encoding ntp Configure NTP . . . Router(config-if)#</pre>	<p>Enter ? to display a list of all the interface configuration commands available for the serial interface. This example shows only some of the available interface configuration commands.</p>

Table 2 How to Find Command Options (continued)

Command	Comment
<pre>Router(config-if)# ip ? Interface IP configuration subcommands: access-group Specify access control for packets accounting Enable IP accounting on this interface address Set the IP address of an interface authentication authentication subcommands bandwidth-percent Set EIGRP bandwidth limit broadcast-address Set the broadcast address of an interface cgmp Enable/disable CGMP directed-broadcast Enable forwarding of directed broadcasts dvmrp DVMRP interface commands hello-interval Configures IP-EIGRP hello interval helper-address Specify a destination address for UDP broadcasts hold-time Configures IP-EIGRP hold time . . . Router(config-if)# ip</pre>	<p>Enter the command that you want to configure for the interface. This example uses the ip command.</p> <p>Enter ? to display what you must enter next on the command line. This example shows only some of the available interface IP configuration commands.</p>
<pre>Router(config-if)# ip address ? A.B.C.D IP address negotiated IP Address negotiated over PPP Router(config-if)# ip address</pre>	<p>Enter the command that you want to configure for the interface. This example uses the ip address command.</p> <p>Enter ? to display what you must enter next on the command line. In this example, you must enter an IP address or the negotiated keyword.</p> <p>A carriage return (<cr>) is not displayed; therefore, you must enter additional keywords or arguments to complete the command.</p>
<pre>Router(config-if)# ip address 172.16.0.1 ? A.B.C.D IP subnet mask Router(config-if)# ip address 172.16.0.1</pre>	<p>Enter the keyword or argument that you want to use. This example uses the 172.16.0.1 IP address.</p> <p>Enter ? to display what you must enter next on the command line. In this example, you must enter an IP subnet mask.</p> <p>A <cr> is not displayed; therefore, you must enter additional keywords or arguments to complete the command.</p>

Table 2 How to Find Command Options (continued)

Command	Comment
Router(config-if)# ip address 172.16.0.1 255.255.255.0 ? secondary Make this IP address a secondary address <cr>	Enter the IP subnet mask. This example uses the 255.255.255.0 IP subnet mask.
Router(config-if)# ip address 172.16.0.1 255.255.255.0	Enter ? to display what you must enter next on the command line. In this example, you can enter the secondary keyword, or you can press Enter . A <cr> is displayed; you can press Enter to complete the command, or you can enter another keyword.
Router(config-if)# ip address 172.16.0.1 255.255.255.0 Router(config-if)#	In this example, Enter is pressed to complete the command.

Using the no and default Forms of Commands

Almost every configuration command has a **no** form. In general, use the **no** form to disable a function. Use the command without the **no** keyword to reenable a disabled function or to enable a function that is disabled by default. For example, IP routing is enabled by default. To disable IP routing, use the **no ip routing** command; to reenable IP routing, use the **ip routing** command. The Cisco IOS software command reference publications provide the complete syntax for the configuration commands and describe what the **no** form of a command does.

Configuration commands can also have a **default** form, which returns the command settings to the default values. Most commands are disabled by default, so in such cases using the **default** form has the same result as using the **no** form of the command. However, some commands are enabled by default and have variables set to certain default values. In these cases, the **default** form of the command enables the command and sets the variables to their default values. The Cisco IOS software command reference publications describe the effect of the **default** form of a command if the command functions differently than the **no** form.

Saving Configuration Changes

Use the **copy system:running-config nvram:startup-config** command or the **copy running-config startup-config** command to save your configuration changes to the startup configuration so that the changes will not be lost if the software reloads or a power outage occurs. For example:

```
Router# copy system:running-config nvram:startup-config
Building configuration...
```

It might take a minute or two to save the configuration. After the configuration has been saved, the following output appears:

```
[OK]
Router#
```

On most platforms, this task saves the configuration to NVRAM. On the Class A flash file system platforms, this task saves the configuration to the location specified by the CONFIG_FILE environment variable. The CONFIG_FILE variable defaults to NVRAM.

Filtering Output from the show and more Commands

You can search and filter the output of **show** and **more** commands. This functionality is useful if you need to sort through large amounts of output or if you want to exclude output that you need not see.

To use this functionality, enter a **show** or **more** command followed by the “pipe” character (`|`); one of the keywords **begin**, **include**, or **exclude**; and a regular expression on which you want to search or filter (the expression is case-sensitive):

```
command | {begin | include | exclude} regular-expression
```

The output matches certain lines of information in the configuration file. The following example illustrates how to use output modifiers with the **show interface** command when you want the output to include only lines in which the expression “protocol” appears:

```
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
```

For more information on the search and filter functionality, see the “Using the Cisco IOS Command-Line Interface” chapter in the *Cisco IOS Configuration Fundamentals Configuration Guide*.

Finding Additional Feature Support Information

If you want to use a specific Cisco IOS software feature, you will need to determine in which Cisco IOS software images that feature is supported. Feature support in Cisco IOS software images depends on three main factors: the software version (called the “Release”), the hardware model (the “Platform” or “Series”), and the “Feature Set” (collection of specific features designed for a certain network environment). Although the Cisco IOS software documentation set documents feature support information for Release 12.4 as a whole, it does not generally provide specific hardware and feature set information.

To determine the correct combination of Release (software version), Platform (hardware version), and Feature Set needed to run a particular feature (or any combination of features), use Feature Navigator.

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.

Software features may also have additional limitations or restrictions. For example, a minimum amount of system memory may be required. Or there may be known issues for features on certain platforms that have not yet been resolved (called “Caveats”). For the latest information about these limitations, see the release notes for the appropriate Cisco IOS software release. Release notes provide detailed installation instructions, new feature descriptions, system requirements, limitations and restrictions, caveats, and troubleshooting information for a particular software release.



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?](#)
- [About QoS Architecture](#)
- [Who Could Benefit from Using Cisco IOS QoS?](#)
- [Why Deploy Cisco IOS QoS?](#)
- [End-to-End QoS Models](#)
- [Cisco IOS QoS Features](#)

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

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 signalling 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 signalling; 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 [“Implementing DiffServ for End-to-End Quality of Service Overview”](#) 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](#)
- [signalling](#)
- [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 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 in this book:

- [“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](#), Release 12.4.

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 “[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 signalling traffic and a PVC transporting signalling 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](#), Release 12.4.

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.

signalling

Cisco IOS QoS signalling provides a way for an end station or network node to 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.

Cisco IOS QoS signalling takes advantage of IP. Either in-band (IP Precedence, 802.1p) or out-of-band (RSVP) signalling 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 signalling: IP Precedence signals for differentiated QoS, and RSVP signals for guaranteed QoS.

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

- ATM User Network Interface (UNI) signalling and Frame Relay Local Management Interface (LMI)
Achieves the end-to-end benefits of IP Precedence and RSVP signalling, and provides signalling into their respective backbone technologies.
- Common Open Policy Service (COPS) with RSVP
Achieves centralized monitoring and control of RSVP signalling.

- 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 signalling, see the following chapters in this book:

- [“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*, Release 12.4.

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 information about MLP, see the [“Reducing Latency and Jitter for Real-Time Traffic Using Multilink PPP”](#) chapter.

Frame Relay Fragmentation

Cisco has developed the following three methods of performing Frame Relay fragmentation:

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

For more information about Frame Relay fragmentation, see the [Cisco IOS Wide-Area Networking Configuration Guide](#), Release 12.4.

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 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 [“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 [“Implementing DiffServ for End-to-End Quality of Service Overview”](#) chapter.

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](#) feature module, Cisco IOS Release 12.3(14)T.

Modular QoS Command-Line Interface

The Modular CLI is a CLI structure that allows users to create traffic policies and attach these policies to interfaces. For conceptual information about the Modular QoS CLI, see the [“Modular Quality of Service Command-Line Interface Overview](#) chapter.

The Modular QoS CLI contains the following three steps:

1. Define a traffic class with the **class-map** command.
2. Create a traffic policy by associating the traffic class with one or more QoS features (using the **policy-map** command).
3. Attach the traffic policy to the interface with the **service-policy** command.

For information on how to configure the Modular QoS CLI, see the [“Configuring the Modular Quality of Service Command-Line Interface”](#) chapter.

Security Device Manager

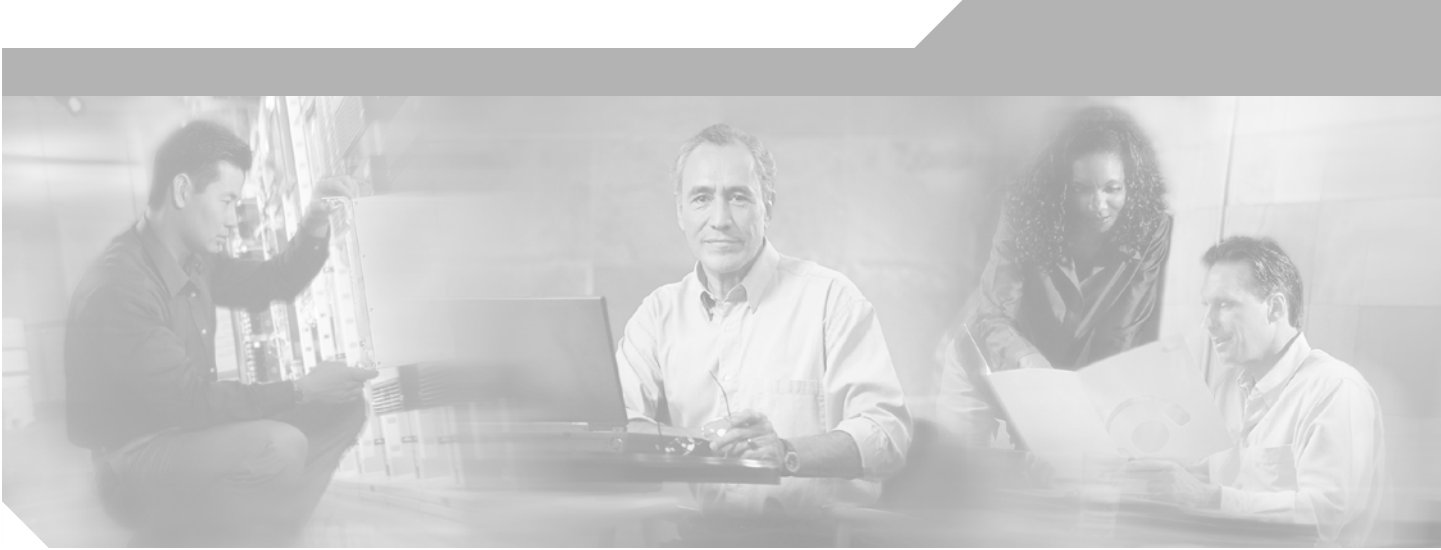
The Cisco Router and Security Device Manager (SDM) provides an intuitive, graphical user interface for configuring and monitoring advanced IP-based QoS functionality within Cisco routers.

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”](#) chapter.



Part 1: 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 a 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 others 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 in this book.

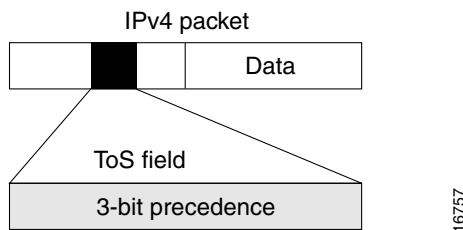
This chapter explains IP Precedence, then it gives a brief description of the kinds of traffic classification provided by the Cisco IOS QoS features. It discusses features described in the following sections:

- [Policy-Based Routing](#)
- [QoS Policy Propagation via Border Gateway Protocol](#)
- [Committed Access Rate](#)
- [Marking Network Traffic](#)
- [Network-Based Application Recognition](#)

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.

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 in regard to the ToS to grant it. 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 up to six classes—the remaining two are reserved for internal network use—and then use policy maps and extended access lists to define network policies in terms of 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 the RFC 791 document. [Table 3](#) lists the numbers and their corresponding names, from least to most important.

Table 3 *IP Precedence Values*

Number	Name
0	routine
1	priority
2	immediate
3	flash
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 stipulating 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 coming into your network can have precedence set by outside devices, we recommend you reset the precedence for all traffic entering 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 any of the features described in the following sections to set the IP Precedence in packets:

- [Policy-Based Routing](#)
- [QoS Policy Propagation via Border Gateway Protocol](#)
- [Committed Access Rate](#)

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.

Policy-Based Routing

PBR gives you a flexible means of routing packets by allowing you to configure a defined policy for traffic flows, lessening reliance on routes derived from routing protocols. To this end, PBR gives you more control over routing by extending and complementing the existing mechanisms provided by routing protocols. PBR allows you to set the IP Precedence. It also allows you to specify a path for certain traffic, such as priority traffic over a high-cost link.

You can set up PBR as a way to route packets based on configured policies. For example, you can implement routing policies to allow or deny paths based on the identity of a particular end system, an application protocol, or the size of packets.

PBR allows you to perform the following tasks:

- Classify traffic based on extended access list criteria. Access lists are used to establish the match criteria.
- Set IP Precedence bits, giving the network the ability to enable differentiated classes of service.
- Route packets to specific traffic-engineered paths; you might need to route them to allow a specific QoS through the network.

Policies can be based on IP address, port numbers, protocols, or size of packets. For a simple policy, you can use any one of these descriptors; for a complicated policy, you can use all of them.

For example, classification of traffic through PBR allows you to identify traffic for different classes of service at the edge of the network and then implement QoS defined for each CoS in the core of the network using priority queueing (PQ), custom queueing (CQ), or WFQ techniques. This process obviates the need to classify traffic explicitly at each WAN interface in the core-backbone network.

For information on how to configure policy-based routing, see the [“Configuring Policy-Based Routing”](#) chapter in this book.

How It Works

All packets received on an interface with PBR enabled are passed through enhanced packet filters known as route maps. The route maps used by PBR dictate the policy, determining to where the packets are forwarded.

Route maps are composed of statements. The route map statements can be marked as permit or deny, and they are interpreted in the following ways:

- If the packets do not match any route map statements, then all the set clauses are applied.
- If a statement is marked as deny, the packets meeting the match criteria are sent back through the normal forwarding channels and destination-based routing is performed.
- If the statement is marked as permit and the packets do not match any route map statements, the packets are sent back through the normal forwarding channels and destination-based routing is performed.

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

When Should You Use Policy-Based Routing?

You might enable PBR if you want certain packets to be routed some way other than the obvious shortest path. For example, PBR can be used to provide the following functionality:

- equal access
- protocol-sensitive routing
- source-sensitive routing
- routing based on interactive versus batch traffic
- routing based on dedicated links

Some applications or traffic can benefit from QoS-specific routing; for example, you could transfer stock records to a corporate office on a higher-bandwidth, higher-cost link for a short time while sending routine application data such as e-mail over a lower-bandwidth, lower-cost link.

QoS Policy Propagation via Border Gateway Protocol

The Border Gateway Protocol (BGP) is an interdomain routing protocol that exchanges reachability information with other BGP systems. It is defined by RFC 1163.

The Policy Propagation via BGP feature allows you to classify packets based on the following:

- Access lists.
- BGP community lists. A community is a group of destinations that share some common attribute. You use community lists to create groups of communities to use in a match clause of a route map. As with access lists, a series of community lists can be created.
- BGP autonomous system paths. An autonomous system path is a collection of networks under a common administration sharing a common routing strategy. BGP carries the autonomous system path in its routing updates. You can filter routing updates by specifying an access list on both incoming and outbound updates based on the BGP autonomous system path.
- IP Precedence. See the “[About IP Precedence](#)” section earlier in this chapter.
- Source and destination address lookup. You can specify whether the IP Precedence level is obtained from the source (input) address or destination (output) address entry in the route table.

After a packet has been classified using BGP, you can use other QoS features such as CAR and WRED to specify and enforce business policies to fit your business model.

BGP Policy Propagation leverages BGP to distribute QoS policy to remote routers in your network. It allows ingress routers to prioritize incoming traffic.

Restrictions

For the Policy Propagation via BGP feature to work, you must enable BGP and Cisco Express Forwarding (CEF)/distributed CEF (DCEF) on the router.

Subinterfaces on an ATM interface that has the **bgp-policy** command enabled must use CEF mode because DCEF is not supported. (Note that DCEF uses the Versatile Interface Processor (VIP) rather than the Route Switch Processor (RSP) to perform forwarding functions.)

For information on how to configure Policy Propagation via BGP, see the “[Configuring QoS Policy Propagation via Border Gateway Protocol](#)” chapter in this book.

Committed Access Rate

CAR is a multifaceted feature that implements both classification services and policing through rate limiting. This section describes its classification capability. For information on its rate limiting features, see the [“Policing and Shaping Overview”](#) chapter in this book.

You can use the classification services of CAR to set the IP Precedence for packets entering 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 of whether a packet will be 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 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 in this book.

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, marking network traffic is the foundation for enabling many quality of service (QoS) features on your network.

For more information about marking network traffic, see the [“Marking Network Traffic”](#) chapter in this book.

Network-Based Application Recognition

The Network-Based Application Recognition (NBAR) feature adds intelligent network classification to network infrastructures. NBAR is a classification engine that recognizes a wide variety of applications, including web-based and other difficult-to-classify protocols that utilize dynamic TCP/User Datagram Ports (UDP) port assignments. When an application is recognized and classified by NBAR, a network can invoke services for that specific application. NBAR ensures that network bandwidth is used efficiently by working with QoS features to provide the following features:

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

NBAR introduces the following new classification features:

- Classification of applications that dynamically assign TCP/UDP port numbers
- Classification of HTTP traffic by URL, HOST, or Multipurpose Internet Mail Extension (MIME) type
- Classification of Citrix Independent Computing Architecture (ICA) traffic by application name
- Classification of application traffic using subport information

NBAR can also classify static port protocols. Although access control lists (ACLs) can also be used for this purpose, NBAR is easier to configure and can provide classification statistics that are not available when ACLs are used.

NBAR provides a special Protocol Discovery feature that determines which application protocols are traversing a network at any given time. The Protocol Discovery feature captures key statistics associated with each protocol in a network. These statistics can be used to define traffic classes and QoS policies for each traffic class.

NBAR addresses IP QoS classification requirements by classifying application-level protocols so that QoS policies can be applied to the classified traffic. NBAR addresses the ongoing need to extend the classification engine for the many existing and emerging application protocols by providing an extensible Packet Description Language (PDL). NBAR can determine which protocols and applications are currently running on a network so that an appropriate QoS policy can be created based upon the current traffic mix and application requirements.

NBAR can now perform subport classification of HTTP traffic by host name in addition to classification by MIME-type or URL. This ability enables users to classify HTTP traffic by web server names. With URL matching, only the portion of the URL following the host name can be specified for a match. To perform a match on the host name portion of the URL, use the new HOST matching criterion. For example, a HOST match on `http://www.cisco.com/latest/whatsnew.html` will classify all traffic from the web server `www.cisco.com`, whereas a URL match can be performed on the `/latest/whatsnew.html` portion of the URL.

NBAR supports the following RFCs:

- RFC 742, *NAME/FINGER Protocol*
- RFC 759, *Internet Message 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 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 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 2060, *Internet Message Access Protocol - Version 4rev1*
- RFC 2068, *Hypertext Transfer Protocol -- HTTP/1.1*
- RFC 2131, *Dynamic Host Configuration Protocol*
- 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*

NBAR supports the following RFCs:

- RFC 0009, *File Transfer Protocol (FTP)*
- RFC 0013, *Domain Names - Concepts and Facilities*
- RFC 0033, *The TFTP Protocol (Revision 2)*
- RFC 0034, *Routing Information Protocol*
- RFC 0053, *Post Office Protocol - Version 3*
- RFC 0056, *RIP Version 2*

For information on how to configure NBAR, see the “[Configuring Network-Based Application Recognition](#)” chapter in this book.

You must enable CEF before you configure NBAR. For more information on CEF, see the *Cisco IOS Switching Services Configuration Guide*.

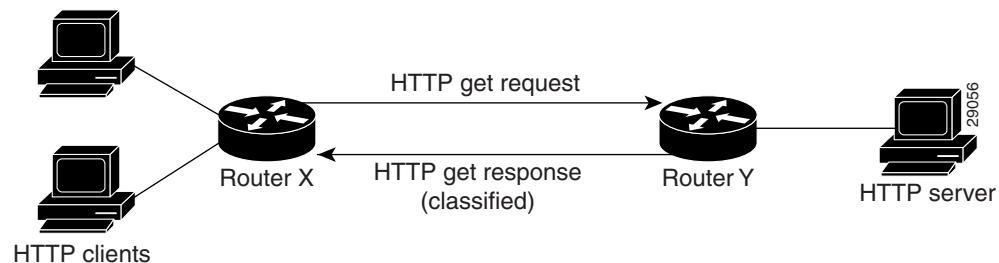
Classification of HTTP by URL, HOST, or MIME

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

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

NBAR recognizes HTTP GET packets containing the URL and classifies all packets that are sent to the source of the HTTP GET request. [Figure 2](#) illustrates a network topology with NBAR in which Router Y is the NBAR-enabled router.

Figure 2 Network Topology with NBAR



When specifying a URL for classification, include only the portion of the URL following the `www.hostname.domain` in the match statement. For example, for the URL `www.cisco.com/latest/whatsnew.html`, include only `/latest/whatsnew.html`.

HOST specification is identical to URL specification. NBAR performs a regular expression match on the HOST field contents inside an HTTP GET 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 Internet Assigned Numbers Authority (IANA)-supported MIME types can be found at the IANA web site.

In MIME type matching, NBAR classifies the packet containing the MIME type and all subsequent packets, which are sent to the source of the HTTP GET request.

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.

Classification of Citrix ICA Traffic by Application Name

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

NBAR statefully tracks Citrix ICA client/server messages and classifies requests for given Citrix application names and traffic. A Citrix application is named when published on a Citrix ICA server. NBAR performs a regular expression match using a user-specified application name string on the contents of the Citrix ICA control packets carrying the published application name. Therefore, users need to specify a regular expression that will result in a match for the published application name if they wish to match a specified application. Refer to the **match protocol citrix** command in the *Cisco IOS Quality of Service Solution Command Reference* for additional information.

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 only be used to classify Citrix applications 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 nonseamless (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.

To turn off session sharing, perform the following steps:

Step 1 At the command prompt of the Citrix server, open the registry editor by entering the **regedit** command.

Step 2 Create the following registry entry (which overrides session sharing):

```
[HKLM] \SYSTEM\CurrentControlSet\Control\Citrix\WFSHELL\TWI
Value name: "SeamlessFlags", type DWORD, possible values :0 or 1
```

Setting this registry value to 1 overrides session sharing. Note that this flag is SERVER GLOBAL.

**Note**

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

Protocol Discovery

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

Packet Description Language Module

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

New PDLMs will be released only by Cisco and can be loaded from Flash memory. Contact your local Cisco representative to request additions or changes to the set of protocols classified by NBAR.

Memory Management

NBAR uses approximately 150 bytes of DRAM for each flow that requires stateful inspection. [Table 4](#) lists the stateful protocols supported by NBAR that require stateful inspection. When NBAR is configured, it allocates 1 MB of DRAM to support up to 5000 concurrent flows. NBAR checks to determine if it needs more memory to handle additional concurrent stateful flows. If such a need is detected, NBAR expands its memory usage in increments of 200 to 400 KB.

Table 4 TCP and UDP Stateful Protocols

Cisco IOS Release ¹	Protocol	Type	Description	Syntax
12.0(5)XE2 12.1(1)E 12.1(5)T	FTP	TCP	File Transfer Protocol	ftp
12.0(5)XE2 12.1(1)E 12.1(5)T	Exchange	TCP	MS-RPC for Exchange	exchange
12.0(5)XE2 12.1(1)E 12.1(5)T (HTTP Host classification is not available on the 12.0 XE train)	HTTP	TCP	HTTP with URL, MIME, or HOST classification	http

Table 4 TCP and UDP Stateful Protocols (continued)

Cisco IOS Release ¹	Protocol	Type	Description	Syntax
12.0(5)XE2 12.1(1)E 12.1(5)T	Netshow	TCP/ UDP	Microsoft Netshow	netshow
12.0(5)XE2 12.1(1)E 12.1(5)T	RealAudio	TCP/ UDP	RealAudio Streaming Protocol	realaudio
12.0(5)XE2 12.1(1)E 12.1(5)T	r-commands	TCP	rsh, rlogin, rexec	rcmd
12.0(5)XE2 12.1(1)E 12.1(5)T	StreamWorks	UDP	Xing Technology Stream Works audio and video	streamwork
12.0(5)XE2 12.1(1)E 12.1(5)T	SQL*NET	TCP/ UDP	SQL*NET for Oracle	sqlnet
12.0(5)XE2 12.1(1)E 12.1(5)T	SunRPC	TCP/ UDP	Sun Remote Procedure Call	sunrpc
12.0(5)XE2 12.1(1)E 12.1(5)T	TFTP	UDP	Trivial File Transfer Protocol	tftp
12.0(5)XE2 12.1(1)E 12.1(5)T	VDOLive	TCP/ UDP	VDOLive Streaming Video	vdolive

1. Indicates the Cisco IOS maintenance release that first supported the protocol.

Supported Protocols

NBAR can classify the following three types of protocols:

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

[Table 5](#) lists all of the non-UDP and non-TCP protocols that NBAR can classify. [Table 6](#) lists the TCP and UDP static port protocols.

Table 5 *Non-UDP and Non-TCP Protocols*

Cisco IOS Release ¹	Protocol	Type	Well-Known Port Number	Description	Syntax
12.0(5)XE2 12.1(1)E 12.1(5)T	EGP	IP	8	Exterior Gateway Protocol	egp
12.0(5)XE2 12.1(1)E 12.1(5)T	GRE	IP	47	Generic Routing Encapsulation	gre
12.0(5)XE2 12.1(1)E 12.1(5)T	ICMP	IP	1	Internet Control Message Protocol	icmp
12.0(5)XE2 12.1(1)E 12.1(5)T	IPINIP	IP	4	IP in IP	ipinip
12.0(5)XE2 12.1(1)E 12.1(5)T	IPSec	IP	50, 51	IP Encapsulating Security Payload/Authentication Header	ipsec
12.0(5)XE2 12.1(1)E 12.1(5)T	EIGRP	IP	88	Enhanced Interior Gateway Routing Protocol	eigrp

1. Indicates the Cisco IOS maintenance release that first supported the protocol.

Table 6 *TCP and UDP Static Port Protocols*

Cisco IOS Release ¹	Protocol	Type	Well-Known Port Number	Description	Syntax
12.0(5)XE2 12.1(1)E 12.1(5)T	BGP	TCP/UDP	179	Border Gateway Protocol	bgp
12.0(5)XE2 12.1(1)E 12.1(5)T	CU-SeeMe	TCP/UDP	7648, 7649	Desktop video conferencing	cuseeme
12.0(5)XE2 12.1(1)E 12.1(5)T	CU-SeeMe	UDP	24032	Desktop video conferencing	cuseeme
12.0(5)XE2 12.1(1)E 12.1(5)T	DHCP/ BOOTP	UDP	67, 68	Dynamic Host Configuration Protocol/ Bootstrap Protocol	dhcp
12.0(5)XE2 12.1(1)E 12.1(5)T	DNS	TCP/UDP	53	Domain Name System	dns
12.0(5)XE2 12.1(1)E 12.1(5)T	Finger	TCP	79	Finger user information protocol	finger

Table 6 TCP and UDP Static Port Protocols (continued)

Cisco IOS Release ¹	Protocol	Type	Well-Known Port Number	Description	Syntax
12.0(5)XE2 12.1(1)E 12.1(5)T	Gopher	TCP/UDP	70	Internet Gopher Protocol	gopher
12.0(5)XE2 12.1(1)E 12.1(5)T	HTTP	TCP	80	Hypertext Transfer Protocol	http
12.0(5)XE2 12.1(1)E 12.1(5)T	HTTPS	TCP	443	Secured HTTP	secure-http
12.0(5)XE2 12.1(1)E 12.1(5)T	IMAP	TCP/UDP	143, 220	Internet Message Access Protocol	imap
12.0(5)XE2 12.1(1)E 12.1(5)T	IRC	TCP/UDP	194	Internet Relay Chat	irc
12.0(5)XE2 12.1(1)E 12.1(5)T	Kerberos	TCP/UDP	88, 749	Kerberos Network Authentication Service	kerberos
12.0(5)XE2 12.1(1)E 12.1(5)T	L2TP	UDP	1701	L2F/L2TP tunnel	l2tp
12.0(5)XE2 12.1(1)E 12.1(5)T	LDAP	TCP/UDP	389	Lightweight Directory Access Protocol	ldap
12.0(5)XE2 12.1(1)E 12.1(5)T	MS-PPTP	TCP	1723	Microsoft Point-to-Point Tunneling Protocol for Virtual Private Networks (VPNs)	pptp
12.0(5)XE2 12.1(1)E 12.1(5)T	MS-SQLServer	TCP	1433	Microsoft SQL Server Desktop Videoconferencing	sqlserver
12.0(5)XE2 12.1(1)E 12.1(5)T	NetBIOS	TCP	137, 139	NetBIOS over IP (MS Windows)	netbios
12.0(5)XE2 12.1(1)E 12.1(5)T	NetBIOS	UDP	137, 138	NetBIOS over IP (MS Windows)	netbios
12.0(5)XE2 12.1(1)E 12.1(5)T	NFS	TCP/UDP	2049	Network File System	nfs
12.0(5)XE2 12.1(1)E 12.1(5)T	NNTP	TCP/UDP	119	Network News Transfer Protocol	nntp

Table 6 TCP and UDP Static Port Protocols (continued)

Cisco IOS Release ¹	Protocol	Type	Well-Known Port Number	Description	Syntax
12.0(5)XE2 12.1(1)E 12.1(5)T	Notes	TCP/UDP	1352	Lotus Notes	notes
12.1(2)E 12.1(5)T	Novadigm	TCP/UDP	3460 to 3465	Novadigm Enterprise Desktop Manager (EDM)	novadigm
12.0(5)XE2 12.1(1)E 12.1(5)T	NTP	TCP/UDP	123	Network Time Protocol	ntp
12.0(5)XE2 12.1(1)E 12.1(5)T	PCAnywhere	TCP	5631, 65301	Symantec PCAnywhere	pcanywhere
12.0(5)XE2 12.1(1)E 12.1(5)T	PCAnywhere	UDP	22, 5632	Symantec PCAnywhere	pcanywhere
12.0(5)XE2 12.1(1)E 12.1(5)T	POP3	TCP/UDP	110	Post Office Protocol	pop3
12.1(2)E 12.1(5)T	Printer	TCP/UDP	515	Printer	printer
12.0(5)XE2 12.1(1)E 12.1(5)T	RIP	UDP	520	Routing Information Protocol	rip
12.0(5)XE2 12.1(1)E 12.1(5)T	RSVP	UDP	1698,1699	Resource Reservation Protocol	rsvp
12.0(5)XE2 12.1(1)E 12.1(5)T	SFTP	TCP	990	Secure FTP	secure-ftp
12.0(5)XE2 12.1(1)E 12.1(5)T	SHTTP	TCP	443	Secure HTTP	secure-http
12.0(5)XE2 12.1(1)E 12.1(5)T	SIMAP	TCP/UDP	585, 993	Secure IMAP	secure-imap
12.0(5)XE2 12.1(1)E 12.1(5)T	SIRC	TCP/UDP	994	Secure IRC	secure-irc
12.0(5)XE2 12.1(1)E 12.1(5)T	SLDAP	TCP/UDP	636	Secure LDAP	secure-ldap

Table 6 TCP and UDP Static Port Protocols (continued)

Cisco IOS Release ¹	Protocol	Type	Well-Known Port Number	Description	Syntax
12.0(5)XE2 12.1(1)E 12.1(5)T	SNMTP	TCP/UDP	563	Secure NNTP	secure-nntp
12.0(5)XE2 12.1(1)E 12.1(5)T	SMTP	TCP	25	Simple Mail Transfer Protocol	smtp
12.0(5)XE2 12.1(1)E 12.1(5)T	SNMP	TCP/UDP	161, 162	Simple Network Management Protocol	snmp
12.0(5)XE2 12.1(1)E 12.1(5)T	SOCKS	TCP	1080	Firewall security protocol	socks
12.0(5)XE2 12.1(1)E 12.1(5)T	SPOP3	TCP/UDP	995	Secure POP3	secure-pop3
12.0(5)XE2 12.1(1)E 12.1(5)T	SSH	TCP	22	Secured Shell	ssh
12.0(5)XE2 12.1(1)E 12.1(5)T	STELNET	TCP	992	Secure Telnet	secure-telnet
12.0(5)XE2 12.1(1)E 12.1(5)T	Syslog	UDP	514	System Logging Utility	syslog
12.0(5)XE2 12.1(1)E 12.1(5)T	Telnet	TCP	23	Telnet Protocol	telnet
12.0(5)XE2 12.1(1)E 12.1(5)T	X Window System	TCP	6000-6003	X11, X Window System	xwindows

1. Indicates the Cisco IOS maintenance release that first supported the protocol.

Restrictions

The NBAR feature does not support the following:

- More than 24 concurrent URLs, HOSTs, or MIME type matches
- Matching beyond the first 400 bytes in a URL
- Non-IP traffic
- 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 originating from or destined to the router running NBAR

NBAR is not configurable on the following logical interfaces:

- Fast EtherChannel
- Interfaces where tunneling or encryption is used
- VLANs



Note NBAR is configurable on VLANs as of Cisco IOS Release 12.1(13)E, but supported in the software switching path only.

- Dialer interfaces
- Multilink PPP



Note

NBAR cannot be used to classify output traffic on a WAN link where tunneling or encryption is used. Therefore, NBAR should be configured on other interfaces on the router (such as a LAN link) to perform input classification before the traffic is switched to the WAN link for output.



Configuring Policy-Based Routing

This chapter describes the tasks for configuring policy-based routing (PBR) on a router.

For complete conceptual information about this feature, see the section [“Policy-Based Routing”](#) in the chapter [“Classification Overview”](#) in this book.

For a complete description of the PBR commands in this chapter, refer to 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.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the [“Finding Additional Feature Support Information”](#) section on page lxix in the [“Using Cisco IOS Software for Release 12.4”](#) chapter in this book.

Policy-Based Routing Configuration Task List

To configure PBR, 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 PBR](#) (Required)
- [Enabling Fast-Switched PBR](#) (Optional)
- [Enabling Local PBR](#) (Optional)
- [Enabling CEF-Switched PBR](#) (Optional)

See the end of this chapter for the section [“Policy-Based Routing Configuration Examples.”](#)

Enabling PBR

To enable PBR, you must create a route map that specifies the match criteria and the resulting action if all of the match clauses are met. Then, you must enable PBR for that route map on a particular interface. All packets arriving on the specified interface matching the match clauses will be subject to PBR.

To enable PBR on an interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# route-map <i>map-tag</i> [permit deny] [<i>sequence-number</i>]	Defines a route map to control where packets are output. This command puts the router into route-map configuration mode.
Step 2	Router(config-route-map)# match length <i>min max</i> Router(config-route-map)# match ip address { <i>access-list-number</i> <i>name</i> } [... <i>access-list-number</i> <i>name</i>]	Specifies the match criteria. Although there are many route-map matching options, here you can specify only length and/or ip address. <ul style="list-style-type: none"> • length matches the Level 3 length of the packet. • ip address matches the source or destination IP address that is permitted by one or more standard or extended access lists. If you do not specify a match command, the route map applies to <i>all</i> packets.
Step 3	Router(config-route-map)# set ip precedence [<i>number</i> <i>name</i>] Router(config-route-map)# set ip df Router(config-route-map)# set ip vrf <i>vrf_name</i> Router(config-route-map)# set ip next-hop <i>ip-address</i> [... <i>ip-address</i>] Router(config-route-map)# set ip next-hop recursive <i>ip-address</i> [... <i>ip-address</i>] Router(config-route-map)# set interface <i>interface-type interface-number</i> [... <i>type number</i>] Router(config-route-map)# set ip default next-hop <i>ip-address</i> [... <i>ip-address</i>] Router(config-route-map)# set default interface <i>interface-type interface-number</i> [... <i>type ...number</i>]	Specifies the action(s) to take on the packets that match the criteria. You can specify any or all of the following: <ul style="list-style-type: none"> • precedence: Sets precedence value in the IP header. You can specify either the precedence number or name. • df: Sets the ‘Don’t Fragment’ (DF) bit in the ip header. • vrf: Sets the VPN Routing and Forwarding (VRF) instance. • next-hop: Sets next hop to which to route the packet. • next-hop recursive: Sets next hop to which to route the packet if the hop is to a router which is not adjacent. • interface: Sets output interface for the packet. • default next-hop: Sets next hop to which to route the packet if there is no explicit route for this destination. • default interface: Sets output interface for the packet if there is no explicit route for this destination.

	Command	Purpose
Step 4	Router(config-route-map)# interface <i>interface-type</i> <i>interface-number</i>	Specifies the interface, and puts the router into interface configuration mode.
Step 5	Router(config-if)# ip policy route-map <i>map-tag</i>	Identifies the route map to use for PBR. One interface can have only one route map tag; but you can have several route map entries, each with its own sequence number. Entries are evaluated in order of their sequence numbers until the first match occurs. If no match occurs, packets are routed as usual.

The **set** commands can be used in conjunction with each other. They are evaluated in the order shown in Step 3 in the previous task table. A usable next hop implies an interface. Once the local router finds a next hop and a usable interface, it routes the packet.

**Note**

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

If you want PBR to be fast-switched, see the section “[Enabling Fast-Switched PBR](#),” which follows.

Enabling Fast-Switched PBR

IP PBR can now be fast-switched. Prior to Cisco IOS Release 12.0, PBR could only be process-switched, which meant that on most platforms the switching rate was approximately 1000 to 10,000 packets per second. This speed was not fast enough for many applications. Users that need PBR to occur at faster speeds can now implement PBR without slowing down the router.

Fast-switched PBR supports all of the **match** commands and most of the **set** commands, with the following restrictions:

- The **set ip default next-hop** and **set default interface** commands are not supported.
- The **set interface** command is supported only over point-to-point links, unless a route cache entry exists using the same interface specified in the **set interface** command in the route map. Also, at the process level, the routing table is consulted to determine if the interface is on a reasonable path to the destination. During fast switching, the software does not make this check. Instead, if the packet matches, the software blindly forwards the packet to the specified interface.

PBR must be configured before you configure fast-switched PBR. Fast switching of PBR is disabled by default. To enable fast-switched PBR, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# ip route-cache policy	Enables fast switching of PBR.

To display the cache entries in the policy route cache, use the **show ip cache policy** command. To display which route map is associated with which interface, use the **show ip policy** command.

Enabling Local PBR

Packets that are generated by the router are not normally policy-routed. To enable local PBR for such packets, indicate which route map the router should use by using the following command in global configuration mode:

Command	Purpose
Router(config)# ip local policy route-map <i>map-tag</i>	Identifies the route map to use for local PBR.

All packets originating on the router will then be subject to local PBR.

Use the **show ip local policy** command to display the route map used for local PBR, if one exists.

Enabling CEF-Switched PBR

Beginning in Cisco IOS Release 12.0, PBR is supported in the Cisco Express Forwarding (CEF) switching path. CEF-switched PBR has better performance than fast-switched PBR and, therefore, is the optimal way to perform PBR on a router.

No special configuration is required to enable CEF-switched PBR. It is on by default as soon as you enable CEF and PBR on the router.



Note

The **ip route-cache policy** command is strictly for fast-switched PBR and, therefore, not required for CEF-switched PBR.

Policy-Based Routing Configuration Examples

The following sections provide PBR configuration examples:

- [Equal Access Example](#)
- [Differing Next Hops Example](#)

For information on how to configure policy-based routing, see the section [“Policy-Based Routing Configuration Task List”](#) in this chapter.

Equal Access Example

The following example provides two sources with equal access to two different service providers. Packets arriving on asynchronous interface 1 from the source 1.1.1.1 are sent to the router at 6.6.6.6 if the router has no explicit route for the destination of the packet. Packets arriving from the source 2.2.2.2 are sent to the router at 7.7.7.7 if the router has no explicit route for the destination of the packet. All other packets for which the router has no explicit route to the destination are discarded.

```
access-list 1 permit ip 1.1.1.1
access-list 2 permit ip 2.2.2.2
!
interface async 1
 ip policy route-map equal-access
!
```

```
route-map equal-access permit 10
 match ip address 1
  set ip default next-hop 6.6.6.6
route-map equal-access permit 20
 match ip address 2
  set ip default next-hop 7.7.7.7
route-map equal-access permit 30
 set default interface null0
```

Differing Next Hops Example

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

```
access-list 1 permit ip 1.1.1.1
access-list 2 permit ip 2.2.2.2
!
interface ethernet 1
 ip policy route-map Texas
!
route-map Texas permit 10
 match ip address 1
  set ip precedence priority
  set ip next-hop 3.3.3.3
!
route-map Texas permit 20
 match ip address 2
  set ip precedence critical
  set ip next-hop 3.3.3.5
```




Configuring QoS Policy Propagation via Border Gateway Protocol

This chapter describes the tasks for configuring Policy Propagation via Border Gateway Protocol (BGP) on a router.

For complete conceptual information about this feature, see the section “[QoS Policy Propagation via Border Gateway Protocol](#)” in the chapter “[Classification Overview](#)” in this book.

For a complete description of the Policy Propagation via BGP commands in this chapter, refer to 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.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the “[Finding Additional Feature Support Information](#)” section on page [lxix](#) in the “[Using Cisco IOS Software for Release 12.4](#)” chapter of this book.

Policy Propagation via BGP Configuration Task Overview

The Policy Propagation via BGP feature allows you to classify packets by IP precedence based on BGP community lists, BGP autonomous system paths, and access lists. After a packet has been classified, you can use other QoS features such as committed access rate (CAR) and Weighted Random Early Detection (WRED) to specify and enforce policies to fit your business model.

To configure Policy Propagation via BGP, perform the following basic tasks:

- Configure BGP and Cisco Express Forwarding (CEF) or distributed CEF (dCEF). To configure BGP, refer to the *Cisco IOS IP Configuration Guide*. To configure CEF and dCEF, refer to the *Cisco IOS Switching Services Configuration Guide*.
- Define the policy.
- Apply the policy through BGP.
- Configure the BGP community list, BGP autonomous system path, or access list and enable the policy on an interface. For information about this task, see the next section in this chapter.
- Enable CAR or WRED to use the policy. To enable CAR, see the chapter “[Configuring Committed Access Rate](#)” in this book. To configure WRED, see the chapter “[Configuring Weighted Random Early Detection](#)” in this book.

This chapter describes how to configure Policy Propagation based on BGP community list, BGP autonomous system path, or access list. It assumes you have already configured BGP and CEF or dCEF.

Policy Propagation via BGP Configuration Task List

To configure Policy Propagation via BGP, perform the tasks described in the following sections. The tasks in the first three sections are required; the task in the remaining section is optional.

- [Configuring Policy Propagation Based on Community Lists](#) (Required)
- [Configuring Policy Propagation Based on the Autonomous System Path Attribute](#) (Required)
- [Configuring Policy Propagation Based on an Access List](#) (Required)
- [Monitoring Policy Propagation via BGP](#) (Optional)



Note

For the Policy Propagation via BGP feature to work, you must enable BGP and CEF/dCEF on the router. Subinterfaces on an ATM interface that have the **bgp-policy** command enabled must use CEF mode because dCEF is not supported. dCEF uses the Versatile Interface Processor (VIP) rather than the Route Switch Processor (RSP) to perform forwarding functions.

See the end of this chapter for the section “[Policy Propagation via BGP Configuration Examples](#).”

Configuring Policy Propagation Based on Community Lists

This section describes how to configure Policy Propagation via BGP using community lists. The tasks listed in this section are required unless noted as optional. This section assumes you have already configured CEF/dCEF and BGP on your router.

To configure the router to propagate the IP precedence based on the community lists, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# route-map <i>route-map-name</i> [permit deny [<i>sequence-number</i>]]	Defines a route map to control redistribution and enters route-map configuration mode.
Step 2	Router(config-route-map)# match community-list <i>community-list-number</i> [exact]	Matches a BGP community list.
Step 3	Router(config-route-map)# set ip precedence [<i>number</i> <i>name</i>]	Sets the IP Precedence field when the community list matches. You can specify either a precedence number or name.
Step 4	Router(config-router)# router bgp <i>autonomous-system</i>	Enters router configuration mode.
Step 5	Router(config-router)# table-map <i>route-map-name</i>	Modifies the metric and tag values when the IP routing table is updated with BGP learned routes.
Step 6	Router(config-router)# ip community-list <i>community-list-number</i> { permit deny } <i>community-number</i>	Creates a community list for BGP and controls access to it.
Step 7	Router(config-router)# interface <i>interface-type</i> <i>interface-number</i>	Specifies the interfaces (or subinterface) and enters interface configuration mode.

	Command	Purpose
Step 8	Router(config-if)# bgp-policy ip-prec-map	Classifies packets using IP Precedence.
Step 9	Router(config-if)# ip bgp-community new-format	(Optional) Configures a new community format so that the community number is displayed in the short form.

Configuring Policy Propagation Based on the Autonomous System Path Attribute

This section describes how to configure Policy Propagation via BGP based on the autonomous system path. This section assumes you have already configured CEF/dCEF and BGP on your router.

To configure the router to propagate the IP precedence based on the autonomous system path attribute, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# route-map <i>route-map-name</i> [permit deny [<i>sequence-number</i>]]	Defines a route map to control redistribution and enters route-map configuration mode.
Step 2	Router(config-route-map)# match as-path <i>path-list-number</i>	Matches a BGP autonomous system path access list.
Step 3	Router(config-route-map)# set ip precedence [<i>number</i> <i>name</i>]	Sets the IP Precedence field when the autonomous system path matches. Specifies either a precedence number or name.
Step 4	Router(config-route-map)# router bgp <i>autonomous-system</i>	Enters router configuration mode.
Step 5	Router(config-router)# table-map <i>route-map-name</i>	Modifies the metric and tag values when the IP routing table is updated with BGP learned routes.
Step 6	Router(config-router)# ip as-path access-list <i>access-list-number</i> { permit deny } <i>as-regular-expression</i>	Defines an autonomous system path access list.
Step 7	Router(config-router)# interface <i>interface-type</i> <i>interface-number</i>	Specifies the interfaces (or subinterface) and enters interface configuration mode.
Step 8	Router(config-if)# bgp-policy ip-prec-map	Classifies packets using IP Precedence.

Configuring Policy Propagation Based on an Access List

This section describes how to configure Policy Propagation via BGP based on an access list. This section assumes you have already configured CEF/dCEF and BGP on your router.

To configure the router to propagate the IP Precedence based on an access list, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# route-map <i>route-map-name</i> [permit deny [<i>sequence-number</i>]]	Defines a route map to control redistribution and enters route-map configuration mode.
Step 2	Router(config-route-map)# match ip address <i>access-list-number</i>	Matches an access list.
Step 3	Router(config-route-map)# set ip precedence [<i>number</i> <i>name</i>]	Sets the IP Precedence field when the autonomous system path matches.
Step 4	Router(config-route-map)# router bgp <i>autonomous-system</i>	Enters router configuration mode.
Step 5	Router(config-router)# table-map <i>route-map-name</i>	Modifies the metric and tag values when the IP routing table is updated with BGP learned routes.
Step 6	Router(config-router)# access-list <i>access-list-number</i> { permit deny } <i>source</i>	Defines an access list.
Step 7	Router(config-router)# interface <i>interface-type</i> <i>interface-number</i>	Specifies the interfaces (or subinterface) and enters interface configuration mode.
Step 8	Router(config-if)# bgp-policy ip-prec-map	Classifies packets using IP Precedence.

Monitoring Policy Propagation via BGP

To monitor the Policy Propagation via BGP configuration, use the following commands in EXEC mode, as needed. The commands listed in this section are optional.

Command	Purpose
Router# show ip bgp	Displays entries in the BGP routing table, to verify that the correct community is set on the prefixes.
Router# show ip bgp community-list <i>community-list-number</i>	Displays routes permitted by the BGP community list, to verify that the correct prefixes are selected.
Router# show ip cef <i>network</i>	Displays entries in the Forwarding Information Base (FIB) table based on the IP address, to verify that CEF has the correct precedence value for the prefix.
Router# show ip interface	Displays information about the interface.
Router# show ip route <i>prefix</i>	Displays the current status of the routing table, to verify that the correct precedence values are set on the prefixes.

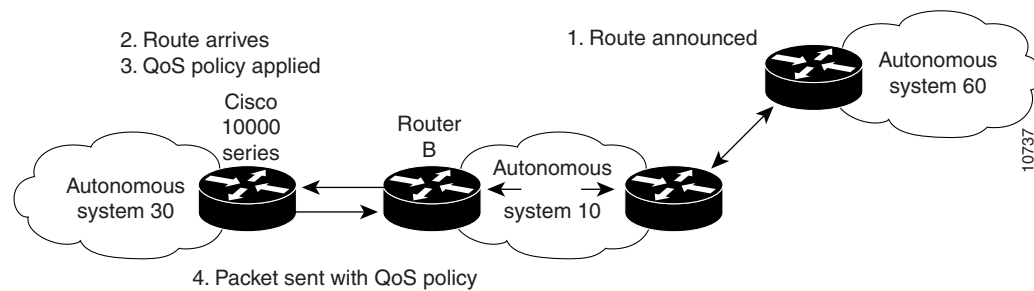
Policy Propagation via BGP Configuration Examples

The following example shows how to create route maps to match access lists, BGP community lists, and BGP autonomous system paths, and apply IP precedence to routes learned from neighbors.

For information on how to configure Policy Propagation via BGP, see the section “[Policy Propagation via BGP Configuration Task Overview](#)” in this chapter.

In [Figure 3](#), Router A learns routes from autonomous system 10 and autonomous system 60. QoS policy is applied to all packets that match the defined route maps. Any packets from Router A to autonomous system 10 or autonomous system 60 are sent the appropriate QoS policy, as the numbered steps indicate.

Figure 3 Router Learning Routes and Applying QoS Policy



Router A Configuration

```
router bgp 30
  table-map precedence-map
  neighbor 20.20.20.1 remote-as 10
  neighbor 20.20.20.1 send-community
  !
  ip bgp-community new-format
  !
  ! Match community 1 and set the IP Precedence to priority
  route-map precedence-map permit 10
  match community 1
  set ip precedence priority
  !
  ! Match community 2 and set the IP Precedence to immediate
  route-map precedence-map permit 20
  match community 2
  set ip precedence immediate
  !
  ! Match community 3 and set the IP Precedence to flash
  route-map precedence-map permit 30
  match community 3
  set ip precedence flash
  !
  ! Match community 4 and set the IP Precedence to flash-override
  route-map precedence-map permit 40
  match community 4
  set ip precedence flash-override
  !
  ! Match community 5 and set the IP Precedence to critical
  route-map precedence-map permit 50
  match community 5
  set ip precedence critical
  !
  ! Match community 6 and set the IP Precedence to internet
```

```

route-map precedence-map permit 60
  match community 6
  set ip precedence internet
!
! Match community 7 and set the IP Precedence to network
route-map precedence-map permit 70
  match community 7
  set ip precedence network
!
! Match ip address access list 69 or match AS path 1
! and set the IP Precedence to critical
route-map precedence-map permit 75
  match ip address 69
  match as-path 1
  set ip precedence critical
!
! For everything else, set the IP Precedence to routine
route-map precedence-map permit 80
  set ip precedence routine
!
! Define the community lists
ip community-list 1 permit 60:1
ip community-list 2 permit 60:2
ip community-list 3 permit 60:3
ip community-list 4 permit 60:4
ip community-list 5 permit 60:5
ip community-list 6 permit 60:6
ip community-list 7 permit 60:7
!
! Define the AS path
ip as-path access-list 1 permit ^10_60
!
! Define the access list
access-list 69 permit 69.0.0.0

```

Router B Configuration

```

router bgp 10
  neighbor 30.30.30.1 remote-as 30
  neighbor 30.30.30.1 send-community
  neighbor 30.30.30.1 route-map send_community out
!
ip bgp-community new-format
!
! Match prefix 10 and set community to 60:1
route-map send_community permit 10
  match ip address 10
  set community 60:1
!
! Match prefix 20 and set community to 60:2
route-map send_community permit 20
  match ip address 20
  set community 60:2
!
! Match prefix 30 and set community to 60:3
route-map send_community permit 30
  match ip address 30
  set community 60:3
!
! Match prefix 40 and set community to 60:4
route-map send_community permit 40
  match ip address 40
  set community 60:4
!

```

```
! Match prefix 50 and set community to 60:5
route-map send_community permit 50
  match ip address 50
  set community 60:5
!
! Match prefix 60 and set community to 60:6
route-map send_community permit 60
  match ip address 60
  set community 60:6
!
! Match prefix 70 and set community to 60:7
route-map send_community permit 70
  match ip address 70
  set community 60:7
!
! For all others, set community to 60:8
route-map send_community permit 80
  set community 60:8
!
! Define the access lists
access-list 10 permit 61.0.0.0
access-list 20 permit 62.0.0.0
access-list 30 permit 63.0.0.0
access-list 40 permit 64.0.0.0
access-list 50 permit 65.0.0.0
access-list 60 permit 66.0.0.0
access-list 70 permit 67.0.0.0
```




Configuring Committed Access Rate

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

For complete conceptual information about these features, see the section [“Committed Access Rate”](#) in the [“Classification Overview”](#) chapter and the section [“Policing with CAR”](#) in the [“Policing and Shaping Overview”](#) chapter in this book.

For a complete description of the CAR commands in this chapter, refer to 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.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the [“Finding Additional Feature Support Information”](#) section on page lxix in the [“Using Cisco IOS Software for Release 12.4”](#) chapter in this book.



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.

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.

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 <i>interface-type</i> <i>interface-number</i>	Specifies the interface or subinterface. This command puts the router in interface configuration mode.
Step 2	Router(config-if)# rate-limit { input output } <i>bps</i> <i>burst-normal burst-max conform-action action</i> exceed-action <i>action</i>	Specifies a basic CAR policy for all IP traffic. See Table 7 for a description of conform and exceed <i>action</i> 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 7](#).

Table 7 *Rate-Limit Command Action Keywords*

Keyword	Description
continue	Evaluates the next rate-limit command.
drop	Drops the packet.
set-prec-continue <i>new-prec</i>	Sets the IP Precedence and evaluates the next rate-limit command.
set-prec-transmit <i>new-prec</i>	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 (or DCAR 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 <i>interface-type</i> <i>interface-number</i>	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] <i>acl-index</i>] <i>bps</i> <i>burst-normal burst-max conform-action action</i> exceed-action <i>action</i>	Specifies the rate policy for each particular class of traffic. See Table 7 for a description of the rate-limit command action keywords. Repeat this command for each different class of traffic.
Step 3	Router(config-if)# access-list rate-limit <i>acl-index</i> { <i>precedence</i> <i>mac-address</i> mask <i>prec-mask</i> }	(Optional) Specifies a rate-limited access list. Repeat this command if you wish to specify a new access list.
Step 4	Router(config-if)# access-list <i>acl-index</i> { deny permit } <i>source</i> [<i>source-wildcard</i>] or Router(config-if)# access-list <i>acl-index</i> { deny permit } <i>protocol source source-wildcard destination</i> <i>destination-wildcard</i> [precedence <i>precedence</i>] [tos <i>tos</i>] [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 <i>interface-type</i> <i>interface-number</i>	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] <i>acl-index</i>] <i>bps burst-normal</i> <i>burst-max conform-action action exceed-action action</i>	Specifies the rate policy for each particular class of traffic. See Table 7 for a description of the rate-limit command action keywords. Repeat this command for each different class of traffic.

	Command	Purpose
Step 3	Router(config-if)# random-detect precedence <i>precedence min-threshold max-threshold mark-prob-denominator</i>	Configures WRED and specifies parameters for packets with specific IP Precedence.
Step 4	Router(config-if)# access-list <i>acl-index</i> {deny permit} <i>source</i> [<i>source-wildcard</i>] or Router(config-if)# access-list <i>acl-index</i> {deny permit} <i>protocol source source-wildcard destination destination-wildcard</i> [precedence <i>precedence</i>] [tos <i>tos</i>] [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 [<i>access-list-number</i>]	Displays information about rate-limited access lists.
Router# show interfaces [<i>interface-type interface-number</i>] 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

```

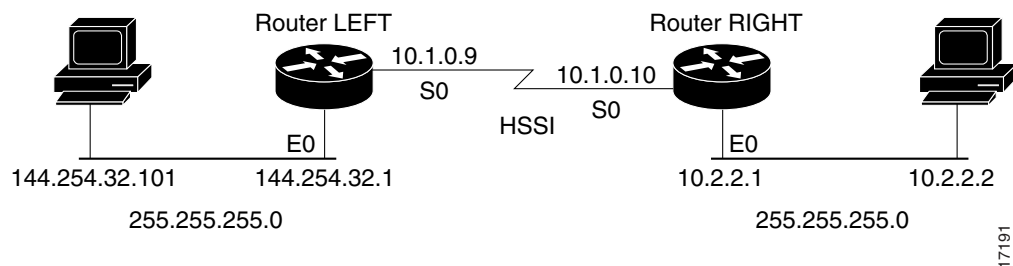
Rate Limiting by Access List Example

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 16,000 bytes and an Excess Burst size of 24,000 bytes. Traffic that conforms is sent with an IP precedence of 5. Traffic that does not conform is dropped.

Figure 4 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 4 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

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.

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 features. To find information about feature support and configuration, use the [“Feature Information for Marking Network Traffic”](#) section on page 79.

Contents

- [Prerequisites for Marking Network Traffic, page 57](#)
- [Restrictions for Marking Network Traffic, page 58](#)
- [Information About Marking Network Traffic, page 58](#)
- [How to Mark Network Traffic, page 65](#)
- [Configuration Examples for Marking Network Traffic, page 74](#)
- [Additional References, page 77](#)
- [Glossary, page 79](#)
- [Feature Information for Marking Network Traffic, page 79](#)

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

Information About Marking Network Traffic

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

- [Purpose of Marking Network Traffic, page 58](#)
- [Benefits of Marking Network Traffic, page 59](#)
- [Two Methods for Marking Traffic Attributes, page 59](#)
- [MQC and Network Traffic Marking, page 64](#)
- [Traffic Classification Compared with Traffic Marking, page 64](#)

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-class value
- DSCP value in the type of service (ToS) byte
- Discard eligible (DE) bit setting in the address field of a Frame Relay frame
- ToS bits in the header of an IP packet
- Multiprotocol Label Switching (MPLS) experimental (EXP) field on all imposed label entries
- MPLS EXP field value in the topmost label on either an input or an output interface
- Precedence value in the packet header
- QoS group identifier (ID)

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 with a **set** command.
With this method you configure individual **set** commands for the traffic attribute 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 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 8](#) lists the available **set** commands and the corresponding attribute. [Table 8](#) also includes the network layer and the network protocol typically associated with the traffic attribute.

Table 8 *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. See the command documentation for the Cisco IOS release you are using for more information.

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 10](#).

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

```
enable
configure terminal
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 67.

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 70.

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 9](#) lists the traffic attributes for which a to-from relationship can be established using the table map.

Table 9 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 66](#).

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 10](#).

Table 10 Commands Used in Policy Maps to Map Attributes

Command Used in Policy Maps	Maps These Attributes
set cos dscp table <i>table-map-name</i>	CoS to DSCP
set cos precedence table <i>table-map-name</i>	CoS to Precedence

Table 10 **Commands Used in Policy Maps to Map Attributes (continued)**

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

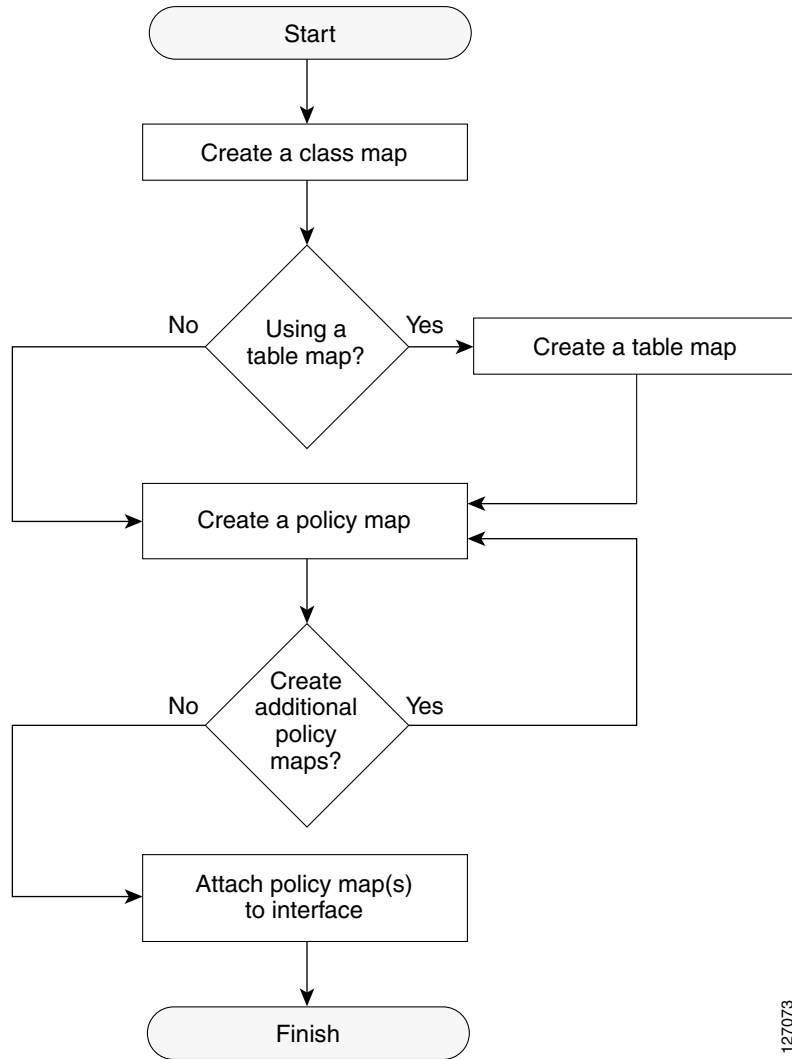
For information on configuring a policy map to use a table map, see the [“Creating a Policy Map for Applying a QoS Feature to Network Traffic”](#) section on page 67.

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 70.

Traffic Marking Procedure Flowchart

[Figure 5](#) illustrates the order of the procedures for configuring traffic marking.

Figure 5 Traffic Marking Procedure Flowchart



127073

For more information on class maps and policy maps, see the [“MQC and Network Traffic Marking” section on page 64](#).

For more information on table maps, see the [“Creating a Table Map for Marking Network Traffic” section on page 66](#).

For more information on completing the processes shown in this flow chart, see the [“How to Mark Network Traffic” section on page 65](#).

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 “Modular Quality of Service Command-Line Interface” section of the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.3.

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 11 compares the features of traffic classification and traffic marking.

Table 11 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 <i>table-map-name</i> 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 65](#) (required)
- [Creating a Table Map for Marking Network Traffic, page 66](#) (optional)
- [Creating a Policy Map for Applying a QoS Feature to Network Traffic, page 67](#) (required)
- [Attaching the Policy Map to an Interface, page 70](#) (required)
- [Configuring QoS When Using IPSec VPNs, page 72](#) (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.

SUMMARY STEPS

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



Note 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 you are using for a complete list of **match** commands.

5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	class-map <i>class-map-name</i> [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. <ul style="list-style-type: none"> • Enter the class map name.

	Command or Action	Purpose
Step 4	match fr-dlci <i>dlci-number</i> 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 Relay DLCI 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 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 67.

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. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	table-map <i>table-map-name</i> map from <i>from-value</i> to <i>to-value</i> [default <i>default-action-or-value</i>] 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. <ul style="list-style-type: none"> 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 <i>default-action-or-value</i> argument sets the default value (or action) to be used if a value is not explicitly designated.
Step 4	exit Example: Router(config-tablemap)# exit	(Optional) Exits tablemap configuration 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.

SUMMARY STEPS

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



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 8 on page 60](#).

or

set cos dscp table *table-map-name*



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 10 on page 61](#).

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. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	policy-map <i>policy-map-name</i> Example: Router(config)# policy-map policy1	Specifies the name of the policy map created earlier and enters policy-map configuration mode. <ul style="list-style-type: none"> • Enter the policy map name.

	Command or Action	Purpose
Step 4	<p>class {<i>class-name</i> class-default}</p> <p>Example: Router(config-pmap)# class class1</p>	<p>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.</p> <ul style="list-style-type: none"> Enter the name of the class or enter the class-default keyword.
Step 5	<p>set cos <i>cos-value</i></p> <p>or</p> <p>set cos dscp table <i>table-map-name</i></p> <p>Example: Router(config-pmap-c)# set cos 2</p> <p>or</p> <p>Example: Router(config-pmap-c)# set cos dscp table table-map1</p>	<p>(Optional) Sets the CoS value in the type of service (ToS) byte.</p> <p>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 8 on page 60.</p> <p>or</p> <p>(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.</p> <p>Note The set cos dscp table <i>table-map-name</i> command is an example of one of the commands that can be used. For a list of other commands, see Table 10 on page 61.</p>
Step 6	<p>end</p> <p>Example: Router(config-pmap-c)# end</p>	<p>Returns to privileged EXEC mode.</p>

	Command or Action	Purpose
Step 7	<pre>show policy-map</pre> <p>or</p> <pre>show policy-map policy-map class class-name</pre> <p>Example: Router# show policy-map</p> <p>or</p> <p>Example: Router# show policy-map policy1 class class7</p>	<p>(Optional) Displays all configured policy maps.</p> <p>or</p> <p>(Optional) Displays the configuration for the specified class of the specified policy map.</p> <ul style="list-style-type: none"> Enter the policy map name and the class name.
Step 8	<pre>exit</pre> <p>Example: Router# exit</p>	<p>(Optional) Exits privileged EXEC mode.</p>

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 67. Then attach the policy maps to the appropriate interface, following the instructions in the [“Attaching the Policy Map to an Interface”](#) section on page 70.

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

- enable
- configure terminal
- interface *type number* [name-tag]
- pvc [*name*] *vpilvci* [ilmi | qsaal | smds | l2transport]
- exit
- service-policy {input | output} *policy-map-name*
- end

8. `show policy-map interface interface-name`
9. `exit`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p><code>enable</code></p> <p>Example: Router> enable</p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	<p><code>configure terminal</code></p> <p>Example: Router# configure terminal</p>	<p>Enters global configuration mode.</p>
Step 3	<p><code>interface type number [name-tag]</code></p> <p>Example: Router(config)# interface serial4/0</p>	<p>Configures an interface type and enters interface configuration mode.</p> <ul style="list-style-type: none"> Enter the interface type and number.
Step 4	<p><code>pvc [name] vpi/vci [ilmi qsaal smds l2transport]</code></p> <p>Example: Router(config-if)# pvc cisco 0/16</p>	<p>(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.</p> <ul style="list-style-type: none"> Enter the PVC name, the ATM network virtual path identifier, and the network virtual channel identifier. <p>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.</p>
Step 5	<p><code>exit</code></p> <p>Example: Router(config-atm-vc)# exit</p>	<p>(Optional) Returns to interface configuration mode.</p> <p>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.</p>
Step 6	<p><code>service-policy {input output} policy-map-name</code></p> <p>Example: Router(config-if)# service-policy input policy1</p>	<p>Attaches a policy map to an input or output interface.</p> <ul style="list-style-type: none"> Enter the policy map name. <p>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.</p>

	Command or Action	Purpose
Step 7	<code>end</code> Example: Router(config-if)# end	Returns to privileged EXEC mode.
Step 8	<code>show policy-map interface interface-name</code> Example: Router# show policy-map interface serial4/0	(Optional) Displays the traffic 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. <ul style="list-style-type: none"> • Enter the interface name.
Step 9	<code>exit</code> 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 [Cisco IOS Security Configuration Guide](#), Release 12.3.

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

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. `exit`

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	crypto map <i>map-name seq-num</i> Example: Router(config)# crypto map mymap 10	Enters crypto map configuration mode and creates or modifies a crypto map entry. <ul style="list-style-type: none">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 <i>type number [name-tag]</i> Example: Router(config)# interface serial4/0	Configures an interface type and enters interface configuration mode. <ul style="list-style-type: none">Enter the interface type and number.
Step 6	qos pre-classify Example: Router(config-if)# qos pre-classify	Enables QoS classification prior to IPSec encryption.
Step 7	exit Example: Router(config-if)# exit	(Optional) Exits interface configuration mode.

Configuration Examples for Marking Network Traffic

This section contains the following examples:

- [Creating a Class Map for Marking Network Traffic: Example, page 74](#)
- [Creating a Table Map for Marking Network Traffic: Example, page 74](#)
- [Creating a Policy Map for Applying a QoS Feature to Network Traffic: Examples, page 74](#)
- [Attaching the Policy Map to an Interface: Example, page 77](#)
- [Configuring QoS When Using IPSec VPNs: Example, page 77](#)

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)# exit
```

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
Router# show policy-map class1
Router# exit
```


Policy Map Configured to Use a Table Map

A policy map called `policy2` 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)# exit
```



Note

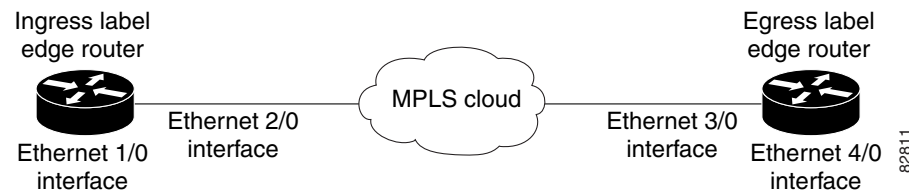
As an alternative to configuring the `set cos dscp table table-map1` command shown in the example, you could configure the command without specifying the `table` keyword and the applicable `table-map-name` argument (that is, you could configure the `set cos dscp` command). When the command is configured 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 MLSP EXP Values

This section contains an example of a policy map configured to map MPLS experimental (EXP) values. [Figure 6](#) illustrates the network topology for this configuration example.

Figure 6 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)# exit
```

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)# exit
```

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)# exit
```

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 77), 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
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 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)# exit
```

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 , Release 12.3 T
MQC	“Modular Quality of Service Command-Line Interface” section of the Cisco IOS Quality of Service Solutions Configuration Guide , Release 12.3
CEF	Cisco IOS Switching Services Configuration Guide , Release 12.3
Classifying network traffic	“Classifying Network Traffic” module
IPSec and VPNs	Cisco IOS Security Configuration Guide , Release 12.3
Committed Access Rate (CAR)	“Classification” section of the Cisco IOS Quality of Service Solutions Configuration Guide , Release 12.3

Additional References

Related Topic	Document Title
Policy-based routing	<i>Cisco IOS Quality of Service Solutions Configuration Guide</i> , Release 12.3
QoS policy propagation via Border Gateway Protocol (BGP)	<i>Cisco IOS Quality of Service Solutions Configuration Guide</i> , Release 12.3

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 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

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.

**Note**

See *[Internetworking Terms and Acronyms](#)* for terms not included in this glossary.

Feature Information for Marking Network Traffic

Table 12 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.

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 Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

**Note**

Table 12 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 12 Feature Information for Marking Network Traffic

Feature Name	Software Releases	Feature Configuration Information
Enhanced Packet Marking	12.2(13)T	<p>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.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Information About Marking Network Traffic, page 58 • How to Mark Network Traffic, page 65
Class-Based Marking	12.2(2)T	<p>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.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Information About Marking Network Traffic, page 58 • How to Mark Network Traffic, page 65
Quality of Service for Virtual Private Networks	12.2(2)T	<p>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.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Configuring QoS When Using IPSec VPNs, page 72 • Configuring QoS When Using IPSec VPNs: Example, page 77



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 a 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.

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 features. To find information about feature support and configuration, use the [“Feature Information for Classifying Network Traffic”](#) section on page 95.

Contents

- [Prerequisites for Classifying Network Traffic, page 81](#)
- [Information About Classifying Network Traffic, page 81](#)
- [How to Classify Network Traffic, page 85](#)
- [Configuration Examples for Classifying Network Traffic, page 92](#)
- [Additional References, page 93](#)
- [Glossary, page 94](#)
- [Feature Information for Classifying Network Traffic, page 95](#)

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, page 82](#)

- [Benefits of Classifying Network Traffic, page 82](#)
- [MQC and Network Traffic Classification, page 82](#)
- [Network Traffic Classification match Commands and Match Criteria, page 83](#)
- [Traffic Classification Compared with Traffic Marking, page 84](#)

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 a 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 Media Access Control (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 a 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 (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 “Modular Quality of Service Command-Line Interface” section of the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.3.

Network Traffic Classification match Commands and Match Criteria

As mentioned earlier, 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 13](#) lists the available **match** commands and the corresponding match criteria.

Table 13 *match Commands and Corresponding Match Criteria*

match Commands¹	Match Criteria
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	destination Media Access Control (MAC) address
match discard-class	discard class value
match dscp	differentiated services code point (DSCP) value
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) protocol 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 criteria value to use as an unsuccessful match criteria
match packet length (class-map)	Layer 3 packet length in the IP header
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	Real-Time Transport Protocol (RTP) traffic

Table 13 *match Commands and Corresponding Match Criteria (continued)*

match Commands¹	Match Criteria
match qos group	quality of service (QoS) group value
match source-address mac	source MAC address

1. Cisco IOS **match** commands can vary by release. Refer to the command documentation for the Cisco IOS release you are using for more information.

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 criteria 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 14](#) compares the features of traffic classification and traffic marking.

Table 14 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 <i>table-map-name</i> 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 85](#) (required)
- [Creating a Policy Map for Applying a QoS Feature to Network Traffic, page 86](#) (required)
- [Attaching the Policy Map to an Interface, page 88](#) (required)
- [Configuring QoS When Using IPsec VPNs, page 90](#) (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.

SUMMARY STEPS

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

**Note**

The **match fr-dlci** command classifies 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. Others can be used. For a list of other **match** commands, see [Table 13 on page 83](#).

5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	class-map <i>class-map-name</i> [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. <ul style="list-style-type: none">Enter the class map name.
Step 4	match fr-dlci <i>dlci-number</i> 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 Relay DLCI number. The match fr-dlci command is just an example of one of the match commands that can be used. Others can be used. For a list of other match commands, see Table 13 on page 83 .
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.

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*}



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 you want to use.

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. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	policy-map <i>policy-map-name</i> Example: Router(config)# policy-map policyl	Specifies the name of the policy map created earlier and enters policy-map configuration mode. <ul style="list-style-type: none"> Enter the policy map name.
Step 4	class { <i>class-name</i> class-default }	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. <ul style="list-style-type: none"> Enter the name of the class or enter the class-default keyword.
Step 5	bandwidth { <i>bandwidth-kbps</i> remaining percent <i>percentage</i> percent <i>percentage</i> } Example: Router(config-pmap-c)# bandwidth percent 50	(Optional) Specifies or modifies the bandwidth allocated for a class belonging to a policy map. <ul style="list-style-type: none"> Enter the amount of bandwidth as a number of kbps, a relative percentage of bandwidth, or an absolute amount of bandwidth. <p>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 you want to use.</p>

	Command or Action	Purpose
Step 6	<code>end</code> Example: Router(config-pmap-c)# end	Returns to privileged EXEC mode.
Step 7	<code>show policy-map</code> or <code>show policy-map policy-map class class-name</code> Example: Router# show policy-map or Example: Router# show policy-map policy1 class class7	(Optional) Displays all configured policy maps. or (Optional) Displays the configuration for the specified class of the specified policy map. <ul style="list-style-type: none">• Enter the policy map name and the class name.
Step 8	<code>exit</code> 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 86. Then attach the policy maps to the appropriate interface, following the instructions in the [“Attaching the Policy Map to an Interface”](#) section on page 88.

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. **exit**
8. **show policy-map interface** *interface-name*
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>enable</p> <p>Example: Router> enable</p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	<p>configure terminal</p> <p>Example: Router# configure terminal</p>	<p>Enters global configuration mode.</p>
Step 3	<p>interface <i>type number</i> [name-tag]</p> <p>Example: Router(config)# interface serial4/0</p>	<p>Configures an interface type and enters interface configuration mode.</p> <ul style="list-style-type: none"> Enter the interface type and number.
Step 4	<p>pvc [<i>name</i>] <i>vpi/vci</i> [ilmi qsaal smds l2transport]</p> <p>Example: Router(config-if)# pvc cisco 0/16</p>	<p>(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.</p> <ul style="list-style-type: none"> Enter the PVC name, the ATM network virtual path identifier, and the network virtual channel identifier. <p>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.</p>
Step 5	<p>exit</p> <p>Example: Router(config-atm-vc)# exit</p>	<p>(Optional) Returns to interface configuration mode.</p> <p>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.</p>
Step 6	<p>service-policy {input output} <i>policy-map-name</i></p> <p>Example: Router(config-if)# service-policy input policy1</p>	<p>Attaches a policy map to an input or output interface.</p> <ul style="list-style-type: none"> Enter the policy map name. <p>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.</p>

	Command or Action	Purpose
Step 7	<code>end</code> Example: Router(config-if)# end	Returns to privileged EXEC mode.
Step 8	<code>show policy-map interface interface-name</code> 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. <ul style="list-style-type: none"> • Enter the interface name.
Step 9	<code>exit</code> 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 [Cisco IOS Security Configuration Guide](#), Release 12.3.

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

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. `exit`

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	crypto map <i>map-name seq-num</i> Example: Router(config)# crypto map mymap 10	Enters crypto map configuration mode and creates or modifies a crypto map entry. <ul style="list-style-type: none">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 <i>type number [name-tag]</i> Example: Router(config)# interface serial4/0	Configures an interface type and enters interface configuration mode. <ul style="list-style-type: none">Enter the interface type and number.
Step 6	qos pre-classify Example: Router(config-if)# qos pre-classify	Enables QoS classification prior to IPSec encryption.
Step 7	exit Example: Router(config-if)# exit	(Optional) Exits interface configuration mode.

Configuration Examples for Classifying Network Traffic

This section contains the following examples:

- [Creating a Class Map for Classifying Network Traffic: Example, page 92](#)
- [Creating a Policy Map for Applying a QoS Feature to Network Traffic: Example, page 92](#)
- [Attaching the Policy Map to an Interface: Example, page 92](#)
- [Configuring QoS When Using IPsec VPNs: Example, page 93](#)

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
```

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 class7
Router# exit
```

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 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
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 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)# exit
```

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 , Release 12.3 T
MQC	“Modular Quality of Service Command-Line Interface” section of the Cisco IOS Quality of Service Solutions Configuration Guide , Release 12.3
Marking network traffic	“Marking Network Traffic” module
IPsec and VPNs	Cisco IOS Security Configuration Guide , Release 12.3
Network-Based Application Recognition (NBAR)	“Classification” section of the Cisco IOS Quality of Service Solutions Configuration Guide , Release 12.3
Committed Access Rate (CAR)	“Classification” section of the Cisco IOS Quality of Service Solutions Configuration Guide , Release 12.3
Policy-based routing	Cisco IOS Quality of Service Solutions Configuration Guide , Release 12.3
QoS policy propagation via Border Gateway Protocol (BGP)	Cisco IOS Quality of Service Solutions Configuration Guide , Release 12.3

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 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

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.

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.

**Note**

See *Internetworking Terms and Acronyms* for terms not included in this glossary.

Feature Information for Classifying Network Traffic

[Table 15](#) 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.

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 Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

**Note**

[Table 15](#) 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 15 Feature Information for Classifying Network Traffic

Feature Name	Software Releases	Feature Configuration Information
Packet Classification Using Frame Relay DLCI Number	12.2(13)T	<p>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.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Information About Classifying Network Traffic, page 81 • How to Classify Network Traffic, page 85
Packet Classification Based on Layer 3 Packet Length	12.2(13)T	<p>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.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Information About Classifying Network Traffic, page 81 • How to Classify Network Traffic, page 85
Quality of Service for Virtual Private Networks	12.2(2)T	<p>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.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Configuring QoS When Using IPSec VPNs, page 90 • Configuring QoS When Using IPSec VPNs: Example, page 93



Network-Based Application Recognition

This part consists of the following:

- [Configuring Network-Based Application Recognition](#)
- [Network-Based Application Recognition and Distributed Network-Based Application Recognition](#)
- [Network-Based Application Recognition Protocol Discovery Management Information Base](#)



Configuring Network-Based Application Recognition

This chapter describes the tasks for configuring the Network-Based Application Recognition (NBAR) feature.

For complete conceptual information, see the section [“Network-Based Application Recognition”](#) in the [“Classification Overview”](#) chapter in this book.

For a complete description of the NBAR commands mentioned in this chapter, refer to 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.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the [“Finding Additional Feature Support Information”](#) section on page lxix in the [“Using Cisco IOS Software for Release 12.4”](#) chapter in this book.

NBAR Configuration Task List

Your interface to NBAR is through the Modular QoS Command-Line Interface (Modular QoS CLI) feature. The Modular QoS CLI provides a model for QoS configuration under Cisco IOS software. The Modular QoS CLI provides a clean separation between the specification of a classification policy and the specification of other policies that act based on the results of the applied classification.

Configuring a QoS policy typically requires the configuration of traffic classes, the configuration of policies that will be applied to those traffic classes, and the attaching of policies to interfaces using the following commands:

- **class-map**
- **policy-map**
- **service-policy**

Use the **class-map** command to define one or more traffic classes by specifying the criteria by which traffic is classified.

Use the **policy-map** command to define one or more QoS policies (such as shaping, policing, and so on) to apply to traffic defined by a class map.

Use the **service-policy** command to attach a traffic to an interface on the router.

For additional information on the Modular QoS CLI, see the “[Modular Quality of Service Command-Line Interface Overview](#)” chapter in this book.

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

- [Configuring a Traffic Class](#) (Required)
- [Configuring a Traffic Policy](#) (Required)
- [Attaching a Traffic Policy to an Interface](#) (Required)
- [Verifying Traffic Policy Configuration](#) (Optional)
- [Monitoring and Maintaining NBAR](#) (Optional)


Note

You must enable Cisco Express Forwarding (CEF) on the router prior to configuring the NBAR feature. For information on how to enable CEF, refer to the *Cisco IOS Switching Services Configuration Guide*.

See the end of this chapter for the section “[NBAR Configuration Example](#).”

Configuring a Traffic Class

To configure a traffic class and the match criteria that will be used to identify traffic as belonging to that class, use the **class-map** global configuration command. For more information about match criteria, see the section “Creating a Traffic Class” in the chapter “[Modular Quality of Service Command-Line Interface Overview](#)” in this book. To define the match criteria, use the following commands beginning in global configuration mode.

In the following procedure, all traffic matching a specified protocol will be classified as belonging to the traffic class. The traffic class will classify traffic while the traffic policy configuration will determine how to treat the traffic.

For instance, if you wanted all FTP traffic to be marked with the QoS group value of 1, you would use the **match protocol ftp** command in class-map configuration mode, and use the **set qos-group 1** command in policy-map class configuration mode (assuming the traffic policy uses the specified class). Therefore, the classification purpose (classifying FTP traffic) would be handled in the traffic class, while the QoS feature (marking the QoS group value to 1) would be handled in the traffic policy.

	Command	Purpose
Step 1	Router(config)# class-map [match-all match-any] <i>class-name</i>	Specifies the user-defined name of the class map. The match-all option specifies that all match criteria in the class map must be matched. The match-any option specifies that one or more match criteria must match.
Step 2	Router(config-cmap)# match protocol <i>protocol-name</i>	Specifies a protocol supported by NBAR as a matching criterion.

Configuring a Traffic Policy

To specify the QoS policies to apply to traffic classes defined by a traffic class, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# policy-map <i>policy-name</i>	Specifies the traffic policy name entered by the user.
Step 2	Router(config-pmap)# class <i>class-name</i>	Specifies the name of a previously defined traffic class.
Step 3	Router(config-pmap-c)#	Enters policy-map class configuration mode, a prerequisite for entering QoS policies.

Attaching a Traffic Policy to an Interface

To attach a traffic policy to an interface and to specify the direction in which the traffic policy should be applied (on either packets coming into the interface or packets leaving the interface), use the following commands in interface configuration mode, as needed:

Command	Purpose
Router(config-if)# service-policy output <i>policy-map-name</i>	Specifies the name of the traffic policy to be attached where it can be applied to all packets leaving the interface.
Router(config-if)# service-policy input <i>policy-map-name</i>	Specifies the name of the traffic policy to be attached where it can be applied to all packets entering the interface.

To detach a policy map from an interface, use the **no service-policy [input | output] *policy-map-name*** command.

Verifying Traffic Policy Configuration

To display the configuration of a traffic policy and its associated traffic classes, use the following commands in privileged EXEC mode, as needed:

Command	Purpose
Router# show class-map	Displays all traffic class information.
Router# show class-map <i>class-name</i>	Displays the traffic class information of the user-specified traffic class.
Router# show policy-map	Displays all configured policy maps.
Router# show policy-map <i>policy-map-name</i>	Displays the user-specified policy map.
Router# show policy-map interface	Displays statistics and configurations of all input and output policies attached to an interface.
Router# show policy-map <i>interface-spec</i>	Displays configuration and statistics of the input and output policies attached to a particular interface.

NBAR Configuration Example

Command	Purpose
Router# show policy-map <i>interface-spec</i> [input]	Displays configuration and statistics of the input policy attached to an interface.
Router# show policy-map <i>interface-spec</i> [output]	Displays configuration statistics of the output policy attached to an interface.
Router# show policy-map interface-spec [input output] class <i>class-name</i>	Displays the configuration and statistics for the class name configured in the policy.

Monitoring and Maintaining NBAR

NBAR can determine which protocols and applications are currently running on a network. NBAR includes the Protocol Discovery feature that provides an easy way of discovering application protocols operating on an interface so that appropriate QoS policies can be developed and applied. With Protocol Discovery, you can discover any protocol traffic supported by NBAR and obtain statistics associated with that protocol.

To monitor and maintain the NBAR feature, use the following commands in privileged EXEC mode, as needed:

Command	Purpose
Router# show ip nbar port-map [<i>protocol-name</i>]	Displays the TCP/User Datagram Protocol (UDP) port number(s) used by NBAR to classify a given protocol.
Router# show ip nbar protocol-discovery	Displays the statistics for all interfaces on which Protocol Discovery is enabled.

NBAR Configuration Example

The following “[Configuring a Traffic Class with NBAR Example](#)” section provides an NBAR configuration example.

For information on how to configure NBAR, see the section “[NBAR Configuration Task List](#)” in this chapter.

Configuring a Traffic Class with NBAR Example

In the following example, the **class-map class1** command uses the NBAR classification of SQL*Net as its matching criterion:

```
Router(config)# class-map class1
Router(config-cmap)# match protocol sqlnet
```



Network-Based Application Recognition and Distributed Network-Based Application Recognition

Feature History

Cisco IOS Release	Modification
12.0(5)XE2	The NBAR feature was introduced. The first implementation of the NBAR feature was available on Cisco 7100 and Cisco 7200 series routers.
12.1(1)E	Support classification of HTTP traffic by host name for NBAR was introduced. The <i>variable-field-name value</i> options were also added to the match protocol command.
12.1(2)E	Support for the Citrix, Novadigm, and Printer protocols for NBAR was introduced.
12.1(5)T	This feature was introduced for the Cisco IOS Release 12.1 T train. NBAR became available on Cisco 2600 and 3600 series routers.
12.1(6)E	The dNBAR feature, which introduced NBAR functionality on the Cisco 7500 series router with a VIP and the Catalyst 6000 family switch with a FlexWAN module, was introduced.
12.1(10)EC	NBAR was introduced for Cisco 7100 uBR and Cisco 7200 uBR routers.
12.1(11b)E	The match protocol rtp command was introduced on the Cisco IOS Release 12.1 E train.
12.1(12c)E	The match protocol gnutella and match protocol fasttrack commands were added because Gnutella and FastTrack became available as NBAR-supported protocols.
12.1(13)E	NBAR was released on the Catalyst 6000 family switch without a FlexWAN module.
12.2(2)T	This feature was introduced on Cisco 1700 series routers.
12.2(4)T3	The dNBAR feature introduced NBAR functionality on the Cisco IOS Release 12.2 T train. This feature was introduced for the Cisco 7500 series router with a VIP only.

Feature History (continued)

Cisco IOS Release	Modification
12.2(8)T	<p>The match protocol rtp command was introduced, allowing NBAR to classify Real-Time Transport Protocol (RTP) traffic.</p> <p>The Cisco 3700 also became available. The initial release of the Cisco 3700 supported NBAR.</p>
12.2(14)S	NBAR and dNBAR were introduced in Cisco IOS Release 12.2 S. The 12.2 S version of NBAR includes everything available on the 12.1 E and 12.2 T implementations of NBAR with the exception of platform support for platforms not supported by 12.2 S.
12.3(4)T	<p>NBAR PDLM Versioning was introduced. This feature introduced versioning of PDLM protocols and the show ip nbar version command. See the “IP NBAR PDLM Module Versioning” section on page 114 for additional information regarding this feature.</p> <p>The NBAR User-Defined Custom Application Classification feature was introduced. See the “Classification of Custom Applications” section on page 112 for additional information on the enhancements to the custom protocol that were introduced as part of this feature.</p> <p>The NBAR Extended Inspection for HTTP Traffic feature was introduced. This feature allows NBAR to scan TCP ports that are not well-known and identify HTTP traffic traversing these ports.</p>
12.3(2)XE	<p>NBAR was introduced on Cisco 800 series routers.</p> <p>To see if NBAR is supported in other platforms, see the “Supported Platforms” section on page 122 of this document.</p>
12.3(7)T	Restrictions on the number of bytes of payload that could be inspected by NBAR were removed. NBAR can now inspect the full packet payload.
12.3(8)T	NBAR was introduced on Cisco 800 series routers running Cisco IOS Release 12.3 T.
12.3(14)T	The variable <i>field-name field-length</i> keyword was added to the ip nbar custom protocol command. This option allowed for sub-classification of custom traffic in NBAR.

This document provides information for the Network-Based Application Recognition (NBAR) and the Distributed Network-Based Application Recognition (dNBAR) features. This document contains all of the updates made to the NBAR and dNBAR features.

Before proceeding, it is important to note that the dNBAR feature, which introduced NBAR on the Cisco 7500 with a Versatile Interface Processor (VIP) and the Catalyst 6000 family switch with a FlexWAN module, is identical in implementation to NBAR. Therefore, unless otherwise noted, the term NBAR is used throughout this document to describe both the NBAR and dNBAR feature. The term dNBAR is used only when appropriate.

This document includes information on the benefits of NBAR, supported platforms, restrictions, definitions, and new and revised command syntax.

This document includes the following sections:

- [Feature Overview, page 105](#)
- [Supported Platforms, page 122](#)
- [Supported Standards, MIBs, and RFCs, page 123](#)
- [Prerequisites, page 125](#)
- [Configuration Tasks, page 125](#)
- [Monitoring and Maintaining NBAR, page 129](#)
- [Configuration Examples, page 129](#)
- [Command Reference, page 130](#)
- [Glossary, page 130](#)
- [Appendix, page 130](#)

Feature Overview

The purpose of IP Quality of Service (QoS) is to provide appropriate network resources (bandwidth, delay, jitter, and packet loss) to applications. QoS maximizes the return on investments on network infrastructure by ensuring that mission critical applications get the required performance and noncritical applications do not hamper the performance of critical applications.

IP QoS can be deployed by defining classes or categories of applications. These classes are defined by using various classification techniques available in Cisco IOS software. After these classes are defined and attached to an interface, the desired QoS features, such as Marking, Congestion Management, Congestion Avoidance, Link Efficiency mechanisms, or Policing and Shaping can then be applied to the classified traffic to provide the appropriate network resources amongst the defined classes.

Classification, therefore, is an important first-step in configuring QoS in a network infrastructure.

NBAR is a classification engine that recognizes a wide variety of applications, including web-based and other difficult-to-classify protocols that utilize dynamic TCP/UDP port assignments. When an application is recognized and classified by NBAR, a network can invoke services for that specific application. NBAR ensures that network bandwidth is used efficiently by classifying packets and then applying Quality of Service (QoS) to the classified traffic. Some examples of class-based QoS features that can be used on traffic after the traffic is classified by NBAR include:

- Class-Based Marking (the **set** command)
- Class-Based Weighted Fair Queueing (the **bandwidth** and **queue-limit** commands)
- Low Latency Queueing (the **priority** command)
- Traffic Policing (the **police** command)
- Traffic Shaping (the **shape** command)



Note

For an animated example of NBAR being used with other QoS features to solve a network problem, click [here](#).

**Note**

The NBAR feature is used for classifying traffic by protocol. The other class-based QoS features determine how the classified traffic is forwarded and are documented separately from NBAR. Furthermore, NBAR is not the only method of classifying network traffic so that QoS features can be applied to classified traffic.

For information on the class-based features that can be used to forward NBAR-classified traffic, see the individual feature modules for the particular class-based feature as well as the *Cisco IOS Quality of Service Solutions Guide*.

Many of the non-NBAR classification options for QoS are documented in the “Modular Quality of Service Command-Line Interface” section of the *Cisco IOS Quality of Service Solutions Guide*. These commands are configured using the **match** command in class map configuration mode.

NBAR introduces several new classification features that identify applications and protocols from Layer 4 through Layer 7:

- Statically assigned TCP and UDP port numbers
- Non-UDP and non-TCP IP protocols
- Dynamically assigned TCP and UDP port numbers. Classification of such applications requires stateful inspection; that is, the ability to discover the data connections to be classified by parsing the connections where the port assignments are made.
- Sub-port classification or classification based on deep packet inspection; that is, classification by looking deeper into the packet.

NBAR can classify static port protocols. Although access control lists (ACLs) can also be used for this purpose, NBAR is easier to configure and can provide classification statistics that are not available when using ACLs.

NBAR includes a Protocol Discovery feature that provides an easy way to discover application protocols that are transversing an interface. The Protocol Discovery feature discovers any protocol traffic supported by NBAR. Protocol Discovery maintains the following per-protocol statistics for enabled interfaces: total number of input and output packets and bytes, and input and output bit rates. The Protocol Discovery feature captures key statistics associated with each protocol in a network that can be used to define traffic classes and QoS policies for each traffic class.

Benefits

Ability to Identify and Classify Network Traffic by Protocol

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 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 implement classes of traffic. These classes of traffic can then be used to provide different levels of service for network traffic, therefore allowing better network management by providing the right level of network resources for network traffic.

NBAR Application Notes

The following section provides information on several topics that could be useful to individuals configuring NBAR in their networks. The following topics are covered in this section:

- [Catalyst 6000 Family Switches without FlexWAN Modules Application Notes](#)
- [Packet Description Language Module](#)
- [Classification of HTTP by URL, Host, or MIME](#)
- [Classification of Citrix ICA Traffic by Application Name](#)
- [RTP Payload Type Classification](#)
- [Classification of Custom Applications](#)
- [Classification of Peer-to-Peer File-Sharing Applications](#)
- [IP NBAR PDL Module Versioning](#)
- [Supported Protocols](#)

Catalyst 6000 Family Switches without FlexWAN Modules Application Notes

When NBAR is enabled on a Catalyst 6000 without a FlexWAN module interface, all traffic flows entering or leaving the NBAR-enabled interface will be processed in software on the Multilayer Switch Feature Card 2 (MSFC2).

The following other restrictions should also be noted when running NBAR:

- NBAR can only be implemented on an MSFC2 with Supervisor Engine 1 or Supervisor Engine 2.
- NBAR Protocol Discovery or QoS service policies using NBAR to match protocols cannot co-exist on an interface that contains Catalyst 6000-specific QoS actions. Refer to the Catalyst 6000 QoS Guide for Catalyst 6000-specific QoS actions (at the time of this publication, the current Catalyst 6000-specific QoS actions were **police** and **trust**, but please refer to the Catalyst 6000 QoS Guide for additional information).

The following table provides configuration results when NBAR is added to an interface. The results vary depending on the current configuration of the policy map on the interface.

Table 16 NBAR Behavior Descriptions

Current Policy Map State	Action	Result
At least one service policy with platform-specific QoS action in the policy map is attached to interface.	Enable Protocol Discovery on the interface.	Protocol Discovery is rejected.
No service policies on the interface have NBAR or a platform-specific QoS action in the policy map.	Enable Protocol Discovery on the interface.	Protocol Discovery is accepted, but the service policy is disabled from the interface.
A service policy on the interface contains match protocol NBAR commands.	Enable Protocol Discovery on the interface.	Protocol Discovery is accepted.

Current Policy Map State	Action	Result
No policy map is on the interface.	Enable Protocol Discovery on the interface.	The command is accepted. Traffic is processed on the MSFC2 once the command is accepted.
No policy map is on the interface	Disable Protocol Discovery.	The command is accepted. Traffic is no longer processed on the MSFC2.
No service policies on the interface have platform-specific QoS actions or match protocol NBAR commands.	Disable Protocol Discovery.	Protocol Discovery is disabled. The service policy is removed from the interface. The service policy can be reattached.
At least one service policy on the interface is using the match protocol NBAR command.	Disable Protocol Discovery.	Protocol Discovery is disabled.
A service policy with a platform-specific QoS action and Protocol Discovery is enabled on the interface.	Attach the service policy to an interface.	Reject the service policy. Protocol Discovery and platform-specific QoS actions cannot be enabled in the same policy map.
Protocol Discovery is enabled on an interface and the service policy has a non-platform specific QoS action.	Attach the service policy to an interface.	The policy map is attached. The policy map has to be attached in IOS QoS mode.
No match protocol NBAR commands are in any service policy on the interface and Protocol Discovery is not enabled.	Attach the service policy to an interface.	The policy map is attached in Catalyst 6000 QoS mode.
Protocol Discovery is not enabled on the interface and match protocol NBAR commands are in at least one service policy on the interface.	Attach the service policy to an interface.	The service policy is attached in IOS mode and traffic is processed using the MSFC2.
A service policy that has no match protocol NBAR commands and no Protocol Discovery needs to be removed from the interface. The interface contains no other service policies that contain match protocol NBAR commands or Protocol Discovery.	Detach the service policy from an interface	The service policy is detached like any other service policy.

Current Policy Map State	Action	Result
A service policy with match protocol NBAR commands needs to be detached from the interface. Another service policy attached in the opposite direction does not contain match protocol NBAR commands. No Protocol Discovery is enabled on the interface.	Detach the service policy with match protocol NBAR commands from the interface.	The service policy is detached and the other service policy in the opposite direction is also removed. Traffic is no longer processed using the MSFC2.
A service policy contains match protocol NBAR commands and the service policy in the other direction needs match protocol NBAR or Protocol Discovery needs to be enabled on the interface.	Detach the service policy from the interface.	The service policy is detached. Continue to process traffic on the MSFC2 so that match protocol can be enabled on the other service policy or Protocol Discovery can be enabled on the interface.
A service policy contains match protocol NBAR commands. No other service policies are on the interface and Protocol Discovery is not enabled.	Detach the service policy from the interface.	Service policy is detached. Traffic is no longer processed on the MSFC2.

Packet Description Language Module

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

New PDLMs will only be released by Cisco and can be loaded from Flash memory. Please contact your local Cisco representative to request additions or changes to the set of protocols classified by NBAR.

To view a list of currently available PDLMs or to download a PDLM, go to the following URL:

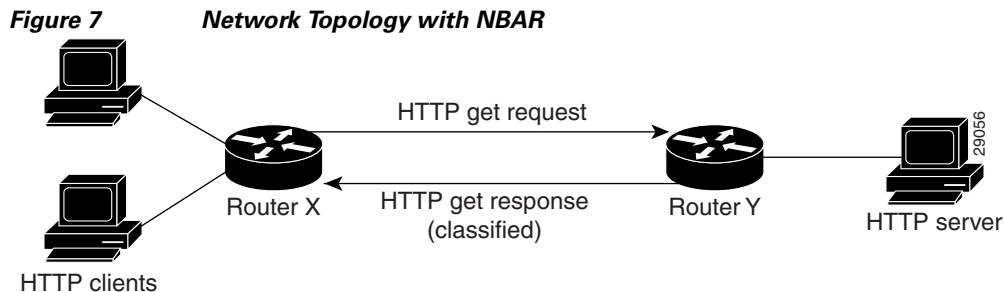
<http://www.cisco.com/cgi-bin/tablebuild.pl/pdlm>

Classification of HTTP 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 on content within the payload such as transaction identifier, message type, or other similar data.

Classification of HTTP 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 URL matching in NBAR supports most HTTP request methods such as GET, PUT, HEAD, POST, DELETE, and TRACE. NBAR uses the UNIX filename specification as the basis for the URL or host specification format. The NBAR engine then converts the specified match string into a regular expression.

NBAR recognizes HTTP packets containing the URL and classifies all packets that are sent to the source of the HTTP request. [Figure 7](#) illustrates a network topology with NBAR in which Router Y is the NBAR-enabled router.



When specifying a URL for classification, include only the portion of the URL following the `www.hostname.domain` in the match statement. For example, for the URL `www.cisco.com/latest/whatsnew.html`, include only `/latest/whatsnew.html`.

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 Internet Assigned Numbers Authority (IANA)-supported MIME types can be found at:

<ftp://ftp.isi.edu/in-notes/iana/assignments/media-types/media-types>

In MIME type matching, NBAR classifies the packet containing the MIME type and all subsequent packets, which are sent to the source of the HTTP request.

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 identify HTTP traffic traversing these ports. HTTP traffic classifications are no longer restrained to the well-known and defined TCP ports.

Classification of Citrix ICA Traffic by Application Name

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

NBAR statefully tracks Citrix ICA server client messages and classifies requests for given Citrix application names and traffic. A Citrix application is named when published on a Citrix ICA server. NBAR performs a regular expression match using a user-specified application name string on the contents of the Citrix ICA control packets carrying the published application name. Therefore, users need to specify a regular expression that will result in a match for the published application name if they want to match a specified application. See the **match protocol citrix** command in the “[Command Reference](#)” section for additional information.

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 only be used to classify Citrix applications 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 non-seamless 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 non-session 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 non-session sharing mode, each application for each particular client uses a separate TCP connection. NBAR can provide interapplication differentiation in seamless non-session sharing mode.

Session sharing can be turned off using the following steps:

Step 1 At the command prompt of the Citrix server, open the registry editor by entering the **regedit** command.

Step 2 Create the following registry entry (which overrides session sharing):

```
[HKLM]\SYSTEM\CurrentControlSet\Control\Citrix\WFSHELL\TWI
Value name: "SeamlessFlags", type DWORD, possible value 0 or 1
```

Setting this registry value to 1 overrides session sharing. Note that this flag is SERVER GLOBAL.



Note

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

RTP Payload Type Classification

RTP is a packet format for multimedia data streams. It can be used for media-on-demand as well as 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). 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 providing support for applications with real-time properties such as continuous media (such as audio and video), which includes timing reconstruction, loss detection, and security and content identification. RTP is discussed in RFC 1889 and RFC 1890.

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 it also can differentiate on the basis of audio and video CODECs to provide more granular Quality of Service. The RTP Payload Type Classification feature, therefore, looks deep into the RTP header to classify RTP packets.

NBAR RTP Payload Type Classification was first introduced in Cisco IOS Release 12.2(8)T and is also available in Cisco IOS Release 12.1(11b)E.

Classification of Custom Applications

The custom protocol supports static port-based protocols and applications that are not currently supported in NBAR. This functionality allows mapping of static TCP and UDP port numbers to custom protocol within NBAR. The custom protocol is also available as a PDLM if your version of Cisco IOS supports NBAR but not the custom protocol.

The initial custom NBAR application had the following features that were later enhanced in Cisco IOS Release 12.3(4)T:

- The custom protocol had to be named custom-xx, with xx being a number.
- 10 custom applications can be assigned using NBAR, and each customer application can have up to 16 TCP and 16 UDP ports each mapped to the individual custom protocol. The real-time statistics of each custom protocol can be monitored using Protocol Discovery.

In Cisco IOS Release 12.3(4)T, the User-Defined Custom Application Classification feature was introduced and the following enhancements to custom protocols were introduced:

- The ability to inspect the payload for certain matching string patterns at a specific offset.
- The ability to allow users to define the names of their custom protocol applications. The user-named protocol can then be used by Protocol Discovery, the Protocol Discovery MIB, **match protocol**, or **ip nbar port-map** as an NBAR-supported protocol.
- The ability to allow NBAR inspection for custom protocols to be specified by direction of traffic (traffic heading toward a source or destination rather than defaulting to traffic in both directions) if desired by user.
- Provides CLI support that allows a user configuring a custom application to specify a range of ports rather than have to enter each port individually.

For additional information on the enhancements to the custom protocol that were introduced in Cisco IOS Release 12.3(4)T, see the **ip nbar custom** command reference in this document.

In Cisco IOS Release 12.3(14)T, the **variable field-name field-length** option was added to the **ip nbar custom protocol** command. This option allows for NBAR classification and identification on a specific value within the custom payload (after creating a variable while creating the custom protocol, you can use the **match protocol custom-protocol-name variable-name value** to classify traffic based on a specific value in the custom protocol). See the **ip nbar custom** command reference in this document for additional information on this option.

Pre-12.3(4)T Custom Application Example

In the following example, a gaming application that runs on TCP port 8877 needs to be classified using NBAR. You can use custom-01 to map TCP port 8877 by entering the following command:

```
Router(config)# ip nbar port-map custom-01 tcp 8877
```

It is important to note that this configuration is also supported on Cisco IOS releases released after Release 12.3(4)T but is required on all prior releases.

12.3(4)T and Later Custom Application Examples

In the following example, the custom protocol app_sales1 will identify TCP packets with a source port of 4567 and contain the term “SALES” in the fifth byte of the payload:

```
ip nbar custom app_sales1 5 ascii SALES source tcp 4567
```

In the following example, the custom protocol virus_home will identify UDP packets with a destination port of 3000 and contain “0x56” in the seventh byte of the payload:

```
ip nbar custom virus_home 7 hex 0x56 dest udp 3000
```

In the following example, custom protocol media_new will identify TCP packets with a destination or source port of 4500 and that have a value of 90 at the sixth byte of the payload:

```
ip nbar custom media_new 6 decimal 90 tcp 4500
```

In the following example, custom protocol msn1 will look for TCP packets with a destination or source port of 6700:

```
ip nbar custom msn1 tcp 6700
```

In the following example, custom protocol mail_x will look for UDP packets with a destination port of 8202:

```
ip nbar custom mail_x destination udp 8202
```

In the following example, custom protocol mail_y will look for UDP packets with destination ports between 3000 and 4000 including 3000 and 4000 as well as port 5500:

```
ip nbar custom mail_y destination udp range 3000 4000 5500
```

Classification of Peer-to-Peer File-Sharing Applications

Gnutella and FastTrack are peer-to-peer file-sharing protocols that became classifiable using NBAR in Cisco IOS Release 12.1(12c)E.

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 *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 this 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 based on a filename extension or on 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*"
```

Applications that use FastTrack include KaZaA, Grokster, and Morpheus (although newer versions of Morpheus use Gnutella).

Some of the applications that use Gnutella include:

- BearShare
- Gnewtellium
- Gnucleus
- Gtk-Gnutella
- JTella
- LimeWire
- Morpheus
- Mutella
- Phex
- Qtella
- Swapper
- XoloX
- XCache

IP NBAR PDLM Module Versioning

A Packet Description Language Module (PDLM) is used to add a new protocol to the list of supported NBAR protocols. Before downloading PDLMs, users should understand some of the interdependencies between the versioning of NBAR in the Cisco IOS code and the PDLM file itself. The following definitions help define some of the aspects of NBAR and PDLM versioning and the interdependencies required between the two before a new protocol can be supported in NBAR via a PDLM download.

The following version numbers are kept by the Cisco IOS software:

- NBAR Software Version—This is the version of NBAR software running on the current version of Cisco IOS.
- Resident Module Version—This is the version of the NBAR-supported PDLM protocol. The Resident Module Version must be less than the NBAR PDLM Interdependency Version of the PDLM for a PDLM file to be downloaded from cisco.com and accepted within NBAR in the IOS software.

The following version numbers is kept by the PDLM:

- NBAR Software Version—The minimum version of the NBAR software required to load this PDLM.

See the **show ip nbar version** command reference in this document for additional information on IP NBAR PDLM Module Versioning.

Supported Protocols

NBAR is capable of classifying the following three 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 dynamically assign port numbers and therefore require stateful inspection. This table includes packets that require sub-port classification and classification based on deep packet inspection.

Table 17 *Non-UDP and Non-TCP Protocols*

Protocol	Type	Well-Known Port Number	Description	Syntax	Cisco IOS Release ¹
EGP	IP	8	Exterior Gateway Protocol	egp	12.0(5)XE2 12.1(1)E 12.1(5)T
EIGRP	IP	88	Enhanced Interior Gateway Routing Protocol	eigrp	12.0(5)XE2 12.1(1)E 12.1(5)T
GRE	IP	47	Generic Routing Encapsulation	gre	12.0(5)XE2 12.1(1)E 12.1(5)T
ICMP	IP	1	Internet Control Message Protocol	icmp	12.0(5)XE2 12.1(1)E 12.1(5)T
IPINIP	IP	4	IP in IP	ipinip	12.0(5)XE2 12.1(1)E 12.1(5)T
IPSec	IP	50, 51	IP Encapsulating Security Payload/Authentication Header	ipsec	12.0(5)XE2 12.1(1)E 12.1(5)T

1. Indicates the Cisco IOS maintenance release that first supported the protocol. This table is updated when a protocol is added to a new Cisco IOS release train.

Table 18 TCP and UDP Static Port Protocols

Protocol	Type	Well-Known Port Number	Description	Syntax	Cisco IOS Release ¹
BGP	TCP/UDP	179	Border Gateway Protocol	bgp	12.0(5)XE2 12.1(1)E 12.1(5)T
CU-SeeMe	TCP/UDP	7648, 7649	Desktop videoconferencing	cuseeme	12.0(5)XE2 12.1(1)E 12.1(5)T
CU-SeeMe	UDP	24032	Desktop video conferencing	cuseeme	12.0(5)XE2 12.1(1)E 12.1(5)T
DHCP/BOOTP	UDP	67, 68	Dynamic Host Configuration Protocol/Bootstrap Protocol	dhcp	12.0(5)XE2 12.1(1)E 12.1(5)T
DNS	TCP/UDP	53	Domain Name System	dns	12.0(5)XE2 12.1(1)E 12.1(5)T
Finger	TCP	79	Finger user information protocol	finger	12.0(5)XE2 12.1(1)E 12.1(5)T
Gopher	TCP/UDP	70	Internet Gopher Protocol	gopher	12.0(5)XE2 12.1(1)E 12.1(5)T
HTTP	TCP	80 ²	Hypertext Transfer Protocol	http	12.0(5)XE2 12.1(1)E 12.1(5)T
HTTPS	TCP	443	Secured HTTP	secure-http	12.0(5)XE2 12.1(1)E 12.1(5)T
IMAP	TCP/UDP	143, 220	Internet Message Access Protocol	imap	12.0(5)XE2 12.1(1)E 12.1(5)T
IRC	TCP/UDP	194	Internet Relay Chat	irc	12.0(5)XE2 12.1(1)E 12.1(5)T
Kerberos	TCP/UDP	88, 749	Kerberos Network Authentication Service	kerberos	12.0(5)XE2 12.1(1)E 12.1(5)T
L2TP	UDP	1701	L2F/L2TP tunnel	l2tp	12.0(5)XE2 12.1(1)E 12.1(5)T

Table 18 *TCP and UDP Static Port Protocols (continued)*

Protocol	Type	Well-Known Port Number	Description	Syntax	Cisco IOS Release¹
LDAP	TCP/UDP	389	Lightweight Directory Access Protocol	ldap	12.0(5)XE2 12.1(1)E 12.1(5)T
MS-PPTP	TCP	1723	Microsoft Point-to-Point Tunneling Protocol for VPN	pptp	12.0(5)XE2 12.1(1)E 12.1(5)T
MS-SQLServer	TCP	1433	Microsoft SQL Server Desktop Videoconferencing	sqlserver	12.0(5)XE2 12.1(1)E 12.1(5)T
NetBIOS	TCP	137, 139	NetBIOS over IP (MS Windows)	netbios	12.0(5)XE2 12.1(1)E 12.1(5)T
NetBIOS	UDP	137, 138	NetBIOS over IP (MS Windows)	netbios	12.0(5)XE2 12.1(1)E 12.1(5)T
NFS	TCP/UDP	2049	Network File System	nfs	12.0(5)XE2 12.1(1)E 12.1(5)T
NNTP	TCP/UDP	119	Network News Transfer Protocol	nntp	12.0(5)XE2 12.1(1)E 12.1(5)T
Notes	TCP/UDP	1352	Lotus Notes	notes	12.0(5)XE2 12.1(1)E 12.1(5)T
Novadigm	TCP/UDP	3460-3465	Novadigm Enterprise Desktop Manager (EDM)	novadigm	12.1(2)E 12.1(5)T
NTP	TCP/UDP	123	Network Time Protocol	ntp	12.0(5)XE2 12.1(1)E 12.1(5)T
PCAnywhere	TCP	5631, 65301	Symantec PCAnywhere	pcanywhere	12.0(5)XE2 12.1(1)E 12.1(5)T
PCAnywhere	UDP	22, 5632	Symantec PCAnywhere	pcanywhere	12.0(5)XE2 12.1(1)E 12.1(5)T
POP3	TCP/UDP	110	Post Office Protocol	pop3	12.0(5)XE2 12.1(1)E 12.1(5)T

Table 18 *TCP and UDP Static Port Protocols (continued)*

Protocol	Type	Well-Known Port Number	Description	Syntax	Cisco IOS Release¹
Printer	TCP/UDP	515	Printer	printer	12.1(2)E 12.1(5)T
RIP	UDP	520	Routing Information Protocol	rip	12.0(5)XE2 12.1(1)E 12.1(5)T
RSVP	UDP	1698, 1699	Resource Reservation Protocol	rsvp	12.0(5)XE2 12.1(1)E 12.1(5)T
SFTP	TCP	990	Secure FTP	secure-ftp	12.0(5)XE2 12.1(1)E 12.1(5)T
SHTTP	TCP	443	Secure HTTP	secure-http	12.0(5)XE2 12.1(1)E 12.1(5)T
SIMAP	TCP/UDP	585, 993	Secure IMAP	secure-imap	12.0(5)XE2 12.1(1)E 12.1(5)T
SIRC	TCP/UDP	994	Secure IRC	secure-irc	12.0(5)XE2 12.1(1)E 12.1(5)T
SLDAP	TCP/UDP	636	Secure LDAP	secure-ldap	12.0(5)XE2 12.1(1)E 12.1(5)T
SMTP	TCP	25	Simple Mail Transfer Protocol	smtp	12.0(5)XE2 12.1(1)E 12.1(5)T
SNMP	TCP/UDP	161, 162	Simple Network Management Protocol	snmp	12.0(5)XE2 12.1(1)E 12.1(5)T
SNNTTP	TCP/UDP	563	Secure NNTP	secure-nntp	12.0(5)XE2 12.1(1)E 12.1(5)T
SOCKS	TCP	1080	Firewall security protocol	socks	12.0(5)XE2 12.1(1)E 12.1(5)T
SPOP3	TCP/UDP	995	Secure POP3	secure-pop3	12.0(5)XE2 12.1(1)E 12.1(5)T
SSH	TCP	22	Secured Shell	ssh	12.0(5)XE2 12.1(1)E 12.1(5)T

Table 18 *TCP and UDP Static Port Protocols (continued)*

Protocol	Type	Well-Known Port Number	Description	Syntax	Cisco IOS Release ¹
STELNET	TCP	992	Secure Telnet	secure-telnet	12.0(5)XE2 12.1(1)E 12.1(5)T
Syslog	UDP	514	System Logging Utility	syslog	12.0(5)XE2 12.1(1)E 12.1(5)T
Telnet	TCP	23	Telnet Protocol	telnet	12.0(5)XE2 12.1(1)E 12.1(5)T
X Windows	TCP	6000-6003	X11, X Windows	xwindows	12.0(5)XE2 12.1(1)E 12.1(5)T

1. Indicates the Cisco IOS maintenance release that first supported the protocol. This table is updated when a protocol is added to a new Cisco IOS release train.
2. In 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 identify HTTP traffic traversing these ports.

Table 19 *TCP and UDP Stateful Protocols*

Protocol	Type	Description	Syntax	Cisco IOS Release ¹
Citrix ICA	TCP/ UDP	Citrix ICA traffic by application name	citrix citrix app	12.1(2)E 12.1(5)T
FTP	TCP	File Transfer Protocol	ftp	12.0(5)XE2 12.1(1)E 12.1(5)T
Exchange	TCP	MS-RPC for Exchange	exchange	12.0(5)XE2 12.1(1)E 12.1(5)T
FastTrack		FastTrack For a list of common FastTrack applications, see the “ Classification of Peer-to-Peer File-Sharing Applications ” section of this document.	fasttrack	12.1(12c)E
Gnutella	TCP	Gnutella For a list of common Gnutella applications, see the “ Classification of Peer-to-Peer File-Sharing Applications ” section of this document.	gnutella	12.1(12c)E

Table 19 *TCP and UDP Stateful Protocols (continued)*

Protocol	Type	Description	Syntax	Cisco IOS Release¹
HTTP	TCP	HTTP with URL, MIME, or host classification	http	12.0(5)XE2 12.1(1)E 12.1(5)T (HTTP host classification is not available on the 12.0 XE release train)
Napster	TCP	Napster traffic	napster	12.1(5)T
Netshow	TCP/ UDP	Microsoft Netshow	netshow	12.0(5)XE2 12.1(1)E 12.1(5)T
r-commands	TCP	rsh, rlogin, rexec	rcmd	12.0(5)XE2 12.1(1)E 12.1(5)T
RealAudio	TCP/ UDP	RealAudio Streaming Protocol	realaudio	12.0(5)XE2 12.1(1)E 12.1(5)T
RTP	TCP/ UDP	Real-Time Transport Protocol Payload Classification	rtp	12.2(8)T
SQL*NET	TCP/ UDP	SQL*NET for Oracle	sqlnet	12.0(5)XE2 12.1(1)E 12.1(5)T
StreamWorks	UDP	Xing Technology Stream Works audio and video	streamwork	12.0(5)XE2 12.1(1)E 12.1(5)T
SunRPC	TCP/ UDP	Sun Remote Procedure Call	sunrpc	12.0(5)XE2 12.1(1)E 12.1(5)T
TFTP	UDP	Trivial File Transfer Protocol	tftp	12.0(5)XE2 12.1(1)E 12.1(5)T
VDOLive	TCP/ UDP	VDOLive Streaming Video	vdolive	12.0(5)XE2 12.1(1)E 12.1(5)T

1. Indicates the Cisco IOS maintenance release that first supported the protocol. This table is updated when a protocol is added to a new Cisco IOS release train.

Restrictions

The NBAR feature does not support the following:

- More than 24 concurrent URLs, hosts, or MIME type matches
- Matching beyond the first 400 bytes in a packet payload 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 support full payload inspection. The only exception is that NBAR can only inspect custom protocol traffic for 255 bytes into the payload.
- Non-IP traffic
- MPLS-labelled packets. NBAR only classifies IP packets. You can, however, use NBAR to classify IP traffic before the traffic is handed over to MPLS. Use the Modular QoS CLI (MQC) to set the IP DSCP field on the NBAR-classified packets and make MPLS map the DSCP setting to the MPLS 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 originating from or destined to the router running NBAR

NBAR is not supported on the following logical interfaces:

- Fast EtherChannel
- Interfaces where tunneling or encryption is used
- NBAR was not supported on Dialer interfaces until Cisco IOS Release 12.2(4)T



Note

NBAR cannot be used to classify output traffic on a WAN link where tunneling or encryption is used. Therefore, NBAR should be configured on other interfaces on the router (such as a LAN link) to perform input classification before the traffic is switched to the WAN link for output.

However, NBAR Protocol Discovery is supported on interfaces where tunneling or encryption is used. You can enable Protocol Discovery directly on the tunnel or on the interface where encryption is performed to gather key statistics on the various applications that are traversing the interface. The input statistics also show the total number of encrypted/tunneled packets received in addition to the per-protocol breakdowns.

In order to run Distributed NBAR on a Cisco 7500 series router, you must be using a processor that has 64 MB of DRAM or more. At the time of this publication, the following processors met this requirement:

- VIP2-50, VIP4-50, VIP4-80, and VIP6-80
- GEIP and GEIP+
- SRPIP

Memory Management

NBAR uses approximately 150 bytes of DRAM for each flow that requires stateful inspection. (See [Table 19](#) for a list of stateful protocols supported by NBAR that require stateful inspection.) When NBAR is configured, it allocates 1 MB of DRAM to support up to 5000 concurrent flows. NBAR checks to see if it needs more memory to handle additional concurrent stateful flows. If such a need is detected, NBAR expands its memory usage in increments of 200 Kb to 400 Kb.

Related Features and Technologies

- Access control lists (ACLs)
- Traffic Policing
- Traffic Shaping
- Class-Based Weighted Fair Queueing (CBWFQ)
- Class-Based Marking
- Low Latency Queueing
- Modular Quality of Service Command-Line Interface (Modular QoS CLI)

Related Documents

- NBAR animation
- *Quality of Service (QoS) Networking*
- *Quality of Service Solutions Configuration Guide*
- *Quality of Service Solutions Command Reference*
- *Access Control Lists: Overview and Guidelines*
- *Network-Based Application Recognition Management Information Base* document

Supported Platforms

To view the platforms that support NBAR and when NBAR support was introduced, check Feature Navigator.

Determining Platform Support Through Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Feature Navigator. Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

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.

To access 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 at <http://www.cisco.com/register>.

Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Feature Navigator home page at the following URL:

<http://www.cisco.com/go/fn>

Supported Standards, MIBs, and RFCs

Standards

- 0009, *File Transfer Protocol (FTP)*
- 0013, *Domain Names - Concepts and Facilities*
- 0033, *The TFTP Protocol (Revision 2)*
- 0034, *Routing Information Protocol*
- 0053, *Post Office Protocol - Version 3*
- 0056, *RIP Version 2*

MIBs

The CISCO-NBAR-PROTOCOL-DISCOVERY MIB is a MIB that utilizes Cisco NBAR Protocol Discovery in SNMP. For information on the CISCO-NBAR-PROTOCOL-DISCOVERY MIB, see the *Network-Based Application Recognition Management Information Base* document.

To obtain lists of supported MIBs by platform and Cisco IOS Release, and to download MIB modules, go to the Cisco MIB web site on cisco.com at the following URL:

<http://www.cisco.com/public/sw-center/netmgmt/ctmk/mibs.shtml>.

RFCs

- RFC 742, *NAME/FINGER Protocol*
- RFC 759, *Internet Message 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 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 2060, *Internet Message Access Protocol - Version 4rev1*
- RFC 2068, *Hypertext Transfer Protocol -- HTTP/1.1*
- RFC 2131, *Dynamic Host Configuration Protocol*
- 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*

Prerequisites

CEF

You must enable Cisco Express Forwarding (CEF) before you configure NBAR. For more information on CEF, refer to the Cisco IOS Release 12.2 *Cisco IOS Switching Services Configuration Guide*.

Configuration Tasks

The NBAR feature has two components: one component monitors applications traversing a network, and the other that classifies traffic by protocol.

In order to monitor applications traversing a network, Protocol Discovery needs to be enabled.

The ability to classify traffic by protocol using NBAR and then applying QoS to the classified traffic is configured using the Modular QoS CLI.

The Modular QoS CLI is a CLI structure that allows users to create traffic policies and attach these policies to interfaces. A traffic policy contains a traffic class and one or more QoS features. A traffic class is used to classify traffic, while the QoS features in the traffic policy determine how to treat the classified traffic.

Modular QoS CLI configuration includes the following three steps:

-
- Step 1** Define a traffic class with the **class-map** command.
 - Step 2** Create a traffic policy by associating the traffic class with one or more QoS features (using the **policy-map** command).
 - Step 3** Attach the traffic policy to the interface with the **service-policy** command.
-

NBAR traffic classification occurs as part of the traffic class configuration.

For additional information on the Modular Quality of Service Command-Line Interface, see the “Configuring the Modular Quality of Service Command-Line Interface” section of the *Cisco IOS Quality of Service Solution Guide* on Cisco.com.

See the following sections for configuration tasks for the NBAR feature. Each task in the list is identified as either optional or required:

- [Enabling Protocol Discovery](#) (optional)
- [Configuring a Traffic Class](#) (required)
- [Configuring a Traffic Policy](#) (required)
- [Attaching a Traffic Policy to an Interface](#) (required)
- [Downloading PDLs](#) (optional)

Enabling Protocol Discovery

Use the **ip nbar protocol-discovery** command in order to enable monitoring of applications on a particular interface:

Command	Purpose
Router(config)# interface <i>interface-name</i>	Specifies the interface to configure.
Router(config-if)# ip nbar protocol-discovery	Enables monitoring by application on a particular interface.

Configuring a Traffic Class

Use the **class-map** configuration command to define a traffic class and the match criteria that will be used to classify network traffic when attached to an interface. When using NBAR to classify traffic, the **match protocol** command will be entered in class map configuration mode.

Command	Purpose
Router(config)# class-map [match-all match-any] <i>class-name</i>	Specifies the user-defined name of the traffic class. The match-all option specifies that all match criteria in the class map must be matched. The match-any option specifies that one or more match criteria must match.
Router(config-cmap)# match protocol <i>protocol-name</i>	Specifies a protocol supported by NBAR as a matching criterion.

Configuring a Traffic Policy

Use the **policy-map** configuration command to specify the QoS policies, such as Traffic Policing, Traffic Shaping, Low Latency Queueing, Class-Based Marking, Class-Based Weighted Fair Queueing or others, to apply to traffic classes defined by a traffic class. A traffic policy does not classify and forward traffic until being attached to an interface.

Command	Purpose
Router(config)# policy-map <i>policy-name</i>	User-specified policy map name.
Router(config-pmap)# class <i>class-name</i>	Specifies the name of a previously defined class map.
Router(config-pmap-c)#	Enter QoS policies in this configuration mode (policy map class).

For additional information on policy map options in the Modular Quality of Service Command-Line Interface, see the *Modular Quality of Service Command-Line Interface* document on Cisco.com.

Attaching a Traffic Policy to an Interface

A traffic policy is not active until it has been attached to an interface. Use the **service-policy** interface configuration command to attach a traffic policy to an interface and to specify the direction in which the policy should be applied (on either packets coming into the interface or packets leaving the interface).

Command	Purpose
Router(config)# interface <i>interface-name</i>	Specifies the interface to configure.
Router(config-if)# service-policy output <i>policy-map-name</i>	Attaches the previously configured traffic policy in the outbound direction of the interface. When this command is entered, all traffic leaving the interface will be classified and forwarded based on the traffic policy configuration.
Router(config-if)# service-policy input <i>policy-map-name</i>	Attaches the previously configured traffic policy in the input direction of the interface. When this command is entered, all traffic entering the interface will be classified and forwarded based on the traffic policy configuration.

Use the **no service-policy [input | output] policy-map-name** command to detach a policy map from an interface.

Downloading PDLMs

To extend or enhance the list of protocols recognized by NBAR through a Cisco-provided PDLM, use the **ip nbar pdlm** command after downloading the PDLM.

Command	Purpose
Router(config)# ip nbar pdlm <i>pdlm-name</i>	Specifies the PDLM used to extend or enhance the NBAR list of protocols.



Note

To view a list of currently available PDLMs or to download a PDLM, go to the following URL: <http://www.cisco.com/cgi-bin/tablebuild.pl/pdlm>

Verifying the Configuration

Use the **show policy-map** [**interface** [*interface-spec* [*input* | *output* [**class** *class-name*]]]] command to display the configuration of a policy map and its associated class maps. Forms of this command are listed in the table below.

Command	Purpose
Router# show class-map	Displays all traffic class information.
Router# show class-map <i>class-name</i>	Displays the traffic class information of the user-specified traffic class.
Router# show policy-map	Displays all configured traffic policies.
Router# show policy-map <i>policy-map-name</i>	Displays the user-specified traffic policies.
Router# show policy-map interface	Displays configurations and statistics of all input and output policies, which are attached to an interface.
Router# show policy-map <i>interface-spec</i>	Displays configuration and statistics of the input and output policies attached to a particular interface.
Router# show policy-map <i>interface-spec</i> [input]	Displays configuration and statistics of the input policy attached to an interface.
Router# show policy-map <i>interface-spec</i> [output]	Displays configuration and statistics of the output policy attached to an interface.
Router# show policy-map <i>interface-spec</i> [input output] class <i>class-name</i>	Displays configuration and statistics for the class name configured in the policy.

Troubleshooting Tips

- You must enable Cisco Express Forwarding (CEF) on the router prior to configuring the NBAR feature.
- Some error messages use the term “heuristic” to refer to a set of NBAR-supported protocols, and some error message documentation recommends actions to these heuristic protocols.

RTP is the only currently available heuristic protocol. If the error message or the error message documentation recommends an action to a heuristic protocol, take the recommended action on RTP.

Monitoring and Maintaining NBAR

NBAR can determine which protocols and applications are currently running on a network. NBAR includes the Protocol Discovery feature that provides an easy way of discovering application protocols operating on an interface so that appropriate QoS policies can be developed and applied. With Protocol Discovery, you can discover any protocol traffic supported by NBAR and obtain statistics associated with that protocol. To monitor and maintain the NBAR feature, use the following commands:

Command	Purpose
Router# <code>show ip nbar port-map [protocol-name]</code>	Displays the TCP/UDP port numbers used by NBAR to classify a given protocol.
Router# <code>show ip nbar protocol-discovery</code>	Displays the statistics for all interfaces on which Protocol Discovery is enabled.

Configuration Examples

This section provides the following configuration examples:

- [Configuring a Traffic Policy with NBAR](#)
- [Adding a PDLM](#)

Configuring a Traffic Policy with NBAR

In the following example, all SQL*Net traffic leaving fastethernet interface 0/1 is marked with the IP precedence value of 4. In the example, NBAR is used to identify SQL*Net traffic, while the treatment of SQL*Net traffic (in this case, it is forwarded with the IP precedence bit set as 4) is determined by the traffic policy configuration (the `set ip precedence 4` command in policy-map class configuration mode).

```
Router(config)# class-map sqlnettraffic
Router(config-cmap)# match protocol sqlnet

Router(config)# policy-map sqlsetipprec1
Router(config-pmap)# class sqlnettraffic
Router(config-pmap-c)# set ip precedence 4

Router(config)# interface fastethernet 0/1
Router(config-if)# service-policy output sqlsetipprec1
```

Adding a PDLM

In the following example, the FastTrack PDLM, which has already been downloaded to the Flash drive, is added as an NBAR-supported protocol:

```
Router(config)# ip nbar pdlm flash://fasttrack.pdlm
```

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at <http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm>.

- **ip nbar custom**
- **ip nbar pdlm**
- **ip nbar port-map**
- **ip nbar protocol-discovery**
- **match protocol**
- **match protocol citrix**
- **match protocol fasttrack**
- **match protocol gnutella**
- **match protocol http**
- **match protocol rtp**
- **show ip nbar pdlm**
- **show ip nbar port-map**
- **show ip nbar protocol-discovery**
- **show ip nbar version**

Glossary

Modular QoS CLI—Modular Quality of Service Command-Line Interface. A CLI for QoS features that makes configuring and implementing packet classification and QoS policies easier than with the existing CLI.

PDLM—Packet Description Language Module. A file containing Packet Description Language statements used to define the signature of one or more application protocols.

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.

Support classification—The classification of network traffic by information contained in the packet payload; that is, information found beyond the TCP or UDP port number.

Appendix

Sample Configuration

Below is a sample of how NBAR can be used.

E-Express Inc.'s network administrators wish to enforce the following policies on a 64-Kb WAN link:

- Reserve a minimum bandwidth of 32 Kb out of the 64 Kb available on the WAN link for all e-commerce traffic. This e-commerce traffic will be secure HTTP traffic or files being served from the `http://www.eexpress.com/transact/` directory through regular HTTP on the E-Express Inc. network.
- SuperNetwork Inc. is a very important partner to E-Express Inc. Reserve a minimum of 10 Kb for all traffic flowing from E-Express Inc. to SuperNetwork Inc.
- Limit to a maximum of 10 Kb all audio, video, and image web traffic.

Follow the steps below to configure the above policies:

Step 1 Classify all secure HTTP and HTTP traffic for the `/transact/` directory:

```
Router(config)# class-map match-all http_transact
Router(config-cmap)# match protocol http url "/transact/*"

Router(config)# class-map match-all http_secure
Router(config-cmap)# match protocol secure-http

Router(config)# class-map match-any ecommerce
Router(config-cmap)# match class-map http_transact
Router(config-cmap)# match class-map http_secure
```

Step 2 Classify all traffic to SuperNetwork Inc:

```
Router(config)# access-list 101 permit ip 10.0.0.1 0.0.0.0 10.0.0.3 0.0.0.0

Router(config)# class-map match-all super_network
Router(config-cmap)# match access-group 101
```

Step 3 Classify all audio, video, and image web traffic:

```
Router(config)# class-map match-any audio_video
Router(config-cmap)# match protocol http mime "audio/*"
Router(config-cmap)# match protocol http mime "video/*"

Router(config)# class-map match-any web_images
Router(config-cmap)# match protocol http url "*.gif"
Router(config-cmap)# match protocol http url "*.jpg|.jpeg"

Router(config)# class-map match-any av_im_web
Router(config-cmap)# match class-map audio_video
Router(config-cmap)# match class-map web_images
```

Step 4 Create the policies:

```
Router(config)# policy-map e-express
Router(config-pmap)# class ecommerce
Router(config-pmap-c)# bandwidth 32
Router(config-pmap-c)# class super_network
Router(config-pmap-c)# bandwidth 10
Router(config-pmap-c)# class av_im_web
Router(config-pmap-c)# police 10000 conform transmit exceed drop
```

Step 5 Attach the policy to the WAN link:

```
Router(config)# interface hssi1/0
Router(config-if)# service-policy output e-express
```



Network-Based Application Recognition Protocol Discovery Management Information Base

The existing Network-Based Application Recognition (NBAR) feature is used to identify protocols so traffic can be classified appropriately for Quality of Service purposes. NBAR also contains a Protocol Discovery feature that displays various statistics of any NBAR-supported protocol traffic traversing an interface for the user.

The NBAR Protocol Discovery Management Information Base (MIB) expands the capabilities of NBAR Protocol Discovery by providing the following new Protocol Discovery functionalities 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

Feature Specifications for Network-Based Application Recognition Protocol Discovery MIB

Feature History

Release	Modification
Release 12.2(15)T	This feature was introduced.

Supported Platforms

Cisco 1700, Cisco 2600, Cisco 3600, Cisco 3700, Cisco 7100, Cisco 7200, Cisco 7300, Cisco 7400, Cisco 7500 with VIP, Catalyst 6500 Family Switch with a FlexWAN card.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

This feature module describes the Network-Based Application Recognition Protocol Discovery MIB and contains the following sections:

- [Prerequisites for NBAR Protocol Discovery MIBs, page 134](#)
- [Restrictions for NBAR Protocol Discovery MIBs, page 134](#)
- [Tables Supported by NBAR Protocol Discovery MIBs, page 134](#)
- [How to Use the NBAR Protocol Discovery MIB, page 140](#)
- [Configuration Examples For NBAR Protocol Discovery MIBs, page 146](#)
- [Additional References, page 187](#)
- [Command Reference, page 188](#)

Prerequisites for NBAR Protocol Discovery MIBs

This feature is a MIB and therefore requires a method to read and configure MIBs in order to be used.

Restrictions for NBAR Protocol Discovery MIBs

- On some platforms, the NBAR Protocol Discovery MIB is not compatible with all possible Cisco IOS images on that platform. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn> for information regarding Protocol Discovery MIB support in your Cisco IOS release.
- When multiple thresholds are active, a negative impact on the router could occur. The number of thresholds that are configurable on a particular platform vary based on platform and threshold types, but users should ensure that unneeded thresholds are deactivated and that thresholds are configured in such a way that unwanted breaches do not occur.

Tables Supported by NBAR Protocol Discovery MIBs

Using the NBAR Protocol Discovery MIB, the following tables can be created:

- [cnpdSupportedProtocolsTable, page 135](#)
- [cnpdStatusTable, page 135](#)
- [cnpdAllStatsTable, page 136](#)
- [cnpdTopNConfigTable, page 137](#)
- [cnpdTopNStatsTable, page 138](#)
- [cnpdThresholdConfigTable, page 139](#)
- [cnpdThresholdHistoryTable, page 140](#)

cnpdSupportedProtocolsTable

The `cnpdSupportedProtocolsTable` lists all the protocols and applications that NBAR is capable of recognizing on the router. It is important to note that a user can add support for additional NBAR protocols and applications by downloading Packet Description Language Modules (PDLMs) or upgrading to an IOS that has added support for additional NBAR protocols or applications; therefore, this table will not be identical on all routers.

The `cnpdSupportedProtocolsTable` is composed of the following elements:

- `cnpdSupportedProtocolsIndex`—The `cnpdSupportedProtocolsIndex` represents the object identifier (OID) of an NBAR-supported protocol or application. This OID is a number and is used to select or identify a particular protocol throughout this MIB.
- `cnpdSupportedProtocolsName`—The `cnpdSupportedProtocolsName` is the name of the protocol associated with a specific OID in the `cnpdSupportedProtocolsTable` line of this table. The last `cnpdSupportedProtocolsName` is always “unknown” and is used to classify protocols and applications that are not recognized using NBAR Protocol Discovery.

cnpdStatusTable

The `cnpdStatusTable` provides the following functionality:

- The ability to enable or disable Protocol Discovery on an interface
- The ability to view if Protocol Discovery is enabled or disabled on an interface
- The ability to view when Protocol Discovery was last enabled on an interface.

The `cnpdStatusTable` contains the following elements:

- `IfIndex`—The `IfIndex` is a number that represents a specific interface.
- `cnpdStatusPdEnable`—The `cnpdStatusPdEnable` object is used to determine if NBAR is enabled or disabled on the interface. If `cnpdStatusPdEnable` is set to `true(1)`, then Protocol Discovery is enabled on that interface. If `cnpdStatusPdEnable` is set to `false(2)`, then Protocol Discovery is not enabled on that interface.

The `cnpdStatusPDEnable` object can be configured using the SNMP **setany** command. For an example of the `cnpdStatusPDEnable` object modification, see the [“Enabling and Disabling NBAR Protocol Discovery on an Interface”](#) section on page 141 of this document.

- `cnpdStatusLastUpdateTime`—The `cnpdStatusLastUpdateTime` displays the last time that Protocol Discovery was enabled on an interface. The number is given in timeticks which are converted to an actual time by the network management system.

cnpdAllStatsTable

The `cnpdAllStatsTable` is a table that contains the statistics collected on all NBAR-supported protocols on a per-interface basis.

The `cnpdAllStatsTable` contains the following elements:

- `IfIndex`—The `IfIndex` is a number that represents a specific interface.
- `ProtocolsIndex`—This number represents the protocol being monitored. To see which protocols are mapped to which numbers, see the [cnpdSupportedProtocolsTable](#) table.
- `cnpdAllStatsInPkts`—The `cnpdAllStatsInPkts` represents the number of packets that have been received by a specific interface. This value is given as a 32-bit variable; if the 32-bit variable is unable to handle the value (because SNMPv2 can support a 64-bit counter), the value used by the Network Management System (NMS) in place of `cnpdAllStatsInPkts` is `cnpdHCInPkts`.
- `cnpdAllStatsOutPkts`—The `cnpdAllStatsOutPkts` value represents the number of packets for traffic of a specific protocol or application that have left a specific interface. This value is given as a 32-bit variable; if the 32-bit variable is too small to handle the value (because SNMPv2 can support a 64-bit counter), the value used by the NMS in place of `cnpdAllStatsOutPkts` is `cnpdHCOuPkts`.
- `cnpdAllStatsInBytes`—The `cnpdAllStatsInBytes` represent the total number of bytes for traffic of a specific protocol or application that have entered a specific interface. This value is given as a 32-bit variable; if the 32-bit variable is too small to handle the value (because SNMPv2 can support a 64-bit counter), the value used by the NMS in place of `cnpdAllStatsInBytes` is `cnpdHCInBytes`.
- `cnpdAllStatsOutBytes`—The `cnpdAllStatsOutBytes` represent the total number of bytes for traffic of a specific protocol or application that have left a specific interface. This value is given as a 32-bit variable; if the 32-bit variable is unable to handle the value (because SNMPv2 can support a 64-bit counter), the value used by the NMS in place of `cnpdAllStatsOutBytes` is `cnpdHCOuBytes`.
- `cnpdAllStatsHCInPkts`—The `cnpdAllStatsHCInPkts` represent the total number of packets for traffic of a specific protocol or application that have entered a specific interface. This value is given as a 64-bit variable. If the `cnpdAllStatsInPkts` value could fit into the 32-bit counter, the `cnpdAllStatsHCInPkts` value will match the `cnpdInPkts` value. If the `cnpdAllStatsInPkts` value could not fit into the 64-bit counter, the NMS will use the `cnpdAllStatsHCInPkts` value in place of the `cnpdAllStatsInPkts` value.
- `cnpdAllStatsHCOuPkts`—The `cnpdAllStatsHCOuPkts` represent the total number of packets for traffic of a specific protocol or application that have left a specific interface. This value is given as a 64-bit variable. If the `cnpdAllStatsOutPkts` value could fit into the 32-bit counter, the `cnpdAllStatsHCOuPkts` value will match the `cnpdOutPkts` value. If the `cnpdAllStatsOutPkts` value could not fit into the 64-bit counter, the NMS will use the `cnpdAllStatsHCOuPkts` value in place of the `cnpdAllStatsOutPkts` value.
- `cnpdAllStatsHCInBytes`—The `cnpdAllStatsHCInBytes` represent the total number of bytes for traffic of a specific protocol or application that have entered a specific interface. This value is given in hexadecimal format. This value is given as a 64-bit variable. If the `cnpdAllStatsInBytes` value could fit into the 32-bit counter, the `cnpdAllStatsHCInBytes` value will match the `cnpdInBytes` value. If the `cnpdAllStatsInBytes` value could not fit into the 64-bit counter, the NMS will use the `cnpdAllStatsHCInBytes` value in place of the `cnpdAllStatsInBytes` value.
- `cnpdAllStatsHCOuBytes`—The `cnpdAllStatsHCOuBytes` represent the total number of bytes for traffic of a specific protocol or application that have left a specific interface. This value is given in hexadecimal format. This value is given as a 64-bit variable. If the `cnpdAllStatsOutBytes` value could fit into the 32-bit counter, the `cnpdAllStatsHCOuBytes` value will match the `cnpdAllStatsOutBytes` value. If the `cnpdAllStatsOutBytes` value could not fit into the 64-bit counter, the NMS will use the `cnpdAllStatsHCOuBytes` value in place of the `cnpdAllStatsOutBytes` value.

- `cnpdAllStatsInBitRate`—The `cnpdAllStatsInBitRate` represents the bit rate for traffic entering a specific interface.
- `cnpdAllStatsOutBitRate`—The `cnpdAllStatsOutBitRate` represents the bit rate for traffic leaving a specific interface.

`cnpdTopNConfigTable`

The `cnpdTopNConfigTable` is used to request a top-n list of protocols and their statistics.

The `cnpdTopNConfigTable` contains the following elements:

- `cnpdTopNConfigIndex`—The `cnpdTopNConfigIndex` is a number that represents a set of configuration parameters that will result in as a single row in the entire table.
- `cnpdTopNConfigIfIndex`—The `cnpdTopNConfigIfIndex` is a number that identifies a specific interface.
- `cnpdTopNConfigStatsSelect`—The `cnpdTopNConfigStatsSelect` determines the statistic used to calculate the order of precedence of the top-n protocols in the `cnpdTopNStatsTable`. The following statistics can be chosen:
 - `bitRateIn(1)`—Incoming bit rate
 - `bitRateOut(2)`—Outgoing bit rate
 - `bitRateSum(3)`—The sum of the incoming and outgoing bit rates
 - `byteCountIn(4)`—Incoming byte count
 - `byteCountOut(5)`—Outgoing byte count
 - `byteCountSum(6)`—Sum of incoming and outgoing byte counts
 - `packetCountIn(7)`—Incoming packet count
 - `packetCountOut(8)`—Outgoing packet count
 - `packetCountSum(9)`—Sum of incoming and outgoing packet counts
- `cnpdTopNConfigSampleTime`—For statistics based on bit rates only, the `cnpdTopNConfigSampleTime` determines the intervals during which the bit rate is sampled.
- `cnpdTopNConfigRequestedSize`—The `cnpdTopNConfigRequestedSize` selects the number of protocols or applications shown in the top-n table (in other words, it represent the *n* variable in the term “top-n”). In some cases, the `cnpdTopNConfigRequestedSize` may not show as many statistics as selected due to memory restrictions. The `cnpdTopNConfigGrantedSize` represents the actual number of protocols or applications displayed in the top-n table.
- `cnpdTopNConfigGrantedSize`—The `cnpdTopNConfigGrantedSize` is the actual number of protocols or applications that are shown in the top-n table. The number of protocols or applications in a top-n table does not always match the `cnpdTopNConfigRequestedSize` due to memory restrictions.
- `cnpdTopNConfigTime`—The `cnpdTopNConfigTime` represents the time in timeticks that a particular row entry was made active.

- `cnpdTopNConfigStatus`—The `cnpdTopNConfigStatus` creates and deletes rows in the `cnpdTopNConfigTable`.

It can have the following values:

- `Active`—A `TopNStats` entry has been generated for this row.
- `NotInService`—A `TopNStats` entry has not been generated for this row because `cnpdTopNConfigStatus` has been manually set to `NotInService`.
- `NotReady`—A `TopNStats` entry has not been generated for this row.
- `createAndGo`—create a table with default values for empty fields.
- `createAndWait`—create a table and set values for all fields.
- `Destroy`—destroy table.

All of these values can be set at an appropriate time, but `Active`, `NotInService`, and `NotReady` are the only states that this object can be converted to after the values are initially set.

Each row in the `cnpdTopNConfigStatus` contains a default value. These default values are used when a top-n table is created using `createAndGo`. If `createAndGo` is used to create a table from scratch, default values are used for each table row. The actual default values are defined in the MIB.

A user creating a top-n table should be aware that when a table is created using `createAndGo`, no default value can be assigned to an interface and the row status will therefore be `NotReady`.

cnpdTopNStatsTable

The `cnpdTopNStatsTable` contains an overall view of the `TopNStats`. In particular, this table takes the values of `TopNConfigTable` and the `cnpdTopNConfigGrantedSize` and produces an overall dynamic top-n table that monitors these objects.

This `cnpdTopNStatsTable` contains the following elements:

- `cnpdTopNConfigIndex`—The `cnpdTopNConfigIndex` represents the value of the index in the associated `cnpdTopNConfigTable`.
- `cnpdTopNStatsIndex`—The `cnpdTopNStatsIndex` indicates a position of a specific protocol in the top-n table.
- `cnpdTopNStatsIfIndex`—The `cnpdTopNStatsIfIndex` specifies the interface where the statistic of the top-n table is being monitored.
- `cnpdTopNStatsProtocolName`—The name of the protocol being measured.
- `cnpdTopNStatsRate`—measures the rate of the measured statistic. This value is given as a 32-bit variable. If this value cannot fit into the 32-bit variable, the `cnpdTopNHCRate` variable is used to provide the rate.
- `cnpdTopNHCRate`—measures the rate of the measured statistic. This value is given as a 64-bit variable. If this value can fit into the 32-bit variable, this value will match `cnpdTopNStatsRate`. If this value cannot fit into the 32-bit variable, the `cnpdTopNHCRate` is used to provide this rate.

cnpdThresholdConfigTable

The `cnpdThresholdConfigTable` is used to configure thresholds based on Protocol Discovery statistics.

The `cnpdThresholdConfigTable` contains the following elements:

- `cnpdThresholdConfigIndex`—Represents a threshold entry or notification if configured.
- `cnpdThresholdConfigIfIndex`—Represents the interface on which the protocol or application will be measured.
- `cnpdThresholdConfigInterval`—Represents the number of seconds that elapse before polling for the application or protocol.
- `cnpdThresholdConfigSampleType`—This value determines how statistics are sampled for the threshold.
 - If the `cnpdThresholdConfigSampleType` is set at `absoluteValue(1)`, the value at the end of the sampling interval `cnpdThresholdConfigInterval` will be compared with the `cnpdThresholdConfigRising` and `cnpdThresholdConfigFalling` thresholds.
 - If the `cnpdThresholdConfigSampleType` is set at `deltaValue(2)`, the difference between the samples at the beginning and at the end of the `cnpdThresholdConfigInterval` will be compared with the `cnpdThresholdConfigRising` and `cnpdThresholdConfigFalling` thresholds.
- `cnpdThresholdConfigProtocol`—The application or protocol that the thresholds for which the thresholds are being set.
- `cnpdThresholdConfigAny`—This setting determines if “any” protocol is being monitored for thresholds or if a particular protocol is being monitored for thresholds.
 - If `cnpdThresholdConfigAny` is set to true, “any” protocol is being monitored for thresholds. When any protocol is set, all protocols are monitored to see if they breach the thresholds and an SNMP trap is sent if any individual protocol breaches the threshold.
 - If `cnpdThresholdConfigAny` is set to false, a particular protocol is monitored to see if it crosses the configured thresholds. The `cnpdThresholdConfigProtocol` is used to set the particular protocol that is being monitored for threshold crossing.
- `cnpdThresholdConfigProtocol`—The `cnpdThresholdConfigProtocol` is used to set the particular protocol that is being monitored for threshold crossing.
- `cnpdThresholdConfigStatsSelect`—The `cnpdThresholdConfigStatsSelect` specifies the statistic that is being measured.
- `cnpdThresholdConfigStartup`—The `cnpdThresholdConfigStartup` controls if and when a notification should be generated the first time a statistic is measured. The following values may appear:
 - `rising(1)`—If `cnpdThresholdConfigStartup` is set to `rising`, no threshold breaches will be reported if the first measure of a statistic is above the rising threshold.
 - `falling(2)`—If `CnpdThresholdConfigStartup` is set to `falling`, no threshold breaches will be reported if the first measure of a statistic is below the falling threshold.
 - `risingOrFalling(3)`—The default setting. If the `cnpdThresholdConfigStartup` is set to `risingOrFalling`, no threshold breaches will be reported when the first measure of a statistic is reported.
- `cnpdThresholdConfigRising`—The `cnpdThresholdConfigRising` specifies the high value of the statistic being monitored that needs to be breached for a notification to be sent.
- `cnpdThresholdConfigFalling`—The `cnpdThresholdConfigFalling` specifies the low value of the statistic being monitored that needs to be breached for a notification to be sent.

- `cnpdThresholdConfigStatus`—Specifies if the row on the table is Active or NotReady.

cnpdThresholdHistoryTable

The `cnpdThresholdHistoryTable` keeps a history of all thresholds that have been breached. It is a two-dimensional table that tracks each particular breached threshold (which is numbered using `cnpdThresholdHistoryIndex`) and the entire `cnpdThresholdConfigTable`.

The `cnpdThresholdHistoryTable` contains the following elements:

- `cnpdThresholdHistoryIndex`—The `cnpdThresholdHistoryIndex` is a number that increments each time a threshold is breached.
- `cnpdThresholdHistoryConfigIndex`—represents a threshold entry or notification if configured. Same value as in the `cnpdThresholdConfigTable`.
- `cnpdThresholdHistoryValue`—The value of the sample at the time of the breach.
- `cnpdThresholdHistoryType`—indicates when the rising or falling threshold was breached in `timeticks`.
- `cnpdThresholdHistoryTime`—The time of the breach.
- `cnpdThresholdHistoryProtocol`—indicates the protocol that has breached the threshold.
- `cnpdThresholdHistoryStatsSelect`—indicates the statistic that was being monitored when this threshold was breached.

How to Use the NBAR Protocol Discovery MIB

The following sections provide information on configuring elements of the Protocol Discovery MIB and contains the following sections:

- [Querying the Supported Protocols Table, page 141](#)
- [Enabling and Disabling NBAR Protocol Discovery on an Interface, page 141](#)
- [Searching the AllStats Table, page 142](#)
- [Creating a Top-N Table, page 143](#)
- [Setting Protocol Thresholds, page 145](#)



Note

Throughout this document, the **setany** and **getmany** commands used with some MIB tools are used in the examples. In these cases, the **setany** command is equivalent to the SNMP `set` command and the **getmany** command is equivalent to the SNMP `getbulk` command.



Note

For detailed configurations and outputs, we strongly suggest viewing the “[Configuration Examples For NBAR Protocol Discovery MIBs](#)” section on page 146 of this document.

Querying the Supported Protocols Table

The Supported Protocols Table is used to see which protocols are supported by the NBAR Protocol Discovery MIB and to display which object identifier (OID) is assigned to each NBAR-supported application or protocol. The OID is a number that is used in the NBAR Protocol Discovery MIB to identify protocols and applications and to configure and read thresholds.

SUMMARY STEPS

1. `getmany -v2c IP-address public cnpdSupportedProtocols`

Examples

To view the supported protocols table, enter the following command in SNMP:

```
>$ getmany -v2c a.b.c.d public cnpdSupportedProtocols
```

(where a.b.c.d is the IP address of the router).

Enabling and Disabling NBAR Protocol Discovery on an Interface

The Status Enabled Table is used to either enable or disable NBAR Protocol Discovery on an interface or to view whether NBAR Protocol Discovery is enabled.

SUMMARY STEPS for Enabling NBAR Protocol Discovery on an Interface

1. `getmany -v2c IP-address public cnpdStatusTable` (optional)
2. `setany -v2c IP-address public cnpdStatusPdEnable.interface-number -i 1`

SUMMARY STEPS for Enabling NBAR Protocol Discovery on an Interface

1. `getmany -v2c IP-address public cnpdStatusTable` (optional)
2. `setany -v2c IP-address public cnpdStatusPdEnable.interface-number -i 2`

Examples

To view the Status Enabled Table, enter the following SNMP command:

```
>$ getmany -v2c a.b.c.d public cnpdStatusTable
```

(where a.b.c.d is the IP address of the router).

To enable Protocol Discovery on an interface, enter the following SNMP command:

```
>$ setany -v2c a.b.c.d public cnpdStatusPdEnable.14 -i 1
```

(where a.b.c.d is the IP address of the router, 14 is the number of the interface where Protocol Discovery is being enabled, and 1 sets the cnpdStatusPdEnable object to true(1) to enable Protocol Discovery).

To disable Protocol Discovery on an interface, enter the following SNMP command:

```
>$ setany -v2c a.b.c.d public cnpdStatusPdEnable.14 -i 2
```

(where a.b.c.d is the IP address of the router, 14 is the number of the interface where Protocol Discovery is being disabled, and 2 sets the cnpdStatusPdEnable object to false(2) to disable Protocol Discovery).

Searching the AllStats Table

The AllStats Table stores all of the statistics currently stored by NBAR Protocol Discovery. If many interfaces have enabled NBAR Protocol Discovery, the All Stats Table can get incredibly large.

To search the All Stats table, enter the following SNMP command:

```
>$ getmany -v2c a.b.c.d public cnpdAllStats
```

(where a.b.c.d is the IP address of the router).

SUMMARY STEPS

1. **getmany -v2c *IP-address* public cnpdAllStats**

Creating a Top-N Table

A top-n table in NBAR is a table that displays the most frequently classified NBAR-supported protocols for a specified statistic on a specified interface.

The top-n functionality in the NBAR Protocol Discovery MIB involves two tables. The first table is a configuration table in which each row represents a group of objects that will create a unique top-n report. The other table is a Statistics Results Table, which records the outputs of each row of the configuration table. The Statistics Results Table can be regenerated by the correct use of the rowStatus object in the configuration table.

Summary Steps

1. **setany -v2c** *IP-address* **public cnpdTopNConfigIfIndex.interface-number -i** *OID-number*
2. **getmany -v2c** *IP-address* **public cnpdTopNConfig**
3. **setany -v2c** *IP-address* **public cnpdTopNConfigStatsSelect.interface-number -i** *stats-value -i IfIndex*
4. **getmany -v2c** *IP-address* **public cnpdTopNConfig**
5. **setany -v2c** *IP-address* **public cnpdTopNConfigSampleTime.interface-number -g** *number-of-timeticks*
6. **getmany -v2c** *IP-address* **public cnpdTopNConfig**
7. **setany -v2c** *IP-address* **public cnpdTopNConfigRequestedSize.interface-number -g** *requested-size*
8. **getmany -v2c** *IP-address* **public cnpdTopNConfig**
9. **setany -v2c** *IP-address* **public cnpdTopNConfigStatus.interface-number -i** *config-status*
10. **getmany -v2c** *IP-address* **public cnpdTopNConfig**
11. **getmany -v2c** *IP-address* **public cnpdTopNStats**

Examples

To create a Top-n Table, enter the following SNMP commands:

```
>$ setany -v2c a.b.c.d public cnpdTopNConfigIfIndex.1 -i 13
```

(where a.b.c.d is the IP address of the router, 1 is the interface on the router, and 13 is the If-Index r of the protocol being monitored).

```
>$ getmany -v2c a.b.c.d public cnpdTopNConfig
```

(where a.b.c.d is the IP address of the router).

```
>$ setany -v2c a.b.c.d public cnpdTopNConfigStatsSelect.1 -i 7
```

(where a.b.c.d is the IP address of the router, 1 is the interface on the router, and 7 is the OID number of the protocol being monitored).

```
>$ getmany -v2c a.b.c.d public cnpdTopNConfig
```

(where a.b.c.d is the IP address of the router).

```
>$ setany -v2c a.b.c.d public cnpdTopNConfigSampleTime.1 -g 13
```

(where a.b.c.d is the IP address of the router, 1 is the interface on the router, and 13 is the number of timeticks between when samples are taken for the table).

```
>$ getmany -v2c a.b.c.d public cnpdTopNConfig  
(where a.b.c.d is the IP address of the router).
```

```
>$ setany -v2c a.b.c.d public cnpdTopNConfigRequestedSize.1 -g 5
```

(where a.b.c.d is the IP address of the router, 1 is the interface on the router, and 5 is the number of statistics to appear on the table [in this case, the top-5 statistics will be monitored by this top-n table assuming this object request is granted]).

```
>$ getmany -v2c a.b.c.d public cnpdTopNConfig
```

(where a.b.c.d is the IP address of the router).

```
>$ setany -v2c a.b.c.d public cnpdTopNConfigStatus.1 -i 4
```

(where a.b.c.d is the IP address of the router, 1 is the interface on the router, and 4 is the number that corresponds to a particular status).

```
>$ getmany -v2c a.b.c.d public cnpdTopNConfig
```

(where a.b.c.d is the IP address of the router).

```
>$ getmany -v2c a.b.c.d public cnpdTopNStats
```

(where a.b.c.d is the IP address of the router).

Setting Protocol Thresholds

The NBAR Protocol Discovery MIB can be used to set two types of thresholds—a threshold that sends a trap when thresholds for an individual protocol are crossed and a threshold that sends a trap when all NBAR-classifiable protocols or applications are crossed.

Setting a Threshold for a Particular Application or Protocol

A specific protocol threshold is a threshold that monitors the traffic of a specific protocol and the breaches of these thresholds are reported and stored when they occur.

The following items should be noted when configuring a specific protocol threshold:

- The `cnpdThresholdConfigProtocolAny` must be set to FALSE(2) and `cnpdThresholdConfigProtocol` must be set to a value that indicates the OID of the protocol being monitored. The OID for each protocol can be seen by querying `cnpdSupportedProtocolsTable`.
- Use `cnpdThresholdConfigIfIndex` to select the interface and `cnpdThresholdConfigRising` and `cnpdThresholdConfigFalling` to set the rising and falling thresholds.
- Set the `cnpdThresholdConfigInterval` to configure the frequency with which thresholds should be checked.
- Set `cnpdThresholdConfigStatus` to 4 (createAndGo) and this config member will become active.
- A hysteresis mechanism is used with thresholds using the NBAR Protocol Discovery MIB to avoid the reporting of multiple breaches.

Summary Steps

1. **setany -v2c IP-address public cnpdThresholdConfigProtocolAny.interface-number -i 2**
2. **setany -v2c IP-address public cnpdThresholdConfigProtocol.interface-number -g protocol-OID**
3. **setany -v2c IP-address public cnpdThresholdConfigStatsSelect.interface-number -i statistic**
4. **setany -v2c IP-address public cnpdThresholdConfigIfIndex.interface-number -i interface**
5. **setany -v2c IP-address public cnpdThresholdConfigRising.interface-number -g rising-threshold**
6. **setany -v2c IP-address public cnpdThresholdConfigFalling.interface-number -g falling-threshold**
7. **setany -v2c IP-address public cnpdThresholdConfigStatus.interface-number -i status**

Examples

```
setany -v2c a.b.c.d public cnpdThresholdConfigProtocolAny.2 -i 2
setany -v2c a.b.c.d public cnpdThresholdConfigProtocol.2 -g 5
setany -v2c a.b.c.d public cnpdThresholdConfigStatsSelect.2 -i 7
setany -v2c a.b.c.d public cnpdThresholdConfigIfIndex.2 -i 13
setany -v2c a.b.c.d public cnpdThresholdConfigRising.2 -g 2000
setany -v2c a.b.c.d public cnpdThresholdConfigFalling.2 -g 1000
setany -v2c a.b.c.d public cnpdThresholdConfigStatus.2 -i 4
```

Setting the Any Protocol Threshold

The Any Protocol Threshold setting is used to send SNMP traps and create history entries if ANY protocol crosses a rising or falling threshold. This threshold does not have a mechanism to report a breach only once in a given time period. For instance, if HTTP breaches a rising threshold, a trap is sent only for the first breach. If another protocol like DHCP breaches a rising threshold, a separate trap is sent for that breach.

The setup for any protocol is identical to a specific protocol except `cnpdThresholdConfigProtocolAny` does not have to be configured and it is unnecessary to specify a `cnpdThresholdConfigProtocol`.

Summary Steps

1. `setany -v2c IP-address public cnpdThresholdConfigStatsSelect.interface-number -i statistic`
2. `setany -v2c IP-address public cnpdThresholdConfigIfIndex.interface-number -i interface`
3. `setany -v2c IP-address public cnpdThresholdConfigRising.interface-number -g rising-threshold`
4. `setany -v2c IP-address public cnpdThresholdConfigFalling.interface-number -g falling-threshold`
5. `setany -v2c IP-address public cnpdThresholdConfigStatus.interface-number -i status`

Examples

```
setany -v2c a.b.c.d public cnpdThresholdConfigStatsSelect.1 -i 7
setany -v2c a.b.c.d public cnpdThresholdConfigIfIndex.1 -i 13
setany -v2c a.b.c.d public cnpdThresholdConfigRising.1 -g 30
setany -v2c a.b.c.d public cnpdThresholdConfigFalling.1 -g 20
setany -v2c a.b.c.d public cnpdThresholdConfigStatus.1 -i 4
```

Configuration Examples For NBAR Protocol Discovery MIBs

This section provides the following configuration examples:

- [Querying Supported Protocols Table, page 147](#)
- [Query Status Enabled Table, page 148](#)
- [Enabling and Disabling Protocol Discovery, page 150](#)
- [Disabling Protocol Discovery on an Interface, page 151](#)
- [Searching the All Stats Table, page 153](#)
- [Creating Top-N Tables Using the NBAR Protocol Discovery MIB, page 168](#)
- [Configuring Thresholds, page 171](#)
- [Threshold Options, page 172](#)



Note

Throughout this document, the **setany** and **getmany** commands used with some MIB tools are used in the examples. In these cases, the **setany** command is equivalent to the SNMP **set** command and the **getmany** command is equivalent to the SNMP **getbulk** command.

Querying Supported Protocols Table

The Querying Supported Protocols Table serves two purposes:

- Lists all the protocols and applications that can be classified by NBAR.
- Gives an OID per application or protocol that can be used in the NBAR Protocol Discovery MIB to identify protocols and applications and to configure and read thresholds.

```
>$ getmany -v2c a.b.c.d public cnpdSupportedProtocols
```

```
cnpdSupportedProtocolsName.1 = rtp  
cnpdSupportedProtocolsName.2 = fasttrack  
cnpdSupportedProtocolsName.3 = napster  
cnpdSupportedProtocolsName.4 = citrix  
cnpdSupportedProtocolsName.5 = http  
cnpdSupportedProtocolsName.6 = custom-10  
cnpdSupportedProtocolsName.7 = custom-09  
cnpdSupportedProtocolsName.8 = custom-08  
cnpdSupportedProtocolsName.9 = custom-07  
cnpdSupportedProtocolsName.10 = custom-06  
cnpdSupportedProtocolsName.11 = custom-05  
cnpdSupportedProtocolsName.12 = custom-04  
cnpdSupportedProtocolsName.13 = custom-03  
cnpdSupportedProtocolsName.14 = custom-02  
cnpdSupportedProtocolsName.15 = custom-01  
cnpdSupportedProtocolsName.16 = gnutella  
cnpdSupportedProtocolsName.17 = streamwork  
cnpdSupportedProtocolsName.18 = sunrpc  
cnpdSupportedProtocolsName.19 = netshow  
cnpdSupportedProtocolsName.20 = rcmd  
cnpdSupportedProtocolsName.21 = sqlnet  
cnpdSupportedProtocolsName.22 = vdolive  
cnpdSupportedProtocolsName.23 = realaudio  
cnpdSupportedProtocolsName.24 = exchange  
cnpdSupportedProtocolsName.25 = tftp  
cnpdSupportedProtocolsName.26 = novadigm  
cnpdSupportedProtocolsName.27 = printer  
cnpdSupportedProtocolsName.28 = xwindows  
cnpdSupportedProtocolsName.29 = secure-ftp  
cnpdSupportedProtocolsName.30 = secure-telnet  
cnpdSupportedProtocolsName.31 = telnet  
cnpdSupportedProtocolsName.32 = syslog  
cnpdSupportedProtocolsName.33 = ssh  
cnpdSupportedProtocolsName.34 = socks  
cnpdSupportedProtocolsName.35 = snmp  
cnpdSupportedProtocolsName.36 = smtp  
cnpdSupportedProtocolsName.37 = rsvp  
cnpdSupportedProtocolsName.38 = rip  
cnpdSupportedProtocolsName.39 = pptp  
cnpdSupportedProtocolsName.40 = secure-pop3  
cnpdSupportedProtocolsName.41 = pop3  
cnpdSupportedProtocolsName.42 = pcanypwhere
```

```

cnpdSupportedProtocolsName.43 = ntp
cnpdSupportedProtocolsName.44 = notes
cnpdSupportedProtocolsName.45 = secure-nntp
cnpdSupportedProtocolsName.46 = nntp
cnpdSupportedProtocolsName.47 = nfs
cnpdSupportedProtocolsName.48 = netbios
cnpdSupportedProtocolsName.49 = sqlserver
cnpdSupportedProtocolsName.50 = secure-ldap
cnpdSupportedProtocolsName.51 = ldap
cnpdSupportedProtocolsName.52 = l2tp
cnpdSupportedProtocolsName.53 = kerberos
cnpdSupportedProtocolsName.54 = secure-irc
cnpdSupportedProtocolsName.55 = irc
cnpdSupportedProtocolsName.56 = secure-imap
cnpdSupportedProtocolsName.57 = imap
cnpdSupportedProtocolsName.58 = secure-http
cnpdSupportedProtocolsName.59 = gopher
cnpdSupportedProtocolsName.60 = finger
cnpdSupportedProtocolsName.61 = dns
cnpdSupportedProtocolsName.62 = dhcp
cnpdSupportedProtocolsName.63 = cuseeme
cnpdSupportedProtocolsName.64 = bgp
cnpdSupportedProtocolsName.65 = ipsec
cnpdSupportedProtocolsName.66 = ipinip
cnpdSupportedProtocolsName.67 = eigrp
cnpdSupportedProtocolsName.68 = icmp
cnpdSupportedProtocolsName.69 = gre
cnpdSupportedProtocolsName.70 = egp
cnpdSupportedProtocolsName.71 = ftp
cnpdSupportedProtocolsName.72 = unknown

```

Query Status Enabled Table

The Query Status Enabled Table has two purposes:

- Each row represents an interface on the router and whether NBAR Protocol Discovery is enabled or disabled on that interface.
- Each row can be configured to enable or disable NBAR Protocol Discovery using SNMP set.

```
>$ getmany -v2c a.b.c.d public cnpdStatusTable
```

```

cnpdStatusPdEnable.1 = false(2)
cnpdStatusPdEnable.2 = false(2)
cnpdStatusPdEnable.3 = false(2)
cnpdStatusPdEnable.4 = false(2)
cnpdStatusPdEnable.5 = false(2)
cnpdStatusPdEnable.6 = false(2)
cnpdStatusPdEnable.7 = false(2)
cnpdStatusPdEnable.8 = false(2)
cnpdStatusPdEnable.9 = false(2)
cnpdStatusPdEnable.10 = false(2)
cnpdStatusPdEnable.11 = false(2)

```

```
cnpdStatusPdEnable.12 = false(2)
cnpdStatusPdEnable.13 = true(1)
cnpdStatusPdEnable.14 = false(2)
cnpdStatusPdEnable.15 = false(2)
cnpdStatusPdEnable.16 = false(2)
cnpdStatusPdEnable.17 = false(2)
cnpdStatusPdEnable.18 = false(2)
cnpdStatusPdEnable.19 = false(2)
cnpdStatusPdEnable.20 = false(2)
cnpdStatusPdEnable.21 = false(2)
cnpdStatusPdEnable.22 = false(2)
cnpdStatusPdEnable.23 = false(2)
cnpdStatusPdEnable.24 = false(2)
cnpdStatusPdEnable.25 = false(2)
cnpdStatusPdEnable.26 = false(2)
cnpdStatusPdEnable.27 = false(2)
cnpdStatusPdEnable.28 = false(2)
cnpdStatusPdEnable.29 = false(2)
cnpdStatusPdEnable.30 = false(2)
cnpdStatusPdEnable.31 = false(2)
cnpdStatusPdEnable.32 = false(2)
cnpdStatusPdEnable.33 = false(2)
cnpdStatusPdEnable.34 = false(2)
cnpdStatusPdEnable.35 = false(2)
cnpdStatusLastUpdateTime.1 = 0
cnpdStatusLastUpdateTime.2 = 0
cnpdStatusLastUpdateTime.3 = 0
cnpdStatusLastUpdateTime.4 = 0
cnpdStatusLastUpdateTime.5 = 0
cnpdStatusLastUpdateTime.6 = 0
cnpdStatusLastUpdateTime.7 = 0
cnpdStatusLastUpdateTime.8 = 0
cnpdStatusLastUpdateTime.9 = 0
cnpdStatusLastUpdateTime.10 = 0
cnpdStatusLastUpdateTime.11 = 0
cnpdStatusLastUpdateTime.12 = 0
cnpdStatusLastUpdateTime.13 = 1111
cnpdStatusLastUpdateTime.14 = 0
cnpdStatusLastUpdateTime.15 = 0
cnpdStatusLastUpdateTime.16 = 0
cnpdStatusLastUpdateTime.17 = 0
cnpdStatusLastUpdateTime.18 = 0
cnpdStatusLastUpdateTime.19 = 0
cnpdStatusLastUpdateTime.20 = 0
cnpdStatusLastUpdateTime.21 = 0
cnpdStatusLastUpdateTime.22 = 0
cnpdStatusLastUpdateTime.23 = 0
cnpdStatusLastUpdateTime.24 = 0
cnpdStatusLastUpdateTime.25 = 0
cnpdStatusLastUpdateTime.26 = 0
cnpdStatusLastUpdateTime.27 = 0
cnpdStatusLastUpdateTime.28 = 0
cnpdStatusLastUpdateTime.29 = 0
```

```

cnpdStatusLastUpdateTime.30 = 0
cnpdStatusLastUpdateTime.31 = 0
cnpdStatusLastUpdateTime.32 = 0
cnpdStatusLastUpdateTime.33 = 0
cnpdStatusLastUpdateTime.34 = 0
cnpdStatusLastUpdateTime.35 = 0

```

Enabling and Disabling Protocol Discovery

The following examples show how to enable Protocol Discovery using the NBAR Protocol Discovery MIB using the SNMP set.

```
>$ setany -v2c a.b.c.d public cnpdStatusPdEnable.14 -i 1
```

```
cnpdStatusPdEnable.14 = true(1)
```

```
>$ getmany -v2c a.b.c.d public cnpdStatusTable
```

```

cnpdStatusPdEnable.1 = false(2)
cnpdStatusPdEnable.2 = false(2)
cnpdStatusPdEnable.3 = false(2)
cnpdStatusPdEnable.4 = false(2)
cnpdStatusPdEnable.5 = false(2)
cnpdStatusPdEnable.6 = false(2)
cnpdStatusPdEnable.7 = false(2)
cnpdStatusPdEnable.8 = false(2)
cnpdStatusPdEnable.9 = false(2)
cnpdStatusPdEnable.10 = false(2)
cnpdStatusPdEnable.11 = false(2)
cnpdStatusPdEnable.12 = false(2)
cnpdStatusPdEnable.13 = true(1)
cnpdStatusPdEnable.14 = true(1)
cnpdStatusPdEnable.15 = false(2)
cnpdStatusPdEnable.16 = false(2)
cnpdStatusPdEnable.17 = false(2)
cnpdStatusPdEnable.18 = false(2)
cnpdStatusPdEnable.19 = false(2)
cnpdStatusPdEnable.20 = false(2)
cnpdStatusPdEnable.21 = false(2)
cnpdStatusPdEnable.22 = false(2)
cnpdStatusPdEnable.23 = false(2)
cnpdStatusPdEnable.24 = false(2)
cnpdStatusPdEnable.25 = false(2)
cnpdStatusPdEnable.26 = false(2)
cnpdStatusPdEnable.27 = false(2)
cnpdStatusPdEnable.28 = false(2)
cnpdStatusPdEnable.29 = false(2)
cnpdStatusPdEnable.30 = false(2)
cnpdStatusPdEnable.31 = false(2)
cnpdStatusPdEnable.32 = false(2)

```

```
cnpdStatusPdEnable.33 = false(2)
cnpdStatusPdEnable.34 = false(2)
cnpdStatusPdEnable.35 = false(2)
cnpdStatusLastUpdateTime.1 = 0
cnpdStatusLastUpdateTime.2 = 0
cnpdStatusLastUpdateTime.3 = 0
cnpdStatusLastUpdateTime.4 = 0
cnpdStatusLastUpdateTime.5 = 0
cnpdStatusLastUpdateTime.6 = 0
cnpdStatusLastUpdateTime.7 = 0
cnpdStatusLastUpdateTime.8 = 0
cnpdStatusLastUpdateTime.9 = 0
cnpdStatusLastUpdateTime.10 = 0
cnpdStatusLastUpdateTime.11 = 0
cnpdStatusLastUpdateTime.12 = 0
cnpdStatusLastUpdateTime.13 = 1111
cnpdStatusLastUpdateTime.14 = 44042577
cnpdStatusLastUpdateTime.15 = 0
cnpdStatusLastUpdateTime.16 = 0
cnpdStatusLastUpdateTime.17 = 0
cnpdStatusLastUpdateTime.18 = 0
cnpdStatusLastUpdateTime.19 = 0
cnpdStatusLastUpdateTime.20 = 0
cnpdStatusLastUpdateTime.21 = 0
cnpdStatusLastUpdateTime.22 = 0
cnpdStatusLastUpdateTime.23 = 0
cnpdStatusLastUpdateTime.24 = 0
cnpdStatusLastUpdateTime.25 = 0
cnpdStatusLastUpdateTime.26 = 0
cnpdStatusLastUpdateTime.27 = 0
cnpdStatusLastUpdateTime.28 = 0
cnpdStatusLastUpdateTime.29 = 0
cnpdStatusLastUpdateTime.30 = 0
cnpdStatusLastUpdateTime.31 = 0
cnpdStatusLastUpdateTime.32 = 0
cnpdStatusLastUpdateTime.33 = 0
cnpdStatusLastUpdateTime.34 = 0
cnpdStatusLastUpdateTime.35 = 0
```

Disabling Protocol Discovery on an Interface

The following example shows how to disable Protocol Discovery on an interface using SNMP set.

```
>$ setany -v2c a.b.c.d public cnpdStatusPdEnable.14 -i 2
cnpdStatusPdEnable.14 = false(2)
>$ getmany -v2c a.b.c.d public cnpdStatusTable
cnpdStatusPdEnable.1 = false(2)
cnpdStatusPdEnable.2 = false(2)
cnpdStatusPdEnable.3 = false(2)
cnpdStatusPdEnable.4 = false(2)
```

```
cnpdStatusPdEnable.5 = false(2)
cnpdStatusPdEnable.6 = false(2)
cnpdStatusPdEnable.7 = false(2)
cnpdStatusPdEnable.8 = false(2)
cnpdStatusPdEnable.9 = false(2)
cnpdStatusPdEnable.10 = false(2)
cnpdStatusPdEnable.11 = false(2)
cnpdStatusPdEnable.12 = false(2)
cnpdStatusPdEnable.13 = true(1)
cnpdStatusPdEnable.14 = false(2)
cnpdStatusPdEnable.15 = false(2)
cnpdStatusPdEnable.16 = false(2)
cnpdStatusPdEnable.17 = false(2)
cnpdStatusPdEnable.18 = false(2)
cnpdStatusPdEnable.19 = false(2)
cnpdStatusPdEnable.20 = false(2)
cnpdStatusPdEnable.21 = false(2)
cnpdStatusPdEnable.22 = false(2)
cnpdStatusPdEnable.23 = false(2)
cnpdStatusPdEnable.24 = false(2)
cnpdStatusPdEnable.25 = false(2)
cnpdStatusPdEnable.26 = false(2)
cnpdStatusPdEnable.27 = false(2)
cnpdStatusPdEnable.28 = false(2)
cnpdStatusPdEnable.29 = false(2)
cnpdStatusPdEnable.30 = false(2)
cnpdStatusPdEnable.31 = false(2)
cnpdStatusPdEnable.32 = false(2)
cnpdStatusPdEnable.33 = false(2)
cnpdStatusPdEnable.34 = false(2)
cnpdStatusPdEnable.35 = false(2)
cnpdStatusLastUpdateTime.1 = 0
cnpdStatusLastUpdateTime.2 = 0
cnpdStatusLastUpdateTime.3 = 0
cnpdStatusLastUpdateTime.4 = 0
cnpdStatusLastUpdateTime.5 = 0
cnpdStatusLastUpdateTime.6 = 0
cnpdStatusLastUpdateTime.7 = 0
cnpdStatusLastUpdateTime.8 = 0
cnpdStatusLastUpdateTime.9 = 0
cnpdStatusLastUpdateTime.10 = 0
cnpdStatusLastUpdateTime.11 = 0
cnpdStatusLastUpdateTime.12 = 0
cnpdStatusLastUpdateTime.13 = 1111
cnpdStatusLastUpdateTime.14 = 0
cnpdStatusLastUpdateTime.15 = 0
cnpdStatusLastUpdateTime.16 = 0
cnpdStatusLastUpdateTime.17 = 0
cnpdStatusLastUpdateTime.18 = 0
cnpdStatusLastUpdateTime.19 = 0
cnpdStatusLastUpdateTime.20 = 0
cnpdStatusLastUpdateTime.21 = 0
cnpdStatusLastUpdateTime.22 = 0
```

```
cnpdStatusLastUpdateTime.23 = 0
cnpdStatusLastUpdateTime.24 = 0
cnpdStatusLastUpdateTime.25 = 0
cnpdStatusLastUpdateTime.26 = 0
cnpdStatusLastUpdateTime.27 = 0
cnpdStatusLastUpdateTime.28 = 0
cnpdStatusLastUpdateTime.29 = 0
cnpdStatusLastUpdateTime.30 = 0
cnpdStatusLastUpdateTime.31 = 0
cnpdStatusLastUpdateTime.32 = 0
cnpdStatusLastUpdateTime.33 = 0
cnpdStatusLastUpdateTime.34 = 0
cnpdStatusLastUpdateTime.35 = 0
```

Searching the All Stats Table

The All Stats Table stores all of the statistics currently stored by NBAR Protocol Discovery. If many interfaces have enabled NBAR Protocol Discovery, the All Stats Table can get incredibly large. In the following example, NBAR Protocol Discovery is enabled on one interface only.

```
>$ getmany -v2c a.b.c.d public cnpdAllStats
```

```
cnpdAllStatsProtocolName.13.1 = rtp
cnpdAllStatsProtocolName.13.2 = fasttrack
cnpdAllStatsProtocolName.13.3 = napster
cnpdAllStatsProtocolName.13.4 = citrix
cnpdAllStatsProtocolName.13.5 = http
cnpdAllStatsProtocolName.13.6 = custom-10
cnpdAllStatsProtocolName.13.7 = custom-09
cnpdAllStatsProtocolName.13.8 = custom-08
cnpdAllStatsProtocolName.13.9 = custom-07
cnpdAllStatsProtocolName.13.10 = custom-06
cnpdAllStatsProtocolName.13.11 = custom-05
cnpdAllStatsProtocolName.13.12 = custom-04
cnpdAllStatsProtocolName.13.13 = custom-03
cnpdAllStatsProtocolName.13.14 = custom-02
cnpdAllStatsProtocolName.13.15 = custom-01
cnpdAllStatsProtocolName.13.16 = gnutella
cnpdAllStatsProtocolName.13.17 = streamwork
cnpdAllStatsProtocolName.13.18 = sunrpc
cnpdAllStatsProtocolName.13.19 = netshow
cnpdAllStatsProtocolName.13.20 = rcmd
cnpdAllStatsProtocolName.13.21 = sqlnet
cnpdAllStatsProtocolName.13.22 = vdolive
cnpdAllStatsProtocolName.13.23 = realaudio
cnpdAllStatsProtocolName.13.24 = exchange
cnpdAllStatsProtocolName.13.25 = tftp
cnpdAllStatsProtocolName.13.26 = novadigm
cnpdAllStatsProtocolName.13.27 = printer
cnpdAllStatsProtocolName.13.28 = xwindows
cnpdAllStatsProtocolName.13.29 = secure-ftp
```

```
cnpdAllStatsProtocolName.13.30 = secure-telnet
cnpdAllStatsProtocolName.13.31 = telnet
cnpdAllStatsProtocolName.13.32 = syslog
cnpdAllStatsProtocolName.13.33 = ssh
cnpdAllStatsProtocolName.13.34 = socks
cnpdAllStatsProtocolName.13.35 = snmp
cnpdAllStatsProtocolName.13.36 = smtp
cnpdAllStatsProtocolName.13.37 = rsvp
cnpdAllStatsProtocolName.13.38 = rip
cnpdAllStatsProtocolName.13.39 = pptp
cnpdAllStatsProtocolName.13.40 = secure-pop3
cnpdAllStatsProtocolName.13.41 = pop3
cnpdAllStatsProtocolName.13.42 = pcan anywhere
cnpdAllStatsProtocolName.13.43 = ntp
cnpdAllStatsProtocolName.13.44 = notes
cnpdAllStatsProtocolName.13.45 = secure-nntp
cnpdAllStatsProtocolName.13.46 = nntp
cnpdAllStatsProtocolName.13.47 = nfs
cnpdAllStatsProtocolName.13.48 = netbios
cnpdAllStatsProtocolName.13.49 = sqlserver
cnpdAllStatsProtocolName.13.50 = secure-ldap
cnpdAllStatsProtocolName.13.51 = ldap
cnpdAllStatsProtocolName.13.52 = l2tp
cnpdAllStatsProtocolName.13.53 = kerberos
cnpdAllStatsProtocolName.13.54 = secure-irc
cnpdAllStatsProtocolName.13.55 = irc
cnpdAllStatsProtocolName.13.56 = secure-imap
cnpdAllStatsProtocolName.13.57 = imap
cnpdAllStatsProtocolName.13.58 = secure-http
cnpdAllStatsProtocolName.13.59 = gopher
cnpdAllStatsProtocolName.13.60 = finger
cnpdAllStatsProtocolName.13.61 = dns
cnpdAllStatsProtocolName.13.62 = dhcp
cnpdAllStatsProtocolName.13.63 = cuseeme
cnpdAllStatsProtocolName.13.64 = bgp
cnpdAllStatsProtocolName.13.65 = ipsec
cnpdAllStatsProtocolName.13.66 = ipinip
cnpdAllStatsProtocolName.13.67 = eigrp
cnpdAllStatsProtocolName.13.68 = icmp
cnpdAllStatsProtocolName.13.69 = gre
cnpdAllStatsProtocolName.13.70 = egp
cnpdAllStatsProtocolName.13.71 = ftp
cnpdAllStatsProtocolName.13.72 = unknown
cnpdAllStatsInPkts.13.1 = 0
cnpdAllStatsInPkts.13.2 = 0
cnpdAllStatsInPkts.13.3 = 108392
cnpdAllStatsInPkts.13.4 = 0
cnpdAllStatsInPkts.13.5 = 3681501
cnpdAllStatsInPkts.13.6 = 0
cnpdAllStatsInPkts.13.7 = 0
cnpdAllStatsInPkts.13.8 = 0
cnpdAllStatsInPkts.13.9 = 0
cnpdAllStatsInPkts.13.10 = 0
```



```
cnpdAllStatsInPkts.13.11 = 0
cnpdAllStatsInPkts.13.12 = 0
cnpdAllStatsInPkts.13.13 = 0
cnpdAllStatsInPkts.13.14 = 0
cnpdAllStatsInPkts.13.15 = 0
cnpdAllStatsInPkts.13.16 = 0
cnpdAllStatsInPkts.13.17 = 0
cnpdAllStatsInPkts.13.18 = 0
cnpdAllStatsInPkts.13.19 = 0
cnpdAllStatsInPkts.13.20 = 0
cnpdAllStatsInPkts.13.21 = 0
cnpdAllStatsInPkts.13.22 = 0
cnpdAllStatsInPkts.13.23 = 1348424
cnpdAllStatsInPkts.13.24 = 0
cnpdAllStatsInPkts.13.25 = 0
cnpdAllStatsInPkts.13.26 = 0
cnpdAllStatsInPkts.13.27 = 0
cnpdAllStatsInPkts.13.28 = 0
cnpdAllStatsInPkts.13.29 = 0
cnpdAllStatsInPkts.13.30 = 0
cnpdAllStatsInPkts.13.31 = 0
cnpdAllStatsInPkts.13.32 = 0
cnpdAllStatsInPkts.13.33 = 0
cnpdAllStatsInPkts.13.34 = 0
cnpdAllStatsInPkts.13.35 = 0
cnpdAllStatsInPkts.13.36 = 106086
cnpdAllStatsInPkts.13.37 = 0
cnpdAllStatsInPkts.13.38 = 0
cnpdAllStatsInPkts.13.39 = 0
cnpdAllStatsInPkts.13.40 = 0
cnpdAllStatsInPkts.13.41 = 17941
cnpdAllStatsInPkts.13.42 = 0
cnpdAllStatsInPkts.13.43 = 0
cnpdAllStatsInPkts.13.44 = 0
cnpdAllStatsInPkts.13.45 = 0
cnpdAllStatsInPkts.13.46 = 562337
cnpdAllStatsInPkts.13.47 = 0
cnpdAllStatsInPkts.13.48 = 81449
cnpdAllStatsInPkts.13.49 = 0
cnpdAllStatsInPkts.13.50 = 0
cnpdAllStatsInPkts.13.51 = 366922
cnpdAllStatsInPkts.13.52 = 0
cnpdAllStatsInPkts.13.53 = 0
cnpdAllStatsInPkts.13.54 = 0
cnpdAllStatsInPkts.13.55 = 0
cnpdAllStatsInPkts.13.56 = 0
cnpdAllStatsInPkts.13.57 = 0
cnpdAllStatsInPkts.13.58 = 0
cnpdAllStatsInPkts.13.59 = 17318
cnpdAllStatsInPkts.13.60 = 0
cnpdAllStatsInPkts.13.61 = 36182
cnpdAllStatsInPkts.13.62 = 12134
cnpdAllStatsInPkts.13.63 = 0
```

```
cnpdAllStatsInPkts.13.64 = 0
cnpdAllStatsInPkts.13.65 = 0
cnpdAllStatsInPkts.13.66 = 0
cnpdAllStatsInPkts.13.67 = 0
cnpdAllStatsInPkts.13.68 = 9046
cnpdAllStatsInPkts.13.69 = 0
cnpdAllStatsInPkts.13.70 = 0
cnpdAllStatsInPkts.13.71 = 311978
cnpdAllStatsInPkts.13.72 = 27150
cnpdAllStatsOutPkts.13.1 = 0
cnpdAllStatsOutPkts.13.2 = 0
cnpdAllStatsOutPkts.13.3 = 0
cnpdAllStatsOutPkts.13.4 = 0
cnpdAllStatsOutPkts.13.5 = 0
cnpdAllStatsOutPkts.13.6 = 0
cnpdAllStatsOutPkts.13.7 = 0
cnpdAllStatsOutPkts.13.8 = 0
cnpdAllStatsOutPkts.13.9 = 0
cnpdAllStatsOutPkts.13.10 = 0
cnpdAllStatsOutPkts.13.11 = 0
cnpdAllStatsOutPkts.13.12 = 0
cnpdAllStatsOutPkts.13.13 = 0
cnpdAllStatsOutPkts.13.14 = 0
cnpdAllStatsOutPkts.13.15 = 0
cnpdAllStatsOutPkts.13.16 = 0
cnpdAllStatsOutPkts.13.17 = 0
cnpdAllStatsOutPkts.13.18 = 0
cnpdAllStatsOutPkts.13.19 = 0
cnpdAllStatsOutPkts.13.20 = 0
cnpdAllStatsOutPkts.13.21 = 0
cnpdAllStatsOutPkts.13.22 = 0
cnpdAllStatsOutPkts.13.23 = 0
cnpdAllStatsOutPkts.13.24 = 0
cnpdAllStatsOutPkts.13.25 = 0
cnpdAllStatsOutPkts.13.26 = 0
cnpdAllStatsOutPkts.13.27 = 0
cnpdAllStatsOutPkts.13.28 = 0
cnpdAllStatsOutPkts.13.29 = 0
cnpdAllStatsOutPkts.13.30 = 0
cnpdAllStatsOutPkts.13.31 = 0
cnpdAllStatsOutPkts.13.32 = 0
cnpdAllStatsOutPkts.13.33 = 0
cnpdAllStatsOutPkts.13.34 = 0
cnpdAllStatsOutPkts.13.35 = 0
cnpdAllStatsOutPkts.13.36 = 0
cnpdAllStatsOutPkts.13.37 = 0
cnpdAllStatsOutPkts.13.38 = 0
cnpdAllStatsOutPkts.13.39 = 0
cnpdAllStatsOutPkts.13.40 = 0
cnpdAllStatsOutPkts.13.41 = 0
cnpdAllStatsOutPkts.13.42 = 0
cnpdAllStatsOutPkts.13.43 = 0
cnpdAllStatsOutPkts.13.44 = 0
```

```
cnpdAllStatsOutPkts.13.45 = 0
cnpdAllStatsOutPkts.13.46 = 0
cnpdAllStatsOutPkts.13.47 = 0
cnpdAllStatsOutPkts.13.48 = 0
cnpdAllStatsOutPkts.13.49 = 0
cnpdAllStatsOutPkts.13.50 = 0
cnpdAllStatsOutPkts.13.51 = 0
cnpdAllStatsOutPkts.13.52 = 0
cnpdAllStatsOutPkts.13.53 = 0
cnpdAllStatsOutPkts.13.54 = 0
cnpdAllStatsOutPkts.13.55 = 0
cnpdAllStatsOutPkts.13.56 = 0
cnpdAllStatsOutPkts.13.57 = 0
cnpdAllStatsOutPkts.13.58 = 0
cnpdAllStatsOutPkts.13.59 = 0
cnpdAllStatsOutPkts.13.60 = 0
cnpdAllStatsOutPkts.13.61 = 0
cnpdAllStatsOutPkts.13.62 = 0
cnpdAllStatsOutPkts.13.63 = 0
cnpdAllStatsOutPkts.13.64 = 0
cnpdAllStatsOutPkts.13.65 = 0
cnpdAllStatsOutPkts.13.66 = 0
cnpdAllStatsOutPkts.13.67 = 0
cnpdAllStatsOutPkts.13.68 = 0
cnpdAllStatsOutPkts.13.69 = 0
cnpdAllStatsOutPkts.13.70 = 0
cnpdAllStatsOutPkts.13.71 = 0
cnpdAllStatsOutPkts.13.72 = 0
cnpdAllStatsInBytes.13.1 = 0
cnpdAllStatsInBytes.13.2 = 0
cnpdAllStatsInBytes.13.3 = 20157945
cnpdAllStatsInBytes.13.4 = 0
cnpdAllStatsInBytes.13.5 = 2255639168
cnpdAllStatsInBytes.13.6 = 0
cnpdAllStatsInBytes.13.7 = 0
cnpdAllStatsInBytes.13.8 = 0
cnpdAllStatsInBytes.13.9 = 0
cnpdAllStatsInBytes.13.10 = 0
cnpdAllStatsInBytes.13.11 = 0
cnpdAllStatsInBytes.13.12 = 0
cnpdAllStatsInBytes.13.13 = 0
cnpdAllStatsInBytes.13.14 = 0
cnpdAllStatsInBytes.13.15 = 0
cnpdAllStatsInBytes.13.16 = 0
cnpdAllStatsInBytes.13.17 = 0
cnpdAllStatsInBytes.13.18 = 0
cnpdAllStatsInBytes.13.19 = 0
cnpdAllStatsInBytes.13.20 = 0
cnpdAllStatsInBytes.13.21 = 0
cnpdAllStatsInBytes.13.22 = 0
cnpdAllStatsInBytes.13.23 = 464526676
cnpdAllStatsInBytes.13.24 = 0
cnpdAllStatsInBytes.13.25 = 0
```

```
cnpdAllStatsInBytes.13.26 = 0
cnpdAllStatsInBytes.13.27 = 0
cnpdAllStatsInBytes.13.28 = 0
cnpdAllStatsInBytes.13.29 = 0
cnpdAllStatsInBytes.13.30 = 0
cnpdAllStatsInBytes.13.31 = 0
cnpdAllStatsInBytes.13.32 = 0
cnpdAllStatsInBytes.13.33 = 0
cnpdAllStatsInBytes.13.34 = 0
cnpdAllStatsInBytes.13.35 = 0
cnpdAllStatsInBytes.13.36 = 10280188
cnpdAllStatsInBytes.13.37 = 0
cnpdAllStatsInBytes.13.38 = 0
cnpdAllStatsInBytes.13.39 = 0
cnpdAllStatsInBytes.13.40 = 0
cnpdAllStatsInBytes.13.41 = 1298760
cnpdAllStatsInBytes.13.42 = 0
cnpdAllStatsInBytes.13.43 = 0
cnpdAllStatsInBytes.13.44 = 0
cnpdAllStatsInBytes.13.45 = 0
cnpdAllStatsInBytes.13.46 = 149193275
cnpdAllStatsInBytes.13.47 = 0
cnpdAllStatsInBytes.13.48 = 14181177
cnpdAllStatsInBytes.13.49 = 0
cnpdAllStatsInBytes.13.50 = 0
cnpdAllStatsInBytes.13.51 = 25078176
cnpdAllStatsInBytes.13.52 = 0
cnpdAllStatsInBytes.13.53 = 0
cnpdAllStatsInBytes.13.54 = 0
cnpdAllStatsInBytes.13.55 = 0
cnpdAllStatsInBytes.13.56 = 0
cnpdAllStatsInBytes.13.57 = 0
cnpdAllStatsInBytes.13.58 = 0
cnpdAllStatsInBytes.13.59 = 1128158
cnpdAllStatsInBytes.13.60 = 0
cnpdAllStatsInBytes.13.61 = 5164765
cnpdAllStatsInBytes.13.62 = 4214412
cnpdAllStatsInBytes.13.63 = 0
cnpdAllStatsInBytes.13.64 = 0
cnpdAllStatsInBytes.13.65 = 0
cnpdAllStatsInBytes.13.66 = 0
cnpdAllStatsInBytes.13.67 = 0
cnpdAllStatsInBytes.13.68 = 633220
cnpdAllStatsInBytes.13.69 = 0
cnpdAllStatsInBytes.13.70 = 0
cnpdAllStatsInBytes.13.71 = 21057032
cnpdAllStatsInBytes.13.72 = 2339425
cnpdAllStatsOutBytes.13.1 = 0
cnpdAllStatsOutBytes.13.2 = 0
cnpdAllStatsOutBytes.13.3 = 0
cnpdAllStatsOutBytes.13.4 = 0
cnpdAllStatsOutBytes.13.5 = 0
cnpdAllStatsOutBytes.13.6 = 0
```

```
cnpdAllStatsOutBytes.13.7 = 0
cnpdAllStatsOutBytes.13.8 = 0
cnpdAllStatsOutBytes.13.9 = 0
cnpdAllStatsOutBytes.13.10 = 0
cnpdAllStatsOutBytes.13.11 = 0
cnpdAllStatsOutBytes.13.12 = 0
cnpdAllStatsOutBytes.13.13 = 0
cnpdAllStatsOutBytes.13.14 = 0
cnpdAllStatsOutBytes.13.15 = 0
cnpdAllStatsOutBytes.13.16 = 0
cnpdAllStatsOutBytes.13.17 = 0
cnpdAllStatsOutBytes.13.18 = 0
cnpdAllStatsOutBytes.13.19 = 0
cnpdAllStatsOutBytes.13.20 = 0
cnpdAllStatsOutBytes.13.21 = 0
cnpdAllStatsOutBytes.13.22 = 0
cnpdAllStatsOutBytes.13.23 = 0
cnpdAllStatsOutBytes.13.24 = 0
cnpdAllStatsOutBytes.13.25 = 0
cnpdAllStatsOutBytes.13.26 = 0
cnpdAllStatsOutBytes.13.27 = 0
cnpdAllStatsOutBytes.13.28 = 0
cnpdAllStatsOutBytes.13.29 = 0
cnpdAllStatsOutBytes.13.30 = 0
cnpdAllStatsOutBytes.13.31 = 0
cnpdAllStatsOutBytes.13.32 = 0
cnpdAllStatsOutBytes.13.33 = 0
cnpdAllStatsOutBytes.13.34 = 0
cnpdAllStatsOutBytes.13.35 = 0
cnpdAllStatsOutBytes.13.36 = 0
cnpdAllStatsOutBytes.13.37 = 0
cnpdAllStatsOutBytes.13.38 = 0
cnpdAllStatsOutBytes.13.39 = 0
cnpdAllStatsOutBytes.13.40 = 0
cnpdAllStatsOutBytes.13.41 = 0
cnpdAllStatsOutBytes.13.42 = 0
cnpdAllStatsOutBytes.13.43 = 0
cnpdAllStatsOutBytes.13.44 = 0
cnpdAllStatsOutBytes.13.45 = 0
cnpdAllStatsOutBytes.13.46 = 0
cnpdAllStatsOutBytes.13.47 = 0
cnpdAllStatsOutBytes.13.48 = 0
cnpdAllStatsOutBytes.13.49 = 0
cnpdAllStatsOutBytes.13.50 = 0
cnpdAllStatsOutBytes.13.51 = 0
cnpdAllStatsOutBytes.13.52 = 0
cnpdAllStatsOutBytes.13.53 = 0
cnpdAllStatsOutBytes.13.54 = 0
cnpdAllStatsOutBytes.13.55 = 0
cnpdAllStatsOutBytes.13.56 = 0
cnpdAllStatsOutBytes.13.57 = 0
cnpdAllStatsOutBytes.13.58 = 0
cnpdAllStatsOutBytes.13.59 = 0
```

```
cnpdAllStatsOutBytes.13.60 = 0
cnpdAllStatsOutBytes.13.61 = 0
cnpdAllStatsOutBytes.13.62 = 0
cnpdAllStatsOutBytes.13.63 = 0
cnpdAllStatsOutBytes.13.64 = 0
cnpdAllStatsOutBytes.13.65 = 0
cnpdAllStatsOutBytes.13.66 = 0
cnpdAllStatsOutBytes.13.67 = 0
cnpdAllStatsOutBytes.13.68 = 0
cnpdAllStatsOutBytes.13.69 = 0
cnpdAllStatsOutBytes.13.70 = 0
cnpdAllStatsOutBytes.13.71 = 0
cnpdAllStatsOutBytes.13.72 = 0
cnpdAllStatsHCInPkts.13.1 = 0x00000000
cnpdAllStatsHCInPkts.13.2 = 0x00000000
cnpdAllStatsHCInPkts.13.3 = 0x00001a768
cnpdAllStatsHCInPkts.13.4 = 0x00000000
cnpdAllStatsHCInPkts.13.5 = 0x000382cdd
cnpdAllStatsHCInPkts.13.6 = 0x00000000
cnpdAllStatsHCInPkts.13.7 = 0x00000000
cnpdAllStatsHCInPkts.13.8 = 0x00000000
cnpdAllStatsHCInPkts.13.9 = 0x00000000
cnpdAllStatsHCInPkts.13.10 = 0x00000000
cnpdAllStatsHCInPkts.13.11 = 0x00000000
cnpdAllStatsHCInPkts.13.12 = 0x00000000
cnpdAllStatsHCInPkts.13.13 = 0x00000000
cnpdAllStatsHCInPkts.13.14 = 0x00000000
cnpdAllStatsHCInPkts.13.15 = 0x00000000
cnpdAllStatsHCInPkts.13.16 = 0x00000000
cnpdAllStatsHCInPkts.13.17 = 0x00000000
cnpdAllStatsHCInPkts.13.18 = 0x00000000
cnpdAllStatsHCInPkts.13.19 = 0x00000000
cnpdAllStatsHCInPkts.13.20 = 0x00000000
cnpdAllStatsHCInPkts.13.21 = 0x00000000
cnpdAllStatsHCInPkts.13.22 = 0x00000000
cnpdAllStatsHCInPkts.13.23 = 0x000149348
cnpdAllStatsHCInPkts.13.24 = 0x00000000
cnpdAllStatsHCInPkts.13.25 = 0x00000000
cnpdAllStatsHCInPkts.13.26 = 0x00000000
cnpdAllStatsHCInPkts.13.27 = 0x00000000
cnpdAllStatsHCInPkts.13.28 = 0x00000000
cnpdAllStatsHCInPkts.13.29 = 0x00000000
cnpdAllStatsHCInPkts.13.30 = 0x00000000
cnpdAllStatsHCInPkts.13.31 = 0x00000000
cnpdAllStatsHCInPkts.13.32 = 0x00000000
cnpdAllStatsHCInPkts.13.33 = 0x00000000
cnpdAllStatsHCInPkts.13.34 = 0x00000000
cnpdAllStatsHCInPkts.13.35 = 0x00000000
cnpdAllStatsHCInPkts.13.36 = 0x000019e66
cnpdAllStatsHCInPkts.13.37 = 0x00000000
cnpdAllStatsHCInPkts.13.38 = 0x00000000
cnpdAllStatsHCInPkts.13.39 = 0x00000000
cnpdAllStatsHCInPkts.13.40 = 0x00000000
```

```
cnpdAllStatsHCInPkts.13.41 = 0x000004615
cnpdAllStatsHCInPkts.13.42 = 0x000000000
cnpdAllStatsHCInPkts.13.43 = 0x000000000
cnpdAllStatsHCInPkts.13.44 = 0x000000000
cnpdAllStatsHCInPkts.13.45 = 0x000000000
cnpdAllStatsHCInPkts.13.46 = 0x0000894a1
cnpdAllStatsHCInPkts.13.47 = 0x000000000
cnpdAllStatsHCInPkts.13.48 = 0x000013e29
cnpdAllStatsHCInPkts.13.49 = 0x000000000
cnpdAllStatsHCInPkts.13.50 = 0x000000000
cnpdAllStatsHCInPkts.13.51 = 0x00005994a
cnpdAllStatsHCInPkts.13.52 = 0x000000000
cnpdAllStatsHCInPkts.13.53 = 0x000000000
cnpdAllStatsHCInPkts.13.54 = 0x000000000
cnpdAllStatsHCInPkts.13.55 = 0x000000000
cnpdAllStatsHCInPkts.13.56 = 0x000000000
cnpdAllStatsHCInPkts.13.57 = 0x000000000
cnpdAllStatsHCInPkts.13.58 = 0x000000000
cnpdAllStatsHCInPkts.13.59 = 0x0000043a6
cnpdAllStatsHCInPkts.13.60 = 0x000000000
cnpdAllStatsHCInPkts.13.61 = 0x000008d56
cnpdAllStatsHCInPkts.13.62 = 0x000002f66
cnpdAllStatsHCInPkts.13.63 = 0x000000000
cnpdAllStatsHCInPkts.13.64 = 0x000000000
cnpdAllStatsHCInPkts.13.65 = 0x000000000
cnpdAllStatsHCInPkts.13.66 = 0x000000000
cnpdAllStatsHCInPkts.13.67 = 0x000000000
cnpdAllStatsHCInPkts.13.68 = 0x000002356
cnpdAllStatsHCInPkts.13.69 = 0x000000000
cnpdAllStatsHCInPkts.13.70 = 0x000000000
cnpdAllStatsHCInPkts.13.71 = 0x00004c2aa
cnpdAllStatsHCInPkts.13.72 = 0x000006a0e
cnpdAllStatsHCOutPkts.13.1 = 0x000000000
cnpdAllStatsHCOutPkts.13.2 = 0x000000000
cnpdAllStatsHCOutPkts.13.3 = 0x000000000
cnpdAllStatsHCOutPkts.13.4 = 0x000000000
cnpdAllStatsHCOutPkts.13.5 = 0x000000000
cnpdAllStatsHCOutPkts.13.6 = 0x000000000
cnpdAllStatsHCOutPkts.13.7 = 0x000000000
cnpdAllStatsHCOutPkts.13.8 = 0x000000000
cnpdAllStatsHCOutPkts.13.9 = 0x000000000
cnpdAllStatsHCOutPkts.13.10 = 0x000000000
cnpdAllStatsHCOutPkts.13.11 = 0x000000000
cnpdAllStatsHCOutPkts.13.12 = 0x000000000
cnpdAllStatsHCOutPkts.13.13 = 0x000000000
cnpdAllStatsHCOutPkts.13.14 = 0x000000000
cnpdAllStatsHCOutPkts.13.15 = 0x000000000
cnpdAllStatsHCOutPkts.13.16 = 0x000000000
cnpdAllStatsHCOutPkts.13.17 = 0x000000000
cnpdAllStatsHCOutPkts.13.18 = 0x000000000
cnpdAllStatsHCOutPkts.13.19 = 0x000000000
cnpdAllStatsHCOutPkts.13.20 = 0x000000000
cnpdAllStatsHCOutPkts.13.21 = 0x000000000
```

```
cnpdAllStatsHCOutPkts.13.22 = 0x00000000
cnpdAllStatsHCOutPkts.13.23 = 0x00000000
cnpdAllStatsHCOutPkts.13.24 = 0x00000000
cnpdAllStatsHCOutPkts.13.25 = 0x00000000
cnpdAllStatsHCOutPkts.13.26 = 0x00000000
cnpdAllStatsHCOutPkts.13.27 = 0x00000000
cnpdAllStatsHCOutPkts.13.28 = 0x00000000
cnpdAllStatsHCOutPkts.13.29 = 0x00000000
cnpdAllStatsHCOutPkts.13.30 = 0x00000000
cnpdAllStatsHCOutPkts.13.31 = 0x00000000
cnpdAllStatsHCOutPkts.13.32 = 0x00000000
cnpdAllStatsHCOutPkts.13.33 = 0x00000000
cnpdAllStatsHCOutPkts.13.34 = 0x00000000
cnpdAllStatsHCOutPkts.13.35 = 0x00000000
cnpdAllStatsHCOutPkts.13.36 = 0x00000000
cnpdAllStatsHCOutPkts.13.37 = 0x00000000
cnpdAllStatsHCOutPkts.13.38 = 0x00000000
cnpdAllStatsHCOutPkts.13.39 = 0x00000000
cnpdAllStatsHCOutPkts.13.40 = 0x00000000
cnpdAllStatsHCOutPkts.13.41 = 0x00000000
cnpdAllStatsHCOutPkts.13.42 = 0x00000000
cnpdAllStatsHCOutPkts.13.43 = 0x00000000
cnpdAllStatsHCOutPkts.13.44 = 0x00000000
cnpdAllStatsHCOutPkts.13.45 = 0x00000000
cnpdAllStatsHCOutPkts.13.46 = 0x00000000
cnpdAllStatsHCOutPkts.13.47 = 0x00000000
cnpdAllStatsHCOutPkts.13.48 = 0x00000000
cnpdAllStatsHCOutPkts.13.49 = 0x00000000
cnpdAllStatsHCOutPkts.13.50 = 0x00000000
cnpdAllStatsHCOutPkts.13.51 = 0x00000000
cnpdAllStatsHCOutPkts.13.52 = 0x00000000
cnpdAllStatsHCOutPkts.13.53 = 0x00000000
cnpdAllStatsHCOutPkts.13.54 = 0x00000000
cnpdAllStatsHCOutPkts.13.55 = 0x00000000
cnpdAllStatsHCOutPkts.13.56 = 0x00000000
cnpdAllStatsHCOutPkts.13.57 = 0x00000000
cnpdAllStatsHCOutPkts.13.58 = 0x00000000
cnpdAllStatsHCOutPkts.13.59 = 0x00000000
cnpdAllStatsHCOutPkts.13.60 = 0x00000000
cnpdAllStatsHCOutPkts.13.61 = 0x00000000
cnpdAllStatsHCOutPkts.13.62 = 0x00000000
cnpdAllStatsHCOutPkts.13.63 = 0x00000000
cnpdAllStatsHCOutPkts.13.64 = 0x00000000
cnpdAllStatsHCOutPkts.13.65 = 0x00000000
cnpdAllStatsHCOutPkts.13.66 = 0x00000000
cnpdAllStatsHCOutPkts.13.67 = 0x00000000
cnpdAllStatsHCOutPkts.13.68 = 0x00000000
cnpdAllStatsHCOutPkts.13.69 = 0x00000000
cnpdAllStatsHCOutPkts.13.70 = 0x00000000
cnpdAllStatsHCOutPkts.13.71 = 0x00000000
cnpdAllStatsHCOutPkts.13.72 = 0x00000000
cnpdAllStatsHCInBytes.13.1 = 0x00000000
cnpdAllStatsHCInBytes.13.2 = 0x00000000
```



```
cnpdAllStatsHCInBytes.13.3 = 0x0013395f9
cnpdAllStatsHCInBytes.13.4 = 0x000000000
cnpdAllStatsHCInBytes.13.5 = 0x086725280
cnpdAllStatsHCInBytes.13.6 = 0x000000000
cnpdAllStatsHCInBytes.13.7 = 0x000000000
cnpdAllStatsHCInBytes.13.8 = 0x000000000
cnpdAllStatsHCInBytes.13.9 = 0x000000000
cnpdAllStatsHCInBytes.13.10 = 0x000000000
cnpdAllStatsHCInBytes.13.11 = 0x000000000
cnpdAllStatsHCInBytes.13.12 = 0x000000000
cnpdAllStatsHCInBytes.13.13 = 0x000000000
cnpdAllStatsHCInBytes.13.14 = 0x000000000
cnpdAllStatsHCInBytes.13.15 = 0x000000000
cnpdAllStatsHCInBytes.13.16 = 0x000000000
cnpdAllStatsHCInBytes.13.17 = 0x000000000
cnpdAllStatsHCInBytes.13.18 = 0x000000000
cnpdAllStatsHCInBytes.13.19 = 0x000000000
cnpdAllStatsHCInBytes.13.20 = 0x000000000
cnpdAllStatsHCInBytes.13.21 = 0x000000000
cnpdAllStatsHCInBytes.13.22 = 0x000000000
cnpdAllStatsHCInBytes.13.23 = 0x01bb01d54
cnpdAllStatsHCInBytes.13.24 = 0x000000000
cnpdAllStatsHCInBytes.13.25 = 0x000000000
cnpdAllStatsHCInBytes.13.26 = 0x000000000
cnpdAllStatsHCInBytes.13.27 = 0x000000000
cnpdAllStatsHCInBytes.13.28 = 0x000000000
cnpdAllStatsHCInBytes.13.29 = 0x000000000
cnpdAllStatsHCInBytes.13.30 = 0x000000000
cnpdAllStatsHCInBytes.13.31 = 0x000000000
cnpdAllStatsHCInBytes.13.32 = 0x000000000
cnpdAllStatsHCInBytes.13.33 = 0x000000000
cnpdAllStatsHCInBytes.13.34 = 0x000000000
cnpdAllStatsHCInBytes.13.35 = 0x000000000
cnpdAllStatsHCInBytes.13.36 = 0x0009cdfc
cnpdAllStatsHCInBytes.13.37 = 0x000000000
cnpdAllStatsHCInBytes.13.38 = 0x000000000
cnpdAllStatsHCInBytes.13.39 = 0x000000000
cnpdAllStatsHCInBytes.13.40 = 0x000000000
cnpdAllStatsHCInBytes.13.41 = 0x00013d148
cnpdAllStatsHCInBytes.13.42 = 0x000000000
cnpdAllStatsHCInBytes.13.43 = 0x000000000
cnpdAllStatsHCInBytes.13.44 = 0x000000000
cnpdAllStatsHCInBytes.13.45 = 0x000000000
cnpdAllStatsHCInBytes.13.46 = 0x008e4823b
cnpdAllStatsHCInBytes.13.47 = 0x000000000
cnpdAllStatsHCInBytes.13.48 = 0x000d86339
cnpdAllStatsHCInBytes.13.49 = 0x000000000
cnpdAllStatsHCInBytes.13.50 = 0x000000000
cnpdAllStatsHCInBytes.13.51 = 0x0017ea9a0
cnpdAllStatsHCInBytes.13.52 = 0x000000000
cnpdAllStatsHCInBytes.13.53 = 0x000000000
cnpdAllStatsHCInBytes.13.54 = 0x000000000
cnpdAllStatsHCInBytes.13.55 = 0x000000000
```

```
cnpdAllStatsHCInBytes.13.56 = 0x000000000
cnpdAllStatsHCInBytes.13.57 = 0x000000000
cnpdAllStatsHCInBytes.13.58 = 0x000000000
cnpdAllStatsHCInBytes.13.59 = 0x0001136de
cnpdAllStatsHCInBytes.13.60 = 0x000000000
cnpdAllStatsHCInBytes.13.61 = 0x0004ecedd
cnpdAllStatsHCInBytes.13.62 = 0x000404e8c
cnpdAllStatsHCInBytes.13.63 = 0x000000000
cnpdAllStatsHCInBytes.13.64 = 0x000000000
cnpdAllStatsHCInBytes.13.65 = 0x000000000
cnpdAllStatsHCInBytes.13.66 = 0x000000000
cnpdAllStatsHCInBytes.13.67 = 0x000000000
cnpdAllStatsHCInBytes.13.68 = 0x00009a984
cnpdAllStatsHCInBytes.13.69 = 0x000000000
cnpdAllStatsHCInBytes.13.70 = 0x000000000
cnpdAllStatsHCInBytes.13.71 = 0x001414e08
cnpdAllStatsHCInBytes.13.72 = 0x00023b261
cnpdAllStatsHCOutBytes.13.1 = 0x000000000
cnpdAllStatsHCOutBytes.13.2 = 0x000000000
cnpdAllStatsHCOutBytes.13.3 = 0x000000000
cnpdAllStatsHCOutBytes.13.4 = 0x000000000
cnpdAllStatsHCOutBytes.13.5 = 0x000000000
cnpdAllStatsHCOutBytes.13.6 = 0x000000000
cnpdAllStatsHCOutBytes.13.7 = 0x000000000
cnpdAllStatsHCOutBytes.13.8 = 0x000000000
cnpdAllStatsHCOutBytes.13.9 = 0x000000000
cnpdAllStatsHCOutBytes.13.10 = 0x000000000
cnpdAllStatsHCOutBytes.13.11 = 0x000000000
cnpdAllStatsHCOutBytes.13.12 = 0x000000000
cnpdAllStatsHCOutBytes.13.13 = 0x000000000
cnpdAllStatsHCOutBytes.13.14 = 0x000000000
cnpdAllStatsHCOutBytes.13.15 = 0x000000000
cnpdAllStatsHCOutBytes.13.16 = 0x000000000
cnpdAllStatsHCOutBytes.13.17 = 0x000000000
cnpdAllStatsHCOutBytes.13.18 = 0x000000000
cnpdAllStatsHCOutBytes.13.19 = 0x000000000
cnpdAllStatsHCOutBytes.13.20 = 0x000000000
cnpdAllStatsHCOutBytes.13.21 = 0x000000000
cnpdAllStatsHCOutBytes.13.22 = 0x000000000
cnpdAllStatsHCOutBytes.13.23 = 0x000000000
cnpdAllStatsHCOutBytes.13.24 = 0x000000000
cnpdAllStatsHCOutBytes.13.25 = 0x000000000
cnpdAllStatsHCOutBytes.13.26 = 0x000000000
cnpdAllStatsHCOutBytes.13.27 = 0x000000000
cnpdAllStatsHCOutBytes.13.28 = 0x000000000
cnpdAllStatsHCOutBytes.13.29 = 0x000000000
cnpdAllStatsHCOutBytes.13.30 = 0x000000000
cnpdAllStatsHCOutBytes.13.31 = 0x000000000
cnpdAllStatsHCOutBytes.13.32 = 0x000000000
cnpdAllStatsHCOutBytes.13.33 = 0x000000000
cnpdAllStatsHCOutBytes.13.34 = 0x000000000
cnpdAllStatsHCOutBytes.13.35 = 0x000000000
cnpdAllStatsHCOutBytes.13.36 = 0x000000000
```

```
cnpdAllStatsHCOutBytes.13.37 = 0x00000000
cnpdAllStatsHCOutBytes.13.38 = 0x00000000
cnpdAllStatsHCOutBytes.13.39 = 0x00000000
cnpdAllStatsHCOutBytes.13.40 = 0x00000000
cnpdAllStatsHCOutBytes.13.41 = 0x00000000
cnpdAllStatsHCOutBytes.13.42 = 0x00000000
cnpdAllStatsHCOutBytes.13.43 = 0x00000000
cnpdAllStatsHCOutBytes.13.44 = 0x00000000
cnpdAllStatsHCOutBytes.13.45 = 0x00000000
cnpdAllStatsHCOutBytes.13.46 = 0x00000000
cnpdAllStatsHCOutBytes.13.47 = 0x00000000
cnpdAllStatsHCOutBytes.13.48 = 0x00000000
cnpdAllStatsHCOutBytes.13.49 = 0x00000000
cnpdAllStatsHCOutBytes.13.50 = 0x00000000
cnpdAllStatsHCOutBytes.13.51 = 0x00000000
cnpdAllStatsHCOutBytes.13.52 = 0x00000000
cnpdAllStatsHCOutBytes.13.53 = 0x00000000
cnpdAllStatsHCOutBytes.13.54 = 0x00000000
cnpdAllStatsHCOutBytes.13.55 = 0x00000000
cnpdAllStatsHCOutBytes.13.56 = 0x00000000
cnpdAllStatsHCOutBytes.13.57 = 0x00000000
cnpdAllStatsHCOutBytes.13.58 = 0x00000000
cnpdAllStatsHCOutBytes.13.59 = 0x00000000
cnpdAllStatsHCOutBytes.13.60 = 0x00000000
cnpdAllStatsHCOutBytes.13.61 = 0x00000000
cnpdAllStatsHCOutBytes.13.62 = 0x00000000
cnpdAllStatsHCOutBytes.13.63 = 0x00000000
cnpdAllStatsHCOutBytes.13.64 = 0x00000000
cnpdAllStatsHCOutBytes.13.65 = 0x00000000
cnpdAllStatsHCOutBytes.13.66 = 0x00000000
cnpdAllStatsHCOutBytes.13.67 = 0x00000000
cnpdAllStatsHCOutBytes.13.68 = 0x00000000
cnpdAllStatsHCOutBytes.13.69 = 0x00000000
cnpdAllStatsHCOutBytes.13.70 = 0x00000000
cnpdAllStatsHCOutBytes.13.71 = 0x00000000
cnpdAllStatsHCOutBytes.13.72 = 0x00000000
cnpdAllStatsInBitRate.13.1 = 0
cnpdAllStatsInBitRate.13.2 = 0
cnpdAllStatsInBitRate.13.3 = 0
cnpdAllStatsInBitRate.13.4 = 0
cnpdAllStatsInBitRate.13.5 = 0
cnpdAllStatsInBitRate.13.6 = 0
cnpdAllStatsInBitRate.13.7 = 0
cnpdAllStatsInBitRate.13.8 = 0
cnpdAllStatsInBitRate.13.9 = 0
cnpdAllStatsInBitRate.13.10 = 0
cnpdAllStatsInBitRate.13.11 = 0
cnpdAllStatsInBitRate.13.12 = 0
cnpdAllStatsInBitRate.13.13 = 0
cnpdAllStatsInBitRate.13.14 = 0
cnpdAllStatsInBitRate.13.15 = 0
cnpdAllStatsInBitRate.13.16 = 0
cnpdAllStatsInBitRate.13.17 = 0
```

cnpdAllStatsInBitRate.13.18 = 0
cnpdAllStatsInBitRate.13.19 = 0
cnpdAllStatsInBitRate.13.20 = 0
cnpdAllStatsInBitRate.13.21 = 0
cnpdAllStatsInBitRate.13.22 = 0
cnpdAllStatsInBitRate.13.23 = 0
cnpdAllStatsInBitRate.13.24 = 0
cnpdAllStatsInBitRate.13.25 = 0
cnpdAllStatsInBitRate.13.26 = 0
cnpdAllStatsInBitRate.13.27 = 0
cnpdAllStatsInBitRate.13.28 = 0
cnpdAllStatsInBitRate.13.29 = 0
cnpdAllStatsInBitRate.13.30 = 0
cnpdAllStatsInBitRate.13.31 = 0
cnpdAllStatsInBitRate.13.32 = 0
cnpdAllStatsInBitRate.13.33 = 0
cnpdAllStatsInBitRate.13.34 = 0
cnpdAllStatsInBitRate.13.35 = 0
cnpdAllStatsInBitRate.13.36 = 0
cnpdAllStatsInBitRate.13.37 = 0
cnpdAllStatsInBitRate.13.38 = 0
cnpdAllStatsInBitRate.13.39 = 0
cnpdAllStatsInBitRate.13.40 = 0
cnpdAllStatsInBitRate.13.41 = 0
cnpdAllStatsInBitRate.13.42 = 0
cnpdAllStatsInBitRate.13.43 = 0
cnpdAllStatsInBitRate.13.44 = 0
cnpdAllStatsInBitRate.13.45 = 0
cnpdAllStatsInBitRate.13.46 = 0
cnpdAllStatsInBitRate.13.47 = 0
cnpdAllStatsInBitRate.13.48 = 0
cnpdAllStatsInBitRate.13.49 = 0
cnpdAllStatsInBitRate.13.50 = 0
cnpdAllStatsInBitRate.13.51 = 0
cnpdAllStatsInBitRate.13.52 = 0
cnpdAllStatsInBitRate.13.53 = 0
cnpdAllStatsInBitRate.13.54 = 0
cnpdAllStatsInBitRate.13.55 = 0
cnpdAllStatsInBitRate.13.56 = 0
cnpdAllStatsInBitRate.13.57 = 0
cnpdAllStatsInBitRate.13.58 = 0
cnpdAllStatsInBitRate.13.59 = 0
cnpdAllStatsInBitRate.13.60 = 0
cnpdAllStatsInBitRate.13.61 = 0
cnpdAllStatsInBitRate.13.62 = 0
cnpdAllStatsInBitRate.13.63 = 0
cnpdAllStatsInBitRate.13.64 = 0
cnpdAllStatsInBitRate.13.65 = 0
cnpdAllStatsInBitRate.13.66 = 0
cnpdAllStatsInBitRate.13.67 = 0
cnpdAllStatsInBitRate.13.68 = 0
cnpdAllStatsInBitRate.13.69 = 0
cnpdAllStatsInBitRate.13.70 = 0

cnpdAllStatsInBitRate.13.71 = 0
cnpdAllStatsInBitRate.13.72 = 0
cnpdAllStatsOutBitRate.13.1 = 0
cnpdAllStatsOutBitRate.13.2 = 0
cnpdAllStatsOutBitRate.13.3 = 0
cnpdAllStatsOutBitRate.13.4 = 0
cnpdAllStatsOutBitRate.13.5 = 0
cnpdAllStatsOutBitRate.13.6 = 0
cnpdAllStatsOutBitRate.13.7 = 0
cnpdAllStatsOutBitRate.13.8 = 0
cnpdAllStatsOutBitRate.13.9 = 0
cnpdAllStatsOutBitRate.13.10 = 0
cnpdAllStatsOutBitRate.13.11 = 0
cnpdAllStatsOutBitRate.13.12 = 0
cnpdAllStatsOutBitRate.13.13 = 0
cnpdAllStatsOutBitRate.13.14 = 0
cnpdAllStatsOutBitRate.13.15 = 0
cnpdAllStatsOutBitRate.13.16 = 0
cnpdAllStatsOutBitRate.13.17 = 0
cnpdAllStatsOutBitRate.13.18 = 0
cnpdAllStatsOutBitRate.13.19 = 0
cnpdAllStatsOutBitRate.13.20 = 0
cnpdAllStatsOutBitRate.13.21 = 0
cnpdAllStatsOutBitRate.13.22 = 0
cnpdAllStatsOutBitRate.13.23 = 0
cnpdAllStatsOutBitRate.13.24 = 0
cnpdAllStatsOutBitRate.13.25 = 0
cnpdAllStatsOutBitRate.13.26 = 0
cnpdAllStatsOutBitRate.13.27 = 0
cnpdAllStatsOutBitRate.13.28 = 0
cnpdAllStatsOutBitRate.13.29 = 0
cnpdAllStatsOutBitRate.13.30 = 0
cnpdAllStatsOutBitRate.13.31 = 0
cnpdAllStatsOutBitRate.13.32 = 0
cnpdAllStatsOutBitRate.13.33 = 0
cnpdAllStatsOutBitRate.13.34 = 0
cnpdAllStatsOutBitRate.13.35 = 0
cnpdAllStatsOutBitRate.13.36 = 0
cnpdAllStatsOutBitRate.13.37 = 0
cnpdAllStatsOutBitRate.13.38 = 0
cnpdAllStatsOutBitRate.13.39 = 0
cnpdAllStatsOutBitRate.13.40 = 0
cnpdAllStatsOutBitRate.13.41 = 0
cnpdAllStatsOutBitRate.13.42 = 0
cnpdAllStatsOutBitRate.13.43 = 0
cnpdAllStatsOutBitRate.13.44 = 0
cnpdAllStatsOutBitRate.13.45 = 0
cnpdAllStatsOutBitRate.13.46 = 0
cnpdAllStatsOutBitRate.13.47 = 0
cnpdAllStatsOutBitRate.13.48 = 0
cnpdAllStatsOutBitRate.13.49 = 0
cnpdAllStatsOutBitRate.13.50 = 0
cnpdAllStatsOutBitRate.13.51 = 0

```

cnpdAllStatsOutBitRate.13.52 = 0
cnpdAllStatsOutBitRate.13.53 = 0
cnpdAllStatsOutBitRate.13.54 = 0
cnpdAllStatsOutBitRate.13.55 = 0
cnpdAllStatsOutBitRate.13.56 = 0
cnpdAllStatsOutBitRate.13.57 = 0
cnpdAllStatsOutBitRate.13.58 = 0
cnpdAllStatsOutBitRate.13.59 = 0
cnpdAllStatsOutBitRate.13.60 = 0
cnpdAllStatsOutBitRate.13.61 = 0
cnpdAllStatsOutBitRate.13.62 = 0
cnpdAllStatsOutBitRate.13.63 = 0
cnpdAllStatsOutBitRate.13.64 = 0
cnpdAllStatsOutBitRate.13.65 = 0
cnpdAllStatsOutBitRate.13.66 = 0
cnpdAllStatsOutBitRate.13.67 = 0
cnpdAllStatsOutBitRate.13.68 = 0
cnpdAllStatsOutBitRate.13.69 = 0
cnpdAllStatsOutBitRate.13.70 = 0
cnpdAllStatsOutBitRate.13.71 = 0
cnpdAllStatsOutBitRate.13.72 = 0

```

Creating Top-N Tables Using the NBAR Protocol Discovery MIB

The top-n functionality in the NBAR Protocol Discovery MIB involves two tables. The first table is a configuration table in which each row represents a group of objects that will create a unique top-n report. The other table is a Statistics Results Table that records the outputs of each row of the configuration table. The Statistics ResultsTable can be regenerated by the correct use of the rowStatus object in the configuration table.

Create a new TopNConfig Entry

```

>$ setany -v2c a.b.c.d public cnpdTopNConfigIfIndex.1 -i 13
cnpdTopNConfigIfIndex.1 = 13

```

```

>$ getmany -v2c a.b.c.d public cnpdTopNConfig

```

```

cnpdTopNConfigIfIndex.1 = 13
cnpdTopNConfigStatsSelect.1 = byteCountSum(6)
cnpdTopNConfigSampleTime.1 = 0
cnpdTopNConfigRequestedSize.1 = 0
cnpdTopNConfigGrantedSize.1 = 0
cnpdTopNConfigTime.1 = 0
cnpdTopNConfigStatus.1 = notReady(3)

```

Change StatsSelect

```
>$ setany -v2c a.b.c.d public cnpdTopNConfigStatsSelect.1 -i 7

cnpdTopNConfigStatsSelect.1 = packetCountIn(7)

>$ getmany -v2c a.b.c.d public cnpdTopNConfig

cnpdTopNConfigIfIndex.1 = 13
cnpdTopNConfigStatsSelect.1 = packetCountIn(7)
cnpdTopNConfigSampleTime.1 = 0
cnpdTopNConfigRequestedSize.1 = 0
cnpdTopNConfigGrantedSize.1 = 0
cnpdTopNConfigTime.1 = 0
cnpdTopNConfigStatus.1 = notReady(3)
```

Change SampleTime

```
>$ setany -v2c a.b.c.d public cnpdTopNConfigSampleTime.1 -g 13

cnpdTopNConfigSampleTime.1 = 13

>$ getmany -v2c a.b.c.d public cnpdTopNConfig

cnpdTopNConfigIfIndex.1 = 13
cnpdTopNConfigStatsSelect.1 = byteCountOut(5)
cnpdTopNConfigSampleTime.1 = 13
cnpdTopNConfigRequestedSize.1 = 0
cnpdTopNConfigGrantedSize.1 = 0
cnpdTopNConfigTime.1 = 0
cnpdTopNConfigStatus.1 = notReady(3)
```

Change RequestedSize

```
>$ setany -v2c a.b.c.d public cnpdTopNConfigRequestedSize.1 -g 5
```

```
cnpdTopNConfigRequestedSize.1 = 5
```

```
>$ getmany -v2c a.b.c.d public cnpdTopNConfig
```

```
cnpdTopNConfigIfIndex.1 = 13
cnpdTopNConfigStatsSelect.1 = byteCountOut(5)
cnpdTopNConfigSampleTime.1 = 13
cnpdTopNConfigRequestedSize.1 = 5
cnpdTopNConfigGrantedSize.1 = 5
cnpdTopNConfigTime.1 = 0
cnpdTopNConfigStatus.1 = notReady(3)
```

Set Status to createAndGo and Display Results

```
>$ setany -v2c a.b.c.d public cnpdTopNConfigStatus.1 -i 4
```

```
cnpdTopNConfigStatus.1 = createAndGo(4)
```

```
>$ getmany -v2c a.b.c.d public cnpdTopNConfig
```

```
cnpdTopNConfigIfIndex.1 = 13
cnpdTopNConfigStatsSelect.1 = packetCountIn(7)
cnpdTopNConfigSampleTime.1 = 13
cnpdTopNConfigRequestedSize.1 = 5
cnpdTopNConfigGrantedSize.1 = 5
cnpdTopNConfigTime.1 = 44093748
cnpdTopNConfigStatus.1 = active(1)
```

```
>$ getmany -v2c a.b.c.d public cnpdTopNStats
```

```
cnpdTopNStatsProtocolName.1.1 = http
cnpdTopNStatsProtocolName.1.2 = realaudio
cnpdTopNStatsProtocolName.1.3 = nntp
cnpdTopNStatsProtocolName.1.4 = ldap
cnpdTopNStatsProtocolName.1.5 = ftp
cnpdTopNStatsRate.1.1 = 3681501
cnpdTopNStatsRate.1.2 = 1348424
cnpdTopNStatsRate.1.3 = 562337
cnpdTopNStatsRate.1.4 = 366922
cnpdTopNStatsRate.1.5 = 311978
cnpdTopNStatsHCRate.1.1 = 0x000382cdd
cnpdTopNStatsHCRate.1.2 = 0x000149348
```



```
cnpdTopNStatsHCRate.1.3 = 0x0000894a1  
cnpdTopNStatsHCRate.1.4 = 0x00005994a  
cnpdTopNStatsHCRate.1.5 = 0x00004c2aa
```

Configuring Thresholds

The NBAR Protocol Discovery MIB can be used to set two types of thresholds—a threshold that reports a breach when thresholds for an individual protocol are crossed and a threshold that reports a breach when all NBAR-classifiable protocols or applications are crossed.

Specific Protocol Threshold

A specific protocol threshold is a threshold that monitors the traffic of a specific protocol. An SNMP trap is sent when this threshold is crossed.

The following items should be noted when configuring a specific protocol threshold:

- The `cnpdThresholdConfigProtocolAny` must be set to `FALSE(2)` and `cnpdThresholdConfigProtocol` must be set to a value that indicates the OID of the protocol being monitored. The OID for each protocol can be seen by querying `cnpdSupportedProtocolsTable` (for instance, use 5 to monitor HTTP).
- Use `cnpdThresholdConfigIfIndex` to select the interface and `cnpdThresholdConfigRising` and `cnpdThresholdConfigFalling` to set the rising and falling thresholds.
- Set the `cnpdThresholdConfigInterval` to configure the frequency with which thresholds should be checked.
- Set `cnpdThresholdConfigStatus` to 4 (`createAndGo`) and this config member will become active.

Example:

```
setany -v2c a.b.c.d public cnpdThresholdConfigProtocolAny.2 -i 2  
setany -v2c a.b.c.d public cnpdThresholdConfigProtocol.2 -g 5  
setany -v2c a.b.c.d public cnpdThresholdConfigStatsSelect.2 -i 7  
setany -v2c a.b.c.d public cnpdThresholdConfigIfIndex.2 -i 13  
setany -v2c a.b.c.d public cnpdThresholdConfigRising.2 -g 2000  
setany -v2c a.b.c.d public cnpdThresholdConfigFalling.2 -g 1000  
setany -v2c a.b.c.d public cnpdThresholdConfigStatus.2 -i 4
```

Any Protocol Threshold

The Any Protocol Threshold setting is used to report breaches and create history entries if ANY protocol crosses a rising or falling threshold. This threshold does not have a mechanism to report a breach only once (in a given time period). For instance, if HTTP breaches a rising threshold, a trap is sent only for the first breach. If another protocol, such as DHCP, breaches a rising threshold, a separate breach is reported.

The setup for any protocol is identical to a specific protocol except `cnpdThresholdConfigProtocolAny` does not have to be configured and it is unnecessary to specify a `cnpdThresholdConfigProtocol`.

Example:

```
setany -v2c a.b.c.d public cnpdThresholdConfigStatsSelect.1 -i 7
setany -v2c a.b.c.d public cnpdThresholdConfigIfIndex.1 -i 13
setany -v2c a.b.c.d public cnpdThresholdConfigRising.1 -g 30
setany -v2c a.b.c.d public cnpdThresholdConfigFalling.1 -g 20
setany -v2c a.b.c.d public cnpdThresholdConfigStatus.1 -i 4
```

Threshold Options

The `cnpdThresholdConfigSampleType` is used to change the way a threshold value is sampled and can be set to either of the following values:

- `absoluteValue(1)`
- `deltaValue(2)`

When the `cnpdThresholdConfigSampleType` is set to `absoluteValue`, the sampled statistic is compared to `cnpdThresholdConfigRising` and `cnpdThresholdConfigFalling` and a trap is sent if either of these thresholds are crossed. A `ThresholdHistory` event is also registered when either of these thresholds are crossed.



Note

Most Protocol Discovery statistics are aggregates and only rise. Therefore, only one threshold is likely to be breached.

When the `cnpdThresholdConfigSampleType` is set to `deltaValue`, the difference between the current sample and the previous sample is compared to `cnpdThresholdConfigRising` and `cnpdThresholdConfigFalling`. A `ThresholdHistory` event is registered only if this difference is greater or less than the compared values. This `deltaValue` therefore measures the gradient of statistic change is the value being tested and can lead to several breach events and traps.

Examples:

Any Protocol



Note

The `cnpdThresholdConfigProtocolAny` object is set to `TRUE` by default so it does not need to set the value manually.

```
>$ setany -v2c a.b.c.d public cnpdThresholdConfigStatsSelect.1 -i 7

cnpdThresholdConfigStatsSelect.1 = packetCountIn(7)

>$ setany -v2c a.b.c.d public cnpdThresholdConfigIfIndex.1 -i 13

cnpdThresholdConfigIfIndex.1 = 13

>$ setany -v2c a.b.c.d public cnpdThresholdConfigRising.1 -g 30

cnpdThresholdConfigRising.1 = 30

>$ setany -v2c a.b.c.d public cnpdThresholdConfigFalling.1 -g 20

cnpdThresholdConfigFalling.1 = 20

>$ setany -v2c a.b.c.d public cnpdThresholdConfigStatus.1 -i 4

cnpdThresholdConfigStatus.1 = createAndGo(4)

>$ getmany -v2c a.b.c.d public cnpdThresholdConfig

cnpdThresholdConfigIfIndex.1 = 13
cnpdThresholdConfigInterval.1 = 10
cnpdThresholdConfigSampleType.1 = absoluteValue(1)
cnpdThresholdConfigProtocol.1 = 62
cnpdThresholdConfigProtocolAny.1 = true(1)
cnpdThresholdConfigStatsSelect.1 = packetCountIn(7)
cnpdThresholdConfigStartup.1 = risingOrFalling(3)
cnpdThresholdConfigRising.1 = 30
cnpdThresholdConfigFalling.1 = 20
cnpdThresholdConfigStatus.1 = active(1)

>$ getmany -v2c a.b.c.d public cnpdThresholdHistory

cnpdThresholdHistoryConfigIndex.1 = 1
cnpdThresholdHistoryConfigIndex.2 = 1
cnpdThresholdHistoryConfigIndex.3 = 1
cnpdThresholdHistoryConfigIndex.4 = 1
cnpdThresholdHistoryConfigIndex.5 = 1
cnpdThresholdHistoryConfigIndex.6 = 1
cnpdThresholdHistoryConfigIndex.7 = 1
cnpdThresholdHistoryConfigIndex.8 = 1
cnpdThresholdHistoryConfigIndex.9 = 1
cnpdThresholdHistoryConfigIndex.10 = 1
cnpdThresholdHistoryConfigIndex.11 = 1
cnpdThresholdHistoryConfigIndex.12 = 1
cnpdThresholdHistoryConfigIndex.13 = 1
cnpdThresholdHistoryConfigIndex.14 = 1
cnpdThresholdHistoryValue.1 = 73906
cnpdThresholdHistoryValue.2 = 26985
cnpdThresholdHistoryValue.3 = 11250
```

```
cnpdThresholdHistoryValue.4 = 7380
cnpdThresholdHistoryValue.5 = 6552
cnpdThresholdHistoryValue.6 = 2184
cnpdThresholdHistoryValue.7 = 2160
cnpdThresholdHistoryValue.8 = 1620
cnpdThresholdHistoryValue.9 = 724
cnpdThresholdHistoryValue.10 = 364
cnpdThresholdHistoryValue.11 = 360
cnpdThresholdHistoryValue.12 = 181
cnpdThresholdHistoryValue.13 = 2
cnpdThresholdHistoryValue.14 = 31
cnpdThresholdHistoryType.1 = risingBreach(1)
cnpdThresholdHistoryType.2 = risingBreach(1)
cnpdThresholdHistoryType.3 = risingBreach(1)
cnpdThresholdHistoryType.4 = risingBreach(1)
cnpdThresholdHistoryType.5 = risingBreach(1)
cnpdThresholdHistoryType.6 = risingBreach(1)
cnpdThresholdHistoryType.7 = risingBreach(1)
cnpdThresholdHistoryType.8 = risingBreach(1)
cnpdThresholdHistoryType.9 = risingBreach(1)
cnpdThresholdHistoryType.10 = risingBreach(1)
cnpdThresholdHistoryType.11 = risingBreach(1)
cnpdThresholdHistoryType.12 = risingBreach(1)
cnpdThresholdHistoryType.13 = fallingBreach(2)
cnpdThresholdHistoryType.14 = risingBreach(1)
cnpdThresholdHistoryTime.1 = 8579
cnpdThresholdHistoryTime.2 = 8579
cnpdThresholdHistoryTime.3 = 8579
cnpdThresholdHistoryTime.4 = 8579
cnpdThresholdHistoryTime.5 = 8579
cnpdThresholdHistoryTime.6 = 8579
cnpdThresholdHistoryTime.7 = 8579
cnpdThresholdHistoryTime.8 = 8579
cnpdThresholdHistoryTime.9 = 8579
cnpdThresholdHistoryTime.10 = 8579
cnpdThresholdHistoryTime.11 = 8579
cnpdThresholdHistoryTime.12 = 8579
cnpdThresholdHistoryTime.13 = 8579
cnpdThresholdHistoryTime.14 = 97579
cnpdThresholdHistoryProtocol.1 = 5
cnpdThresholdHistoryProtocol.2 = 23
cnpdThresholdHistoryProtocol.3 = 46
cnpdThresholdHistoryProtocol.4 = 51
cnpdThresholdHistoryProtocol.5 = 71
cnpdThresholdHistoryProtocol.6 = 36
cnpdThresholdHistoryProtocol.7 = 3
cnpdThresholdHistoryProtocol.8 = 48
cnpdThresholdHistoryProtocol.9 = 61
cnpdThresholdHistoryProtocol.10 = 59
cnpdThresholdHistoryProtocol.11 = 41
cnpdThresholdHistoryProtocol.12 = 68
cnpdThresholdHistoryProtocol.13 = 62
cnpdThresholdHistoryProtocol.14 = 62
```

```
cnpdThresholdHistoryStatsSelect.1 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.2 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.3 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.4 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.5 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.6 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.7 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.8 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.9 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.10 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.11 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.12 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.13 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.14 = packetCountIn(7)
```

SpecificProtocol (HTTP)

```
>$ setany -v2c a.b.c.d public cnpdThresholdConfigProtocolAny.2 -i 2
cnpdThresholdConfigProtocolAny.2 = false(2)
>$ setany -v2c a.b.c.d public cnpdThresholdConfigProtocol.2 -g 5
cnpdThresholdConfigProtocol.2 = 5
>$ setany -v2c a.b.c.d public cnpdThresholdConfigStatsSelect.2 -i 7
cnpdThresholdConfigStatsSelect.2 = packetCountIn(7)
>$ setany -v2c a.b.c.d public cnpdThresholdConfigIfIndex.2 -i 13
cnpdThresholdConfigIfIndex.2 = 13
>$ setany -v2c a.b.c.d public cnpdThresholdConfigRising.2 -g 2000
cnpdThresholdConfigRising.2 = 2000
>$ setany -v2c a.b.c.d public cnpdThresholdConfigFalling.2 -g 1000
cnpdThresholdConfigFalling.2 = 1000
>$ setany -v2c a.b.c.d public cnpdThresholdConfigStatus.2 -i 4
cnpdThresholdConfigStatus.2 = createAndGo(4)
>$ getmany -v2c a.b.c.d public cnpdThresholdConfig
cnpdThresholdConfigIfIndex.1 = 13
cnpdThresholdConfigIfIndex.2 = 13
cnpdThresholdConfigInterval.1 = 10
```

```

cnpdThresholdConfigInterval.2 = 10
cnpdThresholdConfigSampleType.1 = absoluteValue(1)
cnpdThresholdConfigSampleType.2 = absoluteValue(1)
cnpdThresholdConfigProtocol.1 = 62
cnpdThresholdConfigProtocol.2 = 5
cnpdThresholdConfigProtocolAny.1 = true(1)
cnpdThresholdConfigProtocolAny.2 = false(2)
cnpdThresholdConfigStatsSelect.1 = packetCountIn(7)
cnpdThresholdConfigStatsSelect.2 = packetCountIn(7)
cnpdThresholdConfigStartup.1 = risingOrFalling(3)
cnpdThresholdConfigStartup.2 = risingOrFalling(3)
cnpdThresholdConfigRising.1 = 30
cnpdThresholdConfigRising.2 = 2000
cnpdThresholdConfigFalling.1 = 20
cnpdThresholdConfigFalling.2 = 1000
cnpdThresholdConfigStatus.1 = active(1)
cnpdThresholdConfigStatus.2 = active(1)

```

```
>$ getmany -v2c a.b.c.d public cnpdThresholdHistory
```

```

cnpdThresholdHistoryConfigIndex.1 = 1
cnpdThresholdHistoryConfigIndex.2 = 1
cnpdThresholdHistoryConfigIndex.3 = 1
cnpdThresholdHistoryConfigIndex.4 = 1
cnpdThresholdHistoryConfigIndex.5 = 1
cnpdThresholdHistoryConfigIndex.6 = 1
cnpdThresholdHistoryConfigIndex.7 = 1
cnpdThresholdHistoryConfigIndex.8 = 1
cnpdThresholdHistoryConfigIndex.9 = 1
cnpdThresholdHistoryConfigIndex.10 = 1
cnpdThresholdHistoryConfigIndex.11 = 1
cnpdThresholdHistoryConfigIndex.12 = 1
cnpdThresholdHistoryConfigIndex.13 = 1
cnpdThresholdHistoryConfigIndex.14 = 1
cnpdThresholdHistoryConfigIndex.15 = 2
cnpdThresholdHistoryValue.1 = 73906
cnpdThresholdHistoryValue.2 = 26985
cnpdThresholdHistoryValue.3 = 11250
cnpdThresholdHistoryValue.4 = 7380
cnpdThresholdHistoryValue.5 = 6552
cnpdThresholdHistoryValue.6 = 2184
cnpdThresholdHistoryValue.7 = 2160
cnpdThresholdHistoryValue.8 = 1620
cnpdThresholdHistoryValue.9 = 724
cnpdThresholdHistoryValue.10 = 364
cnpdThresholdHistoryValue.11 = 360
cnpdThresholdHistoryValue.12 = 181
cnpdThresholdHistoryValue.13 = 2
cnpdThresholdHistoryValue.14 = 31
cnpdThresholdHistoryValue.15 = 3681501
cnpdThresholdHistoryType.1 = risingBreach(1)
cnpdThresholdHistoryType.2 = risingBreach(1)
cnpdThresholdHistoryType.3 = risingBreach(1)

```

```
cnpdThresholdHistoryType.4 = risingBreach(1)
cnpdThresholdHistoryType.5 = risingBreach(1)
cnpdThresholdHistoryType.6 = risingBreach(1)
cnpdThresholdHistoryType.7 = risingBreach(1)
cnpdThresholdHistoryType.8 = risingBreach(1)
cnpdThresholdHistoryType.9 = risingBreach(1)
cnpdThresholdHistoryType.10 = risingBreach(1)
cnpdThresholdHistoryType.11 = risingBreach(1)
cnpdThresholdHistoryType.12 = risingBreach(1)
cnpdThresholdHistoryType.13 = fallingBreach(2)
cnpdThresholdHistoryType.14 = risingBreach(1)
cnpdThresholdHistoryType.15 = risingBreach(1)
cnpdThresholdHistoryTime.1 = 8579
cnpdThresholdHistoryTime.2 = 8579
cnpdThresholdHistoryTime.3 = 8579
cnpdThresholdHistoryTime.4 = 8579
cnpdThresholdHistoryTime.5 = 8579
cnpdThresholdHistoryTime.6 = 8579
cnpdThresholdHistoryTime.7 = 8579
cnpdThresholdHistoryTime.8 = 8579
cnpdThresholdHistoryTime.9 = 8579
cnpdThresholdHistoryTime.10 = 8579
cnpdThresholdHistoryTime.11 = 8579
cnpdThresholdHistoryTime.12 = 8579
cnpdThresholdHistoryTime.13 = 8579
cnpdThresholdHistoryTime.14 = 97579
cnpdThresholdHistoryTime.15 = 44261817
cnpdThresholdHistoryProtocol.1 = 5
cnpdThresholdHistoryProtocol.2 = 23
cnpdThresholdHistoryProtocol.3 = 46
cnpdThresholdHistoryProtocol.4 = 51
cnpdThresholdHistoryProtocol.5 = 71
cnpdThresholdHistoryProtocol.6 = 36
cnpdThresholdHistoryProtocol.7 = 3
cnpdThresholdHistoryProtocol.8 = 48
cnpdThresholdHistoryProtocol.9 = 61
cnpdThresholdHistoryProtocol.10 = 59
cnpdThresholdHistoryProtocol.11 = 41
cnpdThresholdHistoryProtocol.12 = 68
cnpdThresholdHistoryProtocol.13 = 62
cnpdThresholdHistoryProtocol.14 = 62
cnpdThresholdHistoryProtocol.15 = 5
cnpdThresholdHistoryStatsSelect.1 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.2 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.3 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.4 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.5 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.6 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.7 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.8 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.9 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.10 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.11 = packetCountIn(7)
```

```

cnpdThresholdHistoryStatsSelect.12 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.13 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.14 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.15 = packetCountIn(7)

```

**Note**

The traps shown below are sample traps that could be generated by traffic patterns that breach the above configuration. The traps are shown only for the sake of the example.

```

Received SNMPv1 Trap:
Community: public
Enterprise: ciscoNbarProtocolDiscoveryMIB
Agent-addr: a.b.c.d
Enterprise Specific trap.
Enterprise Specific trap: 1
Time Ticks: 44261817
cnpdThresholdConfigIfIndex.2 = 13
cnpdThresholdConfigStatsSelect.2 = packetCountIn(7)
cnpdThresholdHistoryValue.2 = 3681501
cnpdThresholdConfigRising.2 = 2000
cnpdThresholdHistoryProtocol.2 = 5
cnpdThresholdHistoryTime.2 = 44261817

```

```

Received SNMPv1 Trap:
Community: public
Enterprise: ciscoNbarProtocolDiscoveryMIB
Agent-addr: a.b.c.d
Enterprise Specific trap.
Enterprise Specific trap: 1
Time Ticks: 44278817
cnpdThresholdConfigIfIndex.2 = 13
cnpdThresholdConfigStatsSelect.2 = packetCountIn(7)
cnpdThresholdHistoryValue.2 = 3681501
cnpdThresholdConfigRising.2 = 2000
cnpdThresholdHistoryProtocol.2 = 5
cnpdThresholdHistoryTime.2 = 44278817

```

```

Received SNMPv1 Trap:
Community: public
Enterprise: ciscoNbarProtocolDiscoveryMIB
Agent-addr: a.b.c.d
Enterprise Specific trap.
Enterprise Specific trap: 1
Time Ticks: 44284817
cnpdThresholdConfigIfIndex.2 = 13
cnpdThresholdConfigStatsSelect.2 = packetCountIn(7)
cnpdThresholdHistoryValue.2 = 3681501

```



```
cnpdThresholdConfigRising.2 = 2000
cnpdThresholdHistoryProtocol.2 = 5
cnpdThresholdHistoryTime.2 = 44284817
```

AnyProtocol=TRUE and SampleType=Delta

```
>$ setany -v2c a.b.c.d public cnpdThresholdConfigStatsSelect.3 -i 7
```

```
cnpdThresholdConfigStatsSelect.3 = packetCountIn(7)
```

```
>$ setany -v2c a.b.c.d public cnpdThresholdConfigIfIndex.3 -i 13
```

```
cnpdThresholdConfigIfIndex.3 = 13
```

```
>$ setany -v2c a.b.c.d public cnpdThresholdConfigRising.3 -g 30
```

```
cnpdThresholdConfigRising.3 = 30
```

```
>$ setany -v2c a.b.c.d public cnpdThresholdConfigFalling.3 -g 20
```

```
cnpdThresholdConfigFalling.3 = 20
```

```
>$ setany -v2c a.b.c.d public cnpdThresholdConfigSampleType.3 -i 2
```

```
cnpdThresholdConfigSampleType.3 = deltaValue(2)
```

```
>$ setany -v2c a.b.c.d public cnpdThresholdConfigStatus.3 -i 4
```

```
cnpdThresholdConfigStatus.3 = createAndGo(4)
```

```
>$ getmany -v2c a.b.c.d public cnpdThresholdConfig
```

```
cnpdThresholdConfigIfIndex.1 = 13
```

```
cnpdThresholdConfigIfIndex.2 = 13
```

```
cnpdThresholdConfigIfIndex.3 = 13
```

```
cnpdThresholdConfigInterval.1 = 10
```

```
cnpdThresholdConfigInterval.2 = 10
```

```
cnpdThresholdConfigInterval.3 = 10
```

```
cnpdThresholdConfigSampleType.1 = absoluteValue(1)
```

```
cnpdThresholdConfigSampleType.2 = absoluteValue(1)
```

```
cnpdThresholdConfigSampleType.3 = deltaValue(2)
```

```
cnpdThresholdConfigProtocol.1 = 62
```

```
cnpdThresholdConfigProtocol.2 = 5
```

```
cnpdThresholdConfigProtocol.3 = 62
```

```
cnpdThresholdConfigProtocolAny.1 = true(1)
```

```
cnpdThresholdConfigProtocolAny.2 = false(2)
```

```
cnpdThresholdConfigProtocolAny.3 = true(1)
```

```
cnpdThresholdConfigStatsSelect.1 = packetCountIn(7)
```

```
cnpdThresholdConfigStatsSelect.2 = packetCountIn(7)
```

```
cnpdThresholdConfigStatsSelect.3 = packetCountIn(7)
```

```
cnpdThresholdConfigStartup.1 = risingOrFalling(3)
```

```

cnpdThresholdConfigStartup.2 = risingOrFalling(3)
cnpdThresholdConfigStartup.3 = risingOrFalling(3)
cnpdThresholdConfigRising.1 = 30
cnpdThresholdConfigRising.2 = 2000
cnpdThresholdConfigRising.3 = 30
cnpdThresholdConfigFalling.1 = 20
cnpdThresholdConfigFalling.2 = 1000
cnpdThresholdConfigFalling.3 = 20
cnpdThresholdConfigStatus.1 = active(1)
cnpdThresholdConfigStatus.2 = active(1)
cnpdThresholdConfigStatus.3 = active(1)

```

```
>$ getmany -v2c a.b.c.d public cnpdThresholdHistory
```

```

cnpdThresholdHistoryConfigIndex.1 = 1
cnpdThresholdHistoryConfigIndex.2 = 1
cnpdThresholdHistoryConfigIndex.3 = 1
cnpdThresholdHistoryConfigIndex.4 = 1
cnpdThresholdHistoryConfigIndex.5 = 1
cnpdThresholdHistoryConfigIndex.6 = 1
cnpdThresholdHistoryConfigIndex.7 = 1
cnpdThresholdHistoryConfigIndex.8 = 1
cnpdThresholdHistoryConfigIndex.9 = 1
cnpdThresholdHistoryConfigIndex.10 = 1
cnpdThresholdHistoryConfigIndex.11 = 1
cnpdThresholdHistoryConfigIndex.12 = 1
cnpdThresholdHistoryConfigIndex.13 = 1
cnpdThresholdHistoryConfigIndex.14 = 1
cnpdThresholdHistoryConfigIndex.15 = 2
cnpdThresholdHistoryConfigIndex.16 = 2
cnpdThresholdHistoryConfigIndex.17 = 2
cnpdThresholdHistoryConfigIndex.18 = 3
cnpdThresholdHistoryConfigIndex.19 = 3
cnpdThresholdHistoryConfigIndex.20 = 3
cnpdThresholdHistoryConfigIndex.21 = 3
cnpdThresholdHistoryConfigIndex.22 = 3
cnpdThresholdHistoryConfigIndex.23 = 3
cnpdThresholdHistoryConfigIndex.24 = 3
cnpdThresholdHistoryConfigIndex.25 = 3
cnpdThresholdHistoryConfigIndex.26 = 3
cnpdThresholdHistoryConfigIndex.27 = 3
cnpdThresholdHistoryConfigIndex.28 = 3
cnpdThresholdHistoryConfigIndex.29 = 3
cnpdThresholdHistoryConfigIndex.30 = 3
cnpdThresholdHistoryConfigIndex.31 = 3
cnpdThresholdHistoryValue.1 = 73906
cnpdThresholdHistoryValue.2 = 26985
cnpdThresholdHistoryValue.3 = 11250
cnpdThresholdHistoryValue.4 = 7380
cnpdThresholdHistoryValue.5 = 6552
cnpdThresholdHistoryValue.6 = 2184
cnpdThresholdHistoryValue.7 = 2160
cnpdThresholdHistoryValue.8 = 1620

```

cnpdThresholdHistoryValue.9 = 724
cnpdThresholdHistoryValue.10 = 364
cnpdThresholdHistoryValue.11 = 360
cnpdThresholdHistoryValue.12 = 181
cnpdThresholdHistoryValue.13 = 2
cnpdThresholdHistoryValue.14 = 31
cnpdThresholdHistoryValue.15 = 3681501
cnpdThresholdHistoryValue.16 = 3681501
cnpdThresholdHistoryValue.17 = 3681501
cnpdThresholdHistoryValue.18 = 3681501
cnpdThresholdHistoryValue.19 = 1348424
cnpdThresholdHistoryValue.20 = 562337
cnpdThresholdHistoryValue.21 = 366922
cnpdThresholdHistoryValue.22 = 311978
cnpdThresholdHistoryValue.23 = 108392
cnpdThresholdHistoryValue.24 = 106086
cnpdThresholdHistoryValue.25 = 81449
cnpdThresholdHistoryValue.26 = 36182
cnpdThresholdHistoryValue.27 = 17941
cnpdThresholdHistoryValue.28 = 17318
cnpdThresholdHistoryValue.29 = 12197
cnpdThresholdHistoryValue.30 = 9046
cnpdThresholdHistoryValue.31 = 1
cnpdThresholdHistoryType.1 = risingBreach(1)
cnpdThresholdHistoryType.2 = risingBreach(1)
cnpdThresholdHistoryType.3 = risingBreach(1)
cnpdThresholdHistoryType.4 = risingBreach(1)
cnpdThresholdHistoryType.5 = risingBreach(1)
cnpdThresholdHistoryType.6 = risingBreach(1)
cnpdThresholdHistoryType.7 = risingBreach(1)
cnpdThresholdHistoryType.8 = risingBreach(1)
cnpdThresholdHistoryType.9 = risingBreach(1)
cnpdThresholdHistoryType.10 = risingBreach(1)
cnpdThresholdHistoryType.11 = risingBreach(1)
cnpdThresholdHistoryType.12 = risingBreach(1)
cnpdThresholdHistoryType.13 = fallingBreach(2)
cnpdThresholdHistoryType.14 = risingBreach(1)
cnpdThresholdHistoryType.15 = risingBreach(1)
cnpdThresholdHistoryType.16 = risingBreach(1)
cnpdThresholdHistoryType.17 = risingBreach(1)
cnpdThresholdHistoryType.18 = risingBreach(1)
cnpdThresholdHistoryType.19 = risingBreach(1)
cnpdThresholdHistoryType.20 = risingBreach(1)
cnpdThresholdHistoryType.21 = risingBreach(1)
cnpdThresholdHistoryType.22 = risingBreach(1)
cnpdThresholdHistoryType.23 = risingBreach(1)
cnpdThresholdHistoryType.24 = risingBreach(1)
cnpdThresholdHistoryType.25 = risingBreach(1)
cnpdThresholdHistoryType.26 = risingBreach(1)
cnpdThresholdHistoryType.27 = risingBreach(1)
cnpdThresholdHistoryType.28 = risingBreach(1)
cnpdThresholdHistoryType.29 = risingBreach(1)
cnpdThresholdHistoryType.30 = risingBreach(1)

```
cnpdThresholdHistoryType.31 = fallingBreach(2)
cnpdThresholdHistoryTime.1 = 8579
cnpdThresholdHistoryTime.2 = 8579
cnpdThresholdHistoryTime.3 = 8579
cnpdThresholdHistoryTime.4 = 8579
cnpdThresholdHistoryTime.5 = 8579
cnpdThresholdHistoryTime.6 = 8579
cnpdThresholdHistoryTime.7 = 8579
cnpdThresholdHistoryTime.8 = 8579
cnpdThresholdHistoryTime.9 = 8579
cnpdThresholdHistoryTime.10 = 8579
cnpdThresholdHistoryTime.11 = 8579
cnpdThresholdHistoryTime.12 = 8579
cnpdThresholdHistoryTime.13 = 8579
cnpdThresholdHistoryTime.14 = 97579
cnpdThresholdHistoryTime.15 = 44261817
cnpdThresholdHistoryTime.16 = 44278817
cnpdThresholdHistoryTime.17 = 44284817
cnpdThresholdHistoryTime.18 = 44288471
cnpdThresholdHistoryTime.19 = 44288471
cnpdThresholdHistoryTime.20 = 44288471
cnpdThresholdHistoryTime.21 = 44288471
cnpdThresholdHistoryTime.22 = 44288471
cnpdThresholdHistoryTime.23 = 44288471
cnpdThresholdHistoryTime.24 = 44288471
cnpdThresholdHistoryTime.25 = 44288471
cnpdThresholdHistoryTime.26 = 44288471
cnpdThresholdHistoryTime.27 = 44288471
cnpdThresholdHistoryTime.28 = 44288471
cnpdThresholdHistoryTime.29 = 44288471
cnpdThresholdHistoryTime.30 = 44288471
cnpdThresholdHistoryTime.31 = 44289471
cnpdThresholdHistoryProtocol.1 = 5
cnpdThresholdHistoryProtocol.2 = 23
cnpdThresholdHistoryProtocol.3 = 46
cnpdThresholdHistoryProtocol.4 = 51
cnpdThresholdHistoryProtocol.5 = 71
cnpdThresholdHistoryProtocol.6 = 36
cnpdThresholdHistoryProtocol.7 = 3
cnpdThresholdHistoryProtocol.8 = 48
cnpdThresholdHistoryProtocol.9 = 61
cnpdThresholdHistoryProtocol.10 = 59
cnpdThresholdHistoryProtocol.11 = 41
cnpdThresholdHistoryProtocol.12 = 68
cnpdThresholdHistoryProtocol.13 = 62
cnpdThresholdHistoryProtocol.14 = 62
cnpdThresholdHistoryProtocol.15 = 5
cnpdThresholdHistoryProtocol.16 = 5
cnpdThresholdHistoryProtocol.17 = 5
cnpdThresholdHistoryProtocol.18 = 5
cnpdThresholdHistoryProtocol.19 = 23
cnpdThresholdHistoryProtocol.20 = 46
cnpdThresholdHistoryProtocol.21 = 51
```

```
cnpdThresholdHistoryProtocol.22 = 71
cnpdThresholdHistoryProtocol.23 = 3
cnpdThresholdHistoryProtocol.24 = 36
cnpdThresholdHistoryProtocol.25 = 48
cnpdThresholdHistoryProtocol.26 = 61
cnpdThresholdHistoryProtocol.27 = 41
cnpdThresholdHistoryProtocol.28 = 59
cnpdThresholdHistoryProtocol.29 = 62
cnpdThresholdHistoryProtocol.30 = 68
cnpdThresholdHistoryProtocol.31 = 62
cnpdThresholdHistoryStatsSelect.1 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.2 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.3 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.4 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.5 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.6 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.7 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.8 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.9 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.10 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.11 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.12 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.13 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.14 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.15 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.16 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.17 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.18 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.19 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.20 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.21 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.22 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.23 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.24 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.25 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.26 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.27 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.28 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.29 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.30 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.31 = packetCountIn(7)
```

**Note**

The traps shown below are sample traps that could be generated by traffic patterns that breach the above configuration. The traps are shown only for the sake of the example.

```
Received SNMPv1 Trap:
Community: public
Enterprise: ciscoNbarProtocolDiscoveryMIB
Agent-addr: a.b.c.d
Enterprise Specific trap.
```

```
Enterprise Specific trap: 1
Time Ticks: 44288471
cnpdThresholdConfigIfIndex.3 = 13
cnpdThresholdConfigStatsSelect.3 = packetCountIn(7)
cnpdThresholdHistoryValue.3 = 366922
cnpdThresholdConfigRising.3 = 30
cnpdThresholdHistoryProtocol.3 = 51
cnpdThresholdHistoryTime.3 = 44288471
```

```
Received SNMPv1 Trap:
Community: public
Enterprise: ciscoNbarProtocolDiscoveryMIB
Agent-addr: a.b.c.d
Enterprise Specific trap.
Enterprise Specific trap: 1
Time Ticks: 44288471
cnpdThresholdConfigIfIndex.3 = 13
cnpdThresholdConfigStatsSelect.3 = packetCountIn(7)
cnpdThresholdHistoryValue.3 = 311978
cnpdThresholdConfigRising.3 = 30
cnpdThresholdHistoryProtocol.3 = 71
cnpdThresholdHistoryTime.3 = 44288471
```

```
Received SNMPv1 Trap:
Community: public
Enterprise: ciscoNbarProtocolDiscoveryMIB
Agent-addr: a.b.c.d
Enterprise Specific trap.
Enterprise Specific trap: 1
Time Ticks: 44288471
cnpdThresholdConfigIfIndex.3 = 13
cnpdThresholdConfigStatsSelect.3 = packetCountIn(7)
cnpdThresholdHistoryValue.3 = 108392
cnpdThresholdConfigRising.3 = 30
cnpdThresholdHistoryProtocol.3 = 3
cnpdThresholdHistoryTime.3 = 44288471
```

```
Received SNMPv1 Trap:
Community: public
Enterprise: ciscoNbarProtocolDiscoveryMIB
Agent-addr: a.b.c.d
Enterprise Specific trap.
Enterprise Specific trap: 1
Time Ticks: 44288471
cnpdThresholdConfigIfIndex.3 = 13
cnpdThresholdConfigStatsSelect.3 = packetCountIn(7)
cnpdThresholdHistoryValue.3 = 106086
cnpdThresholdConfigRising.3 = 30
cnpdThresholdHistoryProtocol.3 = 36
cnpdThresholdHistoryTime.3 = 44288471
```

```
Received SNMPv1 Trap:
Community: public
```

Enterprise: ciscoNbarProtocolDiscoveryMIB
Agent-addr: a.b.c.d
Enterprise Specific trap.
Enterprise Specific trap: 1
Time Ticks: 44288471
cnpdThresholdConfigIfIndex.3 = 13
cnpdThresholdConfigStatsSelect.3 = packetCountIn(7)
cnpdThresholdHistoryValue.3 = 81449
cnpdThresholdConfigRising.3 = 30
cnpdThresholdHistoryProtocol.3 = 48
cnpdThresholdHistoryTime.3 = 44288471

Received SNMPv1 Trap:
Community: public
Enterprise: ciscoNbarProtocolDiscoveryMIB
Agent-addr: a.b.c.d
Enterprise Specific trap.
Enterprise Specific trap: 1
Time Ticks: 44288471
cnpdThresholdConfigIfIndex.3 = 13
cnpdThresholdConfigStatsSelect.3 = packetCountIn(7)
cnpdThresholdHistoryValue.3 = 36182
cnpdThresholdConfigRising.3 = 30
cnpdThresholdHistoryProtocol.3 = 61
cnpdThresholdHistoryTime.3 = 44288471

Received SNMPv1 Trap:
Community: public
Enterprise: ciscoNbarProtocolDiscoveryMIB
Agent-addr: a.b.c.d
Enterprise Specific trap.
Enterprise Specific trap: 1
Time Ticks: 44288471
cnpdThresholdConfigIfIndex.3 = 13
cnpdThresholdConfigStatsSelect.3 = packetCountIn(7)
cnpdThresholdHistoryValue.3 = 17941
cnpdThresholdConfigRising.3 = 30
cnpdThresholdHistoryProtocol.3 = 41
cnpdThresholdHistoryTime.3 = 44288471

Received SNMPv1 Trap:
Community: public
Enterprise: ciscoNbarProtocolDiscoveryMIB
Agent-addr: a.b.c.d
Enterprise Specific trap.
Enterprise Specific trap: 1
Time Ticks: 44288472
cnpdThresholdConfigIfIndex.3 = 13
cnpdThresholdConfigStatsSelect.3 = packetCountIn(7)
cnpdThresholdHistoryValue.3 = 17318
cnpdThresholdConfigRising.3 = 30
cnpdThresholdHistoryProtocol.3 = 59
cnpdThresholdHistoryTime.3 = 44288471

Received SNMPv1 Trap:
Community: public
Enterprise: ciscoNbarProtocolDiscoveryMIB
Agent-addr: a.b.c.d
Enterprise Specific trap.
Enterprise Specific trap: 1
Time Ticks: 44288472
cnpdThresholdConfigIfIndex.3 = 13
cnpdThresholdConfigStatsSelect.3 = packetCountIn(7)
cnpdThresholdHistoryValue.3 = 12197
cnpdThresholdConfigRising.3 = 30
cnpdThresholdHistoryProtocol.3 = 62
cnpdThresholdHistoryTime.3 = 44288471

Received SNMPv1 Trap:
Community: public
Enterprise: ciscoNbarProtocolDiscoveryMIB
Agent-addr: a.b.c.d
Enterprise Specific trap.
Enterprise Specific trap: 1
Time Ticks: 44288472
cnpdThresholdConfigIfIndex.3 = 13
cnpdThresholdConfigStatsSelect.3 = packetCountIn(7)
cnpdThresholdHistoryValue.3 = 9046
cnpdThresholdConfigRising.3 = 30
cnpdThresholdHistoryProtocol.3 = 68
cnpdThresholdHistoryTime.3 = 44288471

Received SNMPv1 Trap:
Community: public
Enterprise: ciscoNbarProtocolDiscoveryMIB
Agent-addr: a.b.c.d
Enterprise Specific trap.
Enterprise Specific trap: 2
Time Ticks: 44289471
cnpdThresholdConfigIfIndex.3 = 13
cnpdThresholdConfigStatsSelect.3 = packetCountIn(7)
cnpdThresholdHistoryValue.3 = 1
cnpdThresholdConfigFalling.3 = 20
cnpdThresholdHistoryProtocol.3 = 62
cnpdThresholdHistoryTime.3 = 44289471

Additional References

For additional information related to <module feature>, refer to the following references:

Related Documents

Related Topic	Document Title
Network-Based Application Recognition	Network-Based Application Recognition and Distributed Network-Based Application Recognition
SNMP	Configuring SNMP Support Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2

Standards

This feature does not address any new Standards.

For information on Standards supported by the general NBAR and dNBAR features, see the *Network-Based Application Recognition and Distributed Network-Based Application Recognition* document.

MIBs

This document documents the CISCO-NBAR-PROTOCOL-DISCOVERY MIB.

For information on other MIBs supported by the general NBAR and dNBAR features, see the *Network-Based Application Recognition and Distributed Network-Based Application Recognition* document.

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

This feature does not address any new RFCs.

For information on RFCs supported by the general NBAR and dNBAR features, see the *Network-Based Application Recognition and Distributed Network-Based Application Recognition* document.

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 modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at <http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm>.

- **snmp-server enable traps cnpd**
- **snmp-server host**



Part 2: 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)
- 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 chapter “[Classification Overview](#)” in this book 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 20 compares the salient features of flow-based WFQ, CBWFQ and DCBWFQ, CQ, and PQ.

Table 20 *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	<ul style="list-style-type: none"> • 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 	<ul style="list-style-type: none"> • 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 	<ul style="list-style-type: none"> • Round-robin service 	<ul style="list-style-type: none"> • 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 21](#) summarizes the differences among WFQ, DWFQ, CBWFQ, and DCBWFQ.

Table 21 *WFQ, DWFQ, CBWFQ, and DCBWFQ Comparison*

WFQ	DWFQ	CBWFQ	DCBWFQ
Flow-based WFQ: <ul style="list-style-type: none"> 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: <ul style="list-style-type: none"> FQ, not weighted Class-based DWFQ: <ul style="list-style-type: none"> Weighted QoS-group-based Type of Service (ToS)-based 	Class-based WFQ: <ul style="list-style-type: none"> Weighted Bandwidth allocation can be specified for a specific class of traffic 	Class-based distributed WFQ: <ul style="list-style-type: none"> 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)

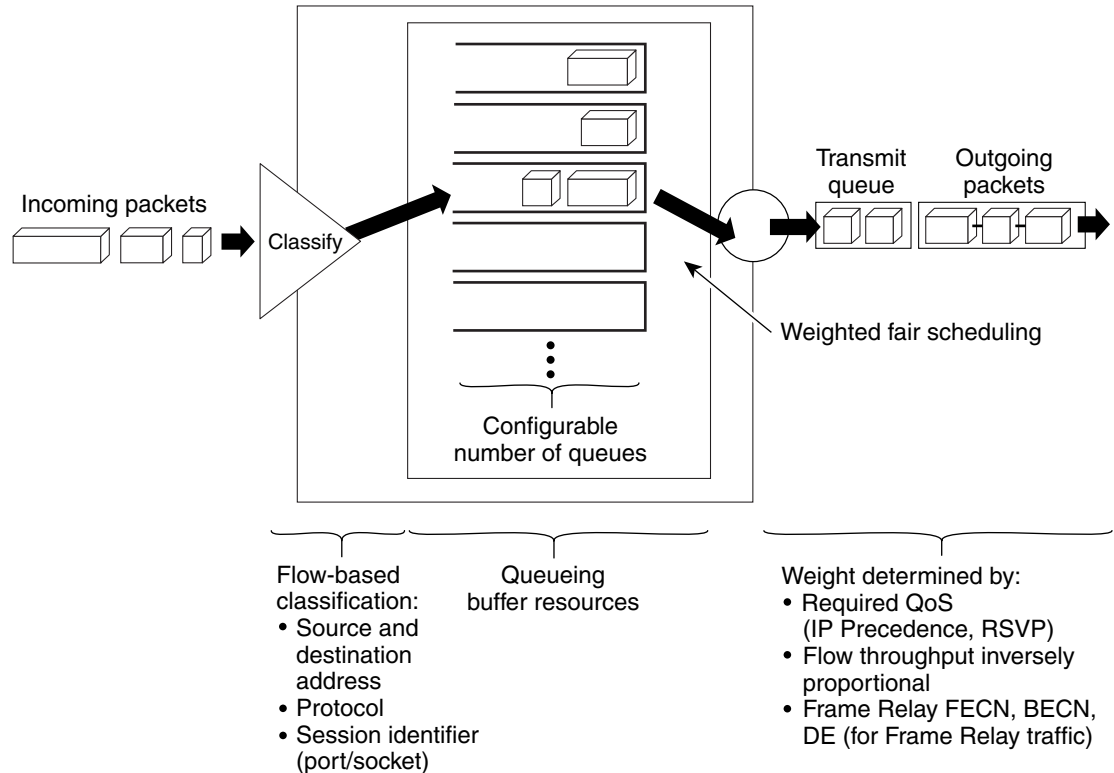
For DWFQ and DCBWFQ, all queueing is transacted by the VIP. On the VIP, all packets are sent directly out the interface. A Route Switch Processor (RSP) resides on the same platform as the VIP. The RSP handles all tasks associated with system maintenance and routing. The VIP and the RSP each handle some 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 chapter [“Configuring Weighted Fair Queueing”](#) in this book. For information on how to configure per-VC WFQ and CBWFQ, see the chapter [“Configuring IP to ATM Class of Service”](#) in this book.

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 8](#) shows how WFQ works.

Figure 8 Weighted Fair Queueing



16756

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.

WFQ and Frame Relay

WFQ weights are affected by Frame Relay discard eligible (DE), forward explicit congestion notification (FECN), and backward explicit congestion notification (BECN) bits when traffic is switched by the Frame Relay switching module. Once congestion is flagged, the weights used by the algorithm are altered so that the conversation encountering the congestion sends less frequently.

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, refer to the *Cisco IOS Switching Services Configuration Guide* and the *Cisco IOS Switching Services Command Reference*.



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 section [“Weighted Fair Queueing”](#) in this chapter.

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 chapter [“Configuring the Modular Quality of Service Command-Line Interface”](#) in this book.

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, enqueueing 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 chapter [“Configuring Weighted Fair Queueing”](#) in this book.

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.

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 chapter [“Configuring Weighted Fair Queueing”](#) in this book.

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 chapter [“Configuring the Modular Quality of Service Command-Line Interface”](#) in this book.

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 chapter [“Configuring Weighted Fair Queuing”](#) in this book.

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 queuing 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 chapter [“Configuring Weighted Fair Queuing”](#) in this book.

Frame Relay PVC Interface Priority Queuing

The Frame Relay PVC Interface Priority Queuing (PIPQ) feature provides an interface-level priority queuing 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 chapter [“Configuring Weighted Fair Queuing”](#) in this book. For information about Frame Relay, refer to the *Cisco IOS Wide-Area Networking Configuration Guide*.

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. The interface-level priority queuing takes the place of the FIFO queuing or dual FIFO queuing normally used by FRTS and FRF.12. 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 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 chapter “[Configuring Weighted Fair Queuing](#)” in this book.

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 chapters [“IP to ATM Class of Service Overview”](#) and [“Configuring IP to ATM Class of Service”](#) later in this book.

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. 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.

This feature also supports the *Class-Based Quality of Service* MIB.

For information on how to configure Distributed LLQ, see the chapter [“Configuring Weighted Fair Queueing”](#) in this book.

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 kbps. 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 percent** 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.

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 chapter “[Configuring Weighted Fair Queueing](#)” in this book.

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 chapter [“Configuring Custom Queueing”](#) in this book.

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.

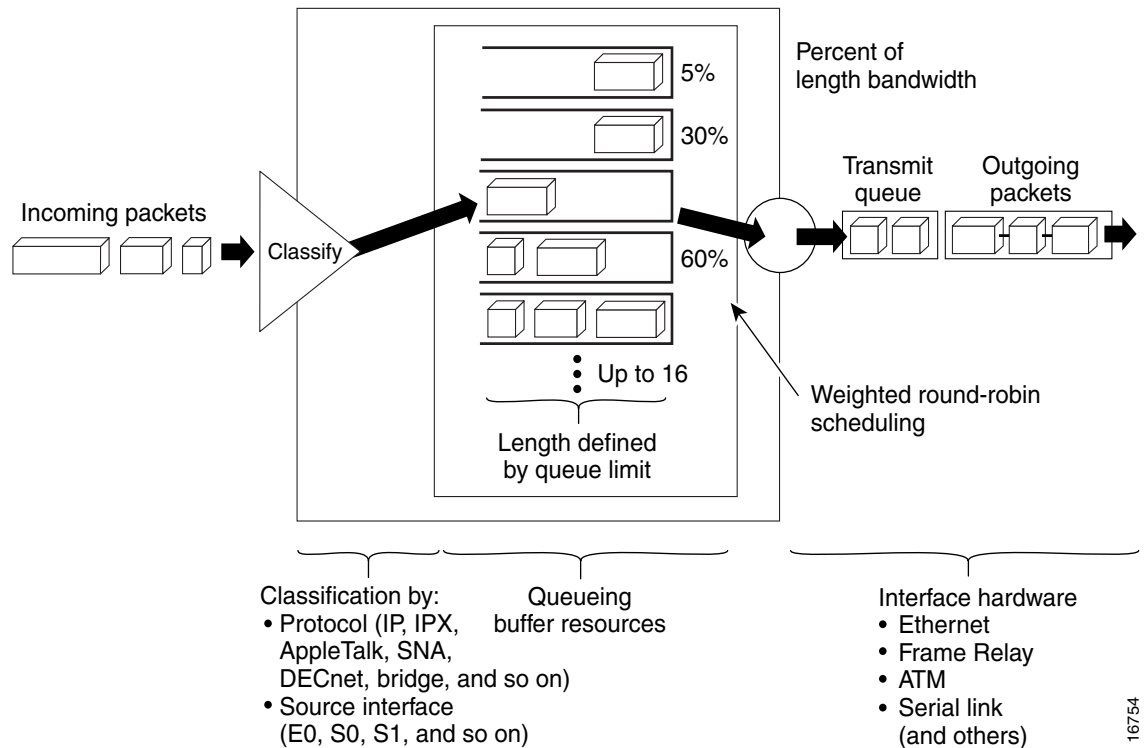
When CQ is enabled on an interface, the system maintains 17 output queues for that interface. You can specify queues 1 through 16. Associated with each output queue is a configurable byte count, which specifies how many bytes of data the system should deliver from the current queue before it moves on 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), dequeuing 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 being processed, 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 9 shows how CQ behaves.

Figure 9 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:
- $$1086/6240, 3492/6240, 1662/6240 = 17.4, 56, \text{ and } 26.6 \text{ percent}$$
- 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:

$$(\text{queue byte count} / \text{total byte count of all queues}) * \text{bandwidth capacity of the interface}$$

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

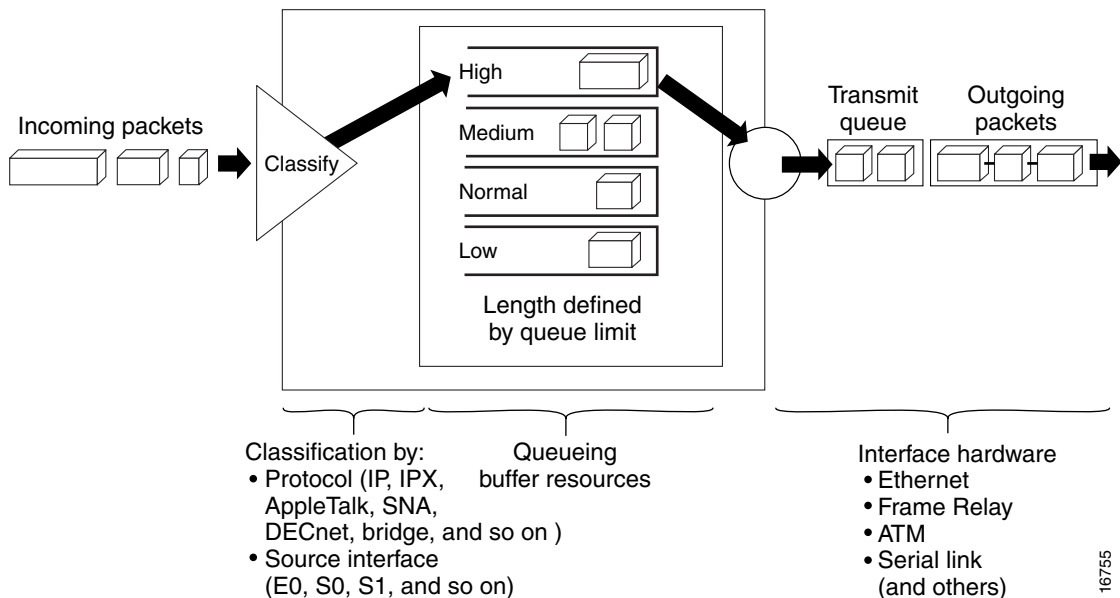
PQ allows you to define how traffic is prioritized in the network. You configure four traffic priorities. You can define a series of filters based on packet characteristics to cause the router to place traffic into these four queues; the queue with the highest priority is serviced first until it is empty, then the lower queues are serviced in sequence.

For information on how to configure PQ, see the chapter “Configuring Priority Queueing” in this book.

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 10](#) illustrates this process.

Figure 10 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.



Weighted Fair Queueing

This part consists of the following:

- [Configuring Weighted Fair Queueing](#)
- [Low Latency Queueing with Priority Percentage Support](#)
- [Low Latency Queueing \(LLQ\) for IPSec Encryption Engines](#)



Configuring Weighted Fair Queueing

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 section “Weighted Fair Queueing” in the chapter [“Congestion Management Overview”](#) in this book.

For a complete description of the QoS commands in this chapter, refer to 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.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the [“Finding Additional Feature Support Information”](#) section on page [lxix](#) in the [Using Cisco IOS Software for Release 12.4](#) chapter in this book.

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 [<i>congestive-discard-threshold</i> [<i>dynamic-queues</i> [<i>reservable-queues</i>]]]	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 (PQ) 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 [<i>interface</i>]	Displays statistical information specific to an interface.
Router# show queue <i>interface-type interface-number</i>	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](#)
- [Configuring QoS-Group-Based DWFQ](#)
- [Configuring Type of Service-Based DWFQ](#)
- [Monitoring DWFQ \(Optional\)](#)

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 <i>aggregate-packet</i>	(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 <i>individual-packet</i>	(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 <i>number</i> weight <i>weight</i>	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 <i>aggregate-packet</i>	(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 <i>individual-packet</i>	(Optional) Sets the maximum queue size for every per-flow queue during periods of congestion.
Step 5	Router(config-if)# fair-queue qos-group <i>number</i> limit <i>class-packet</i>	(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](#) (Required)
- [Configuring Class Policy in the Policy Map](#) (Required)
- [Attaching the Service Policy and Enabling CBWFQ](#) (Required)
- [Modifying the Bandwidth for an Existing Policy Map Class](#) (Optional)
- [Modifying the Queue Limit for an Existing Policy Map Class](#) (Optional)
- [Configuring the Bandwidth Limiting Factor](#) (Optional)
- [Deleting Classes](#) (Optional)

- [Deleting Policy Maps](#) (Optional)
- [Verifying Configuration of Policy Maps and Their Classes](#) (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 chapter [Configuring IP to ATM Class of Service](#) in this book.

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 <i>class-map-name</i>	Specifies the name of the class map to be created.
Step 2	Router (config-cmap) # match access-group { <i>access-group</i> name <i>access-group-name</i> }	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 <i>interface-name</i>	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 <i>protocol</i>	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 <i>number</i>	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 the section “[Creating a Traffic Class](#)” in the chapter [Configuring the Modular Quality of Service Command-Line Interface](#) in this book.

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](#) (Optional)
- [Configuring Class Policy Using WRED Packet Drop](#) (Optional)
- [Configuring the Class-Default Class Policy](#) (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 <i>policy-map</i>	Specifies the name of the policy map to be created or modified.
Step 2	Router(config-pmap)# class <i>class-name</i>	Specifies the name of a class to be created and included in the service policy.
Step 3	Router(config-pmap-c)# bandwidth { <i>bandwidth-kbps</i> percent <i>percent</i> }	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 <i>number-of-packets</i>	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 <i>policy-map</i>	Specifies the name of the policy map to be created or modified.
Step 2	Router(config-pmap)# class <i>class-name</i>	Specifies the name of a class to be created and included in the service policy.
Step 3	Router(config-pmap-c)# bandwidth { <i>bandwidth-kbps</i> percent <i>percent</i> }	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 <i>exponent</i> or Router(config-pmap-c)# random-detect precedence <i>precedence min-threshold max-threshold mark-prob-denominator</i>	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 <i>policy-map</i>	Specifies the name of the policy map to be created or modified.
Step 2	Router(config-pmap)# class class-default <i>default-class-name</i>	Specifies the default class so that you can configure or modify its policy.
Step 3	Router(config-pmap-c)# bandwidth { <i>bandwidth-kbps</i> percent <i>percent</i> }	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.
	or	
	Router(config-pmap-c)# fair-queue [<i>number-of-dynamic-queues</i>]	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 <i>Cisco IOS Quality of Service Solutions Command Reference</i> 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 <i>number-of-packets</i>	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 <i>policy-map</i>	Specifies the name of the policy map to be created or modified.
Step 2	Router(config-pmap)# class class-default <i>default-class-name</i>	Specifies the default class so that you can configure or modify its policy.

Class-Based Weighted Fair Queueing Configuration Task List

	Command	Purpose
Step 3	<pre>Router(config-pmap-c)# bandwidth {<i>bandwidth-kbps</i> percent percent}</pre> <p>or</p> <pre>Router(config-pmap-c)# fair-queue [<i>number-of-dynamic-queues</i>]</pre>	<p>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.</p> <p>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 <i>Cisco IOS Quality of Service Solutions Command Reference</i> for the default number of dynamic queues that WFQ and CBWFQ use when they are enabled on an interface or ATM VC.</p>
Step 4	<pre>Router(config-pmap-c)# random-detect</pre>	Enables WRED. The class policy will drop packets using WRED instead of tail drop.
Step 5	<pre>Router(config-pmap-c)# random-detect exponential-weighting-constant <i>exponent</i></pre> <p>or</p> <pre>Router(config-pmap-c)# random-detect precedence <i>precedence min-threshold max-threshold</i> <i>mark-prob-denominator</i></pre>	<p>Configures the exponential weight factor used in calculating the average queue length.</p> <p>Configures WRED parameters for packets with a specific IP precedence. Repeat this command for each precedence.</p>

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
<pre>Router(config-if)# service-policy output <i>policy-map</i></pre>	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 <i>policy-map</i>	Specifies the name of the policy map containing the class to be modified.
Step 2	Router(config-pmap)# class <i>class-name</i>	Specifies the name of a class whose bandwidth you want to modify.
Step 3	Router(config-pmap-c)# bandwidth { <i>bandwidth-kbps</i> percent <i>percent</i> }	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 <i>policy-map</i>	Specifies the name of the policy map containing the class to be modified.
Step 2	Router(config-pmap)# class <i>class-name</i>	Specifies the name of a class whose queue limit you want to modify.
Step 3	Router(config-pmap-c)# queue-limit <i>number-of-packets</i>	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 <i>percent</i>	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 <i>policy-map</i>	Specifies the name of the policy map containing the classes to be deleted.
Step 2	Router(config-pmap)# no class <i>class-name</i>	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 <i>policy-map</i>	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 <i>policy-map</i>	Displays the configuration of all classes that make up the specified policy map.
Router# show policy-map <i>policy-map</i> class <i>class-name</i>	Displays the configuration of the specified class of the specified policy map.
Router# show policy-map interface <i>interface-name</i>	Displays the configuration of all classes configured for all policy maps on the specified interface.
Router# show queue <i>interface-type</i> <i>interface-number</i>	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](#) (Optional)
- [Modifying the Queue Limit for an Existing Traffic Class](#) (Optional)
- [Monitoring and Maintaining DCBWFQ](#) (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 chapter [Configuring the Modular Quality of Service Command-Line Interface](#) in this book.

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 <i>policy-map</i>	Specifies the name of the traffic policy to be created or modified.
Step 2	Router(config-pmap)# class <i>class-name</i>	Specifies the name of a traffic class whose bandwidth you want to modify.
Step 3	Router(config-pmap-c)# bandwidth <i>bandwidth-kbps</i>	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 chapter [Configuring the Modular Quality of Service Command-Line Interface](#) in this book.

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 <i>policy-map</i>	Specifies the name of the traffic policy to be created or modified.
Step 2	Router(config-pmap)# class <i>class-name</i>	Specifies the name of a traffic class whose queue limit you want to modify.
Step 3	Router(config-pmap-c)# queue-limit <i>number-of-packets</i>	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 chapter [Configuring the Modular Quality of Service Command-Line Interface](#) in this book.

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 <i>policy-map-name</i>	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 <i>interface-spec</i>	Displays configuration and statistics of the input and output policies attached to a particular interface.
Router# show policy-map interface <i>interface-spec</i> <i>input</i>	Displays configuration and statistics of the input policy attached to an interface.
Router# show policy-map interface <i>interface-spec</i> <i>output</i>	Displays configuration statistics of the output policy attached to an interface.
Router# show policy-map [interface [<i>interface-spec</i> [<i>input</i> <i>output</i>] [class <i>class-name</i>]]]	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](#) (Required)
- [Configuring the Bandwidth Limiting Factor](#) (Optional)
- [Verifying IP RTP Priority](#) (Optional)
- [Monitoring and Maintaining IP RTP Priority](#) (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) must be configured before the Frame Relay IP RTP Priority feature is used. For information about configuring FRTS and FRF.12, refer to the *Cisco IOS Wide-Area Networking Configuration Guide*.

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 <i>starting-rtp-port-number port-number-range</i> <i>bandwidth</i>	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, refer to 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 <i>percent</i>	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 <i>interface-type interface-number</i>	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 <i>interface-type interface-number</i>	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](#) (Required)
- [Verifying Frame Relay IP RTP Priority](#) (Optional)
- [Monitoring and Maintaining Frame Relay IP RTP Priority](#) (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 <i>starting-rtp-port-number port-number-range bandwidth</i>	Reserves a strict priority queue for a set of RTP 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 <i>interface-type interface-number</i>	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](#) (Required)
- [Enabling Frame Relay PIPQ and Setting Queue Limits](#) (Required)
- [Assigning a Map Class to a PVC](#) (Required)
- [Verifying Frame Relay PIPQ](#) (Optional)
- [Monitoring and Maintaining Frame Relay PIPQ](#) (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 <i>map-class-name</i>	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 <i>type number</i> [<i>name-tag</i>]	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 [<i>high-limit medium-limit normal-limit low-limit</i>]	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 <i>dlci</i>	Specifies a single PVC on a Frame Relay interface.
Step 2	Router(config-fr-dlci)# class <i>map-class-name</i>	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 <i>interface</i>][<i>dlci</i>]	Displays statistics about PVCs for Frame Relay interfaces.
Router# show interfaces [<i>type number</i>][<i>first</i>][<i>last</i>]	Displays the statistical information specific to a serial interface.
Router# show queueing [custom fair priority random-detect [interface <i>atm_subinterface</i> [vc [[<i>vpi</i> /] <i>vci</i>]]]]]	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 <i>interface</i>][<i>dlci</i>]	Displays statistics about PVCs for Frame Relay interfaces.
Router# show interfaces [<i>type number</i>][<i>first</i>][<i>last</i>]	Displays the statistical information specific to a serial interface.
Router# show queue <i>interface-name interface-number</i> [vc [<i>vpi</i> /] <i>vci</i>][<i>queue-number</i>]	Displays the contents of packets inside a queue for a particular interface or VC.
Router# show queueing [custom fair priority random-detect [interface <i>atm_subinterface</i> [vc [[<i>vpi</i> /] <i>vci</i>]]]]]	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](#) (Required)
- [Configuring the Bandwidth Limiting Factor](#) (Optional)
- [Verifying LLQ](#) (Optional)
- [Monitoring and Maintaining LLQ](#) (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 <i>bandwidth</i>	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 <i>percent</i>	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 <i>interface-type interface-number</i>	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](#) (Required)
- [Configuring a Priority Queue for a Percentage of Available Bandwidth](#) (Required)
- [Configuring a Transmission Ring Limit](#) (Optional)
- [Verifying Distributed LLQ](#) (Optional)
- [Verifying a Transmission Ring Limit](#) (Optional)
- [Monitoring and Maintaining Distributed LLQ](#) (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 <i>policy-name</i>	Specifies the name of the policy map to configure. Enters policy-map configuration mode.
Step 2	Router(config-pmap)# class <i>class-name</i>	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 <i>kpbs</i> [<i>bytes</i>]	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 chapter [Configuring the Modular Quality of Service Command-Line Interface](#) in this book.

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 <i>policy-name</i>	Specifies the name of the traffic policy to configure. Enters policy-map configuration mode.
Step 2	Router(config-pmap)# class <i>class-name</i>	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 <i>percent</i>	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 chapter [Configuring the Modular Quality of Service Command-Line Interface](#) in this book.

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 <i>interface-name</i>	Specifies the name of the ATM interface to configure.
Step 2	Router(config-if)# atm pvc <i>vcd-number vpi-number vci-number Encapsulation-type tx-ring-limit ring-limit</i>	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 <i>subinterface name</i>	Specifies the name of the subinterface to configure.
Step 2	Router(config-subif)# pvc <i>pvc-name</i>	Specifies the name of the PVC to configure.
Step 3	Router(config-if-atm-vc)# tx-ring-limit <i>ring-limit</i>	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 [<i>interface-type interface-number</i>] fair-queue	Displays information and statistics about WFQ for a VIP-based interface.
Router# show policy-map <i>policy-map-name</i>	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 <i>vc-name</i>	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 [<i>interface-type interface-number</i>] fair-queue	Displays information and statistics about WFQ for a VIP-based interface.
Router# show policy-map <i>policy-map-name</i>	Displays the contents of a traffic policy, including the priority setting in a specific policy map.
Router# show policy interface <i>interface-name</i>	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 <i>vc-name</i>	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](#) (Required)
- [Configuring Class Policy in the Policy Map](#) (Required)
- [Attaching the Service Policy and Enabling LLQ for Frame Relay](#) (Required)
- [Verifying Configuration of Policy Maps and Their Classes](#) (Optional)
- [Monitoring and Maintaining LLQ for Frame Relay](#) (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	<code>Router(config)# class-map <i>class-map-name</i></code>	Specifies the name of the class map to be created.
Step 2	<code>Router(config-cmap)# match access-group {<i>access-group</i> <i>name access-group-name</i>}</code>	Specifies the name of the ACL against whose contents packets are checked to determine if they belong to the class.
	or	
	<code>Router(config-cmap)# match input-interface <i>interface-name</i></code>	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	
	<code>Router(config-cmap)# match protocol <i>protocol</i></code>	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](#) (Required)
- [Configuring Class Policy Using a Specified Bandwidth and WRED Packet Drop](#) (Optional)
- [Configuring the Class-Default Class Policy](#) (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 <i>policy-map</i>	Specifies the name of the policy map to be created or modified.
Step 2	Router(config-pmap)# class <i>class-name</i>	Specifies the name of a class to be created and included in the service policy.
Step 3	Router(config-pmap-c)# priority <i>bandwidth-kbps</i>	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 <i>policy-map</i>	Specifies the name of the policy map to be created or modified.
Step 2	Router(config-pmap)# class <i>class-name</i>	Specifies the name of a class to be created and included in the service policy.

	Command	Purpose
Step 3	Router(config-pmap-c)# bandwidth <i>bandwidth-kbps</i>	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 Steps 2 through 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 <i>policy-map</i>	Specifies the name of the policy map to be created or modified.
Step 2	Router(config-pmap)# class class-default <i>default-class-name</i>	Specifies the default class so that you can configure or modify its policy.
Step 3	Router(config-pmap-c)# bandwidth <i>bandwidth-kbps</i> or Router(config-pmap-c)# fair-queue [<i>number-of-dynamic-queues</i>]	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 <i>number-of-packets</i>	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 queuing system.

Command	Purpose
Router(config-map-class)# service-policy output <i>policy-map</i>	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 <i>dlci</i>	Displays statistics about the PVC and the configuration of classes for the policy map on the specified DLCI.
Router# show policy-map interface <i>interface-name</i>	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 <i>interface-name dlci</i> <i>dlci</i>	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](#) (Required)
- [Configuring the LLQ Burst Size](#) (Required)
- [Verifying the LLQ Burst Size](#) (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 global configuration mode:

Command	Purpose
Router(config)# priority <i>bandwidth</i>	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 global configuration mode:

Command	Purpose
Router(config)# priority <i>bandwidth burst</i>	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](#) (Required)
- [Verifying the Configuration of the per-VC Hold Queue on an ATM Adapter](#) (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 chapter [IP to ATM Class of Service Overview](#) and later in this book.

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 <i>number-of-packets</i>	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:

- [Flow-Based DWFQ Example](#)
- [QoS-Group-Based DWFQ Example](#)
- [ToS-Based DWFQ Example](#)

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 DWFQ 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](#)
- [Policy Creation Example](#)
- [Policy Attachment to Interfaces Example](#)
- [CBWFQ Using WRED Packet Drop Example](#)
- [Display Service Policy Map Content Examples](#)

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 serial10/0
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:

- [Traffic Class Configuration Example](#)
- [Traffic Policy Creation Example](#)
- [Traffic Policy Attachment to an Interface Example](#)

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 chapter [Configuring the Modular Quality of Service Command-Line Interface](#) in this book.

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 chapter [Configuring the Modular Quality of Service Command-Line Interface](#) in this book.

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 chapter [Configuring the Modular Quality of Service Command-Line Interface](#) in this book.

IP RTP Priority Configuration Examples

The following sections provide IP RTP Priority configuration examples:

- [CBWFQ Configuration Example](#)
- [Virtual Template Configuration Example](#)
- [Multilink Bundle Configuration Example](#)
- [Debug Example](#)

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 for configuring 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 queuing, 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](#)
- [Virtual Template Configuration Example](#)
- [Multilink Bundle Configuration Example](#)

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](#)
- [Enabling PQ for a Percentage of Available Bandwidth on an ATM Subinterface Example](#)
- [Limiting the Transmission Ring Limit on an ATM Interface Example](#)
- [Limiting the Transmission Ring Limit on an ATM PVC Subinterface Example](#)

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 kbps 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-c)# class bar
Router(config-pmap-c)# bandwidth percent 20
Router(config-pmap-c)# 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
```



Low Latency Queueing with Priority Percentage Support

This feature module describes the Low Latency Queueing with Priority Percentage Support feature and includes the following sections:

- [Feature Overview, page 273](#)
- [Supported Platforms, page 275](#)
- [Supported Standards, MIBs, and RFCs, page 276](#)
- [Configuration Tasks, page 276](#)
- [Configuration Examples, page 277](#)
- [Command Reference, page 278](#)

Feature Overview

This feature allows you to configure bandwidth as a percentage within low latency queueing (LLQ). Specifically, you can designate a percentage of the bandwidth to be allocated to an entity (such as a physical interface, a shaped ATM permanent virtual circuit (PVC), or a shaped Frame Relay PVC) to which a policy map is attached. Traffic associated with the policy map will then be given priority treatment.

This feature also allows you to specify the percentage of bandwidth to be allocated to non-priority traffic classes.

This feature modifies two existing commands—**bandwidth** and **priority**—and this feature provides additional functionality to the way that bandwidth can be allocated using these two commands.

Changes to the bandwidth Command

This feature adds a new keyword to the **bandwidth** command—**remaining percent**. The feature also changes the functionality of the existing **percent** keyword. These changes result in the following commands for bandwidth: **bandwidth percent** and **bandwidth remaining percent**.

The **bandwidth percent** command configures bandwidth as an absolute percentage of the total bandwidth on the interface.

The **bandwidth remaining percent** command allows you to allocate bandwidth as a relative percentage of the total bandwidth available on the interface. This command allows you to specify the relative percentage of the bandwidth to be allocated to the classes of traffic. For instance, you can specify that 30 percent of the available bandwidth be allocated to class1, and 60 percent of the bandwidth be allocated to class2. Essentially, you are specifying the ratio of the bandwidth to be allocated to the traffic class. In this case, the ratio is 1 to 2 (30 percent allocated to class1 and 60 percent allocated to class2). The sum of the numbers used to indicate this ratio cannot exceed 100 percent. This way, you need not know the total amount of bandwidth available, just the relative percentage you want to allocate for each traffic class.

Each traffic class gets a minimum bandwidth as a relative percentage of the remaining bandwidth. The remaining bandwidth is the bandwidth available after the priority queue, if present, is given its required bandwidth, and after any Resource Reservation Protocol (RSVP) flows are given their requested bandwidth.

Because this is a relative bandwidth allocation, the packets for the traffic classes are given a proportionate weight only, and no admission control is performed to determine whether any bandwidth (in kbps) is actually available. The only error checking that is performed is to ensure that the total bandwidth percentages for the classes do not exceed 100 percent.

For more information about how this feature defines and calculates bandwidth, see the [“How These Commands Calculate Bandwidth”](#) section of this document. For the **bandwidth command** syntax description and usage guidelines, see the [“Command Reference”](#) section of this document.

Changes to the priority Command

This feature also adds the **percent** keyword to the **priority** command. The **priority percent** command indicates that the bandwidth will be allocated as a percentage of the total bandwidth of the interface. You can then specify the percentage (that is, a number from 1 to 100) to be allocated by using the *percentage* argument with the **priority percent** command.

Unlike the **bandwidth** command, the **priority** command provides a strict priority to the traffic class, which ensures low latency to high priority traffic classes.

For more information about how this feature defines and calculates bandwidth, see the [“How These Commands Calculate Bandwidth”](#) section of this document. For the **priority command** syntax description and usage guidelines, see the [“Command Reference”](#) section of this document.

How These Commands Calculate Bandwidth

When the **bandwidth** and **priority** commands calculate the total amount of bandwidth available on an entity, the following guidelines are invoked:

- If the entity is a physical interface, the total bandwidth is the bandwidth on the physical interface.
- If the entity is a shaped ATM PVC, the total bandwidth is calculated as follows:
 - For a variable bit rate (VBR) VC, the average shaping rate is used in the calculation.
 - For an available bit rate (ABR) VC, the minimum shaping rate is used in the calculation.
- If the entity is a shaped Frame Relay PVC, the total bandwidth is calculated as follows:
 - If a minimum acceptable committed information rate (minCIR) is not configured, the CIR divided by two is used in the calculation.
 - If a minimum acceptable CIR is configured, the minCIR setting is used in the calculation.

For more information on bandwidth allocation, refer to the chapter “Congestion Management Overview” in the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2.

Benefits

This feature allows the Cisco IOS software to accommodate networks with a large number of interfaces, all with differing bandwidths. This feature is especially useful when all of those interfaces with differing bandwidths need to be associated with a policy map that allocates proportional bandwidths to multiple classes.

Additionally, configuring bandwidth in percentages is most useful when the underlying link bandwidth is unknown or the relative class bandwidth distributions are known. For interfaces that have adaptive shaping rates (such as available bit rate (ABR) virtual circuits), CBWFQ can be configured by configuring class bandwidths in percentages.

Restrictions

Dropping Excess Traffic

If the incoming high priority traffic exceeds the bandwidth percentage calculated by the **priority percent** command, and there is congestion in the network, the excess traffic is dropped. This is identical to the behavior demonstrated when the **priority** command uses bandwidth in kbps. In both cases, if the high priority traffic exceeds the bandwidth, and there is congestion in the network, excess traffic is dropped.

Exceeding the Configured Bandwidth Percentage Calculated by the **bandwidth percent** and **priority percent** Commands

By default, when the **bandwidth percent** and **priority percent** commands are used to allocate bandwidth, the sum of the bandwidth percentage allocated to the high priority traffic and the bandwidth percentage allocated to the non-priority traffic cannot exceed 75 percent of the total bandwidth available on the interface.

The remaining 25 percent of the total bandwidth available on the interface is kept in reserve for the unclassified traffic and routing traffic, if any, and proportionally divided among the defined traffic classes. To override the 75 percent limitation, use the **max-reserved bandwidth** command in interface configuration mode.



Note

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).

Supported Platforms

- Cisco 1000 series
- Cisco 1600 series
- Cisco 1700 series
- Cisco 2600 series
- Cisco 3600 series
- Cisco MC3810

- Cisco 4500 series
- Cisco 5300 series
- Cisco 7100 series
- Cisco 7200 series

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

No new or modified MIBs are supported by this feature.

To obtain lists of supported MIBs 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

No new or modified RFCs are supported by this feature.

Configuration Tasks

See the following sections for configuration tasks for the Low Latency Queueing with Priority Percentage Support feature. Each task in the list is identified as either optional or required:

- [Specifying the Bandwidth Percentage](#) (Required)
- [Verifying the Bandwidth Percentage](#) (Optional)

Specifying the Bandwidth Percentage

To specify the bandwidth percentage, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config-if)# policy-map <i>policy-map</i>	Specifies the name of the policy map to be created or modified. Enters policy-map configuration mode.
Step 2	Router(config-pmap)# class <i>class-name</i>	Specifies the class so that you can configure or modify its policy. Enters policy-map class configuration mode.
Step 3	Router(config-pmap-c)# priority { <i>bandwidth-kbps</i> percent <i>percentage</i> } [<i>burst</i>]	Gives priority to a class of traffic belonging to the policy map.
Step 4	Router(config-pmap-c)# bandwidth { <i>bandwidth-kbps</i> remaining percent <i>percentage</i> percent <i>percentage</i> }	Specifies the bandwidth for a class of traffic belonging to the policy map.

Verifying the Bandwidth Percentage

To display the contents of a policy map and to verify that the percentage has been configured as specified, use the following commands in EXEC mode, as needed:

Command	Purpose
Router# <code>show policy-map policy-map</code>	Displays the configuration of all classes comprising the specified policy map.
Router# <code>show policy-map policy-map class class-name</code>	Displays the configuration of the specified class of the specified policy map.
Router# <code>show policy-map interface interface-name</code>	Displays the configuration of all classes configured for all policy maps on the specified interface.



Note

The counters displayed for classes configured with **bandwidth** or **priority** after using the **show policy-map interface** command are updated only if congestion is present on the interface.

Configuration Examples

This section provides the following configuration examples:

- [Specifying the Bandwidth Percentage](#)
- [Mixing the Units of Bandwidth for Non-Priority Traffic](#)

Specifying the Bandwidth Percentage

The following example uses the **priority percent** command to specify a bandwidth percentage of 10 percent for the class called voice-percent. Then the **bandwidth remaining percent** command is used to specify a bandwidth percentage of 30 percent for the class called data1, and a bandwidth percentage of 20 percent for the class called data2:

```
policy-map policy1
  class voice-percent
    priority percent 10
  class data1
    bandwidth remaining percent 30
  class data2
    bandwidth remaining percent 20
```

As a result of this configuration, 10 percent of the interface bandwidth is guaranteed for the class called voice-percent. The classes called data1 and data2 get 30 percent and 20 percent of the remaining bandwidth, respectively.

Mixing the Units of Bandwidth for Non-Priority Traffic

If a particular unit (that is, kbps or percentages) is used when specifying the bandwidth for a specific class of non-priority traffic, the same bandwidth unit must be used when specifying the bandwidth for the other non-priority classes in that policy map. The bandwidth units within the same policy map must be identical. However, the unit for the **priority** command in the priority class can be different from the bandwidth unit of the non-priority class. The same configuration can contain multiple policy maps, however, which in turn can use different bandwidth units.

The following sample configuration contains three policy maps—policy1, policy2, and policy3. In the policy map called policy1 and the policy map called policy2, the bandwidth is specified by percentage. However, in the policy map called policy3, bandwidth is specified in kbps.

```
policy-map policy1
  class voice-percent
    priority percent 10
  class data1
    bandwidth percent 30
  class data2
    bandwidth percent 20
policy-map policy2
  class voice-percent
    priority percent 10
  class data1
    bandwidth remaining percent 30
  class data2
    bandwidth remaining percent 20
policy-map policy3
  class voice-percent
    priority percent 500
  class data1
    bandwidth 30
  class data2
    bandwidth 20
```

Command Reference

The following modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at <http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm>.

- **bandwidth (policy-map class)**
- **priority**



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:

- [Feature Overview, page 279](#)
- [Supported Platforms, page 281](#)
- [Supported Standards, MIBs, and RFCs, page 282](#)
- [Prerequisites, page 282](#)
- [Configuration Tasks, page 283](#)
- [Monitoring and Maintaining LLQ for IPSec Encryption Engines, page 286](#)
- [Configuration Examples, page 286](#)
- [Command Reference, page 287](#)
- [Glossary, page 287](#)

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*, Cisco IOS Release 12.2
- *Class-Based Weighted Fair Queuing* feature module, Cisco IOS Release 12.1
- *IP RTP Priority* feature module, Cisco IOS Release 12.0

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, refer to 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 [Quality of Service Solutions Command Reference](#), Cisco IOS Release 12.2, to learn more about configuring server policies 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}	Specifies the name of the access control list (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 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 <i>default-class-name</i>	Specifies the default class so that you can configure or modify its policy.
Step 3	Router(config-pmap-c)# bandwidth bandwidth-kbps	Specifies the amount of bandwidth, in kbps, to be assigned to the class.
	or Router(config-pmap-c)# fair-queue [<i>number-of-dynamic-queues</i>]	

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).

	Command	Purpose
Step 2	Router# show policy-map interface <i>interface-name</i>	When LLQ is configured, displays the configuration of classes for all policy maps.
Step 3	Router# show policy-map interface <i>interface-name dlci dlci</i>	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 fas0/0
```

```
Router(config-if)# service-policy output policy1
```

Command Reference

The following command is pertinent to this feature. To see the command pages for this command and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at <http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm>.

- **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 [“Low Latency Queueing”](#) section on page 209 in the chapter [Congestion Management Overview](#) in this book.

For a complete description of the CQ commands in this chapter, refer to 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.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the [“Finding Additional Feature Support Information”](#) section on page lxix in the [Using Cisco IOS Software for Release 12.4](#) chapter in this book.



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:

	Command	Purpose
Step 1	Router(config)# interface <i>interface-type interface-number</i>	Specifies the interface, and then enters interface configuration mode.
Step 2	Router(config-if)# custom-queue-list <i>list</i>	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 <i>list-number queue queue-number limit limit-number</i>	Specifies the maximum number of packets allowed in each of the custom queues. The <i>limit-number</i> 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 <i>list-number queue queue-number byte-count byte-count-number</i>	Designates the average number of bytes forwarded per queue. The <i>byte-count-number</i> 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 <i>list-number</i> protocol <i>protocol-name</i> <i>queue-number</i> <i>queue-keyword</i> <i>keyword-value</i>	Establishes queueing priorities based on the protocol type.
Router(config)# queue-list <i>list-number</i> interface <i>interface-type</i> <i>interface-number</i> <i>queue-number</i>	Establishes CQ based on packets entering from a given interface.
Router(config)# queue-list <i>list-number</i> default <i>queue-number</i>	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 <i>interface-type</i> <i>interface-number</i>	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 <i>interface-type</i> <i>interface-number</i>	Displays the current status of the custom output queues when CQ is enabled.

Custom Queuing Configuration Examples

The following sections provide custom queuing 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 Queuing 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

This chapter describes the tasks for configuring priority queueing (PQ) on a router.

For complete conceptual information, see the [“Priority Queueing” section on page 222](#) in the chapter [Congestion Management Overview](#) in this book.

For a complete description of the PQ commands in this chapter, refer to 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.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the [“Finding Additional Feature Support Information” section on page lxix](#) in the [Using Cisco IOS Software for Release 12.4](#) chapter in this book.

Priority Queueing Configuration Task List

To configure PQ, perform the tasks described in the following sections. The tasks in the 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 set of priority queues. The priority list specifies which queue a packet will be placed in and, optionally, the maximum length of the different queues.

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](#) (Required)
- [Specifying the Maximum Size of the Priority Queues](#) (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 <i>list-number</i> protocol <i>protocol-name</i> { high medium normal low } <i>queue-keyword</i> <i>keyword-value</i>	Establishes queueing priorities based on the protocol type.
Step 2	Router(config)# priority-list <i>list-number</i> interface <i>interface-type</i> <i>interface-number</i> { high medium normal low }	Establishes queueing priorities for packets entering from a given interface.
Step 3	Router(config)# priority-list <i>list-number</i> 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 <i>list-number</i> queue-limit [<i>high-limit</i> [<i>medium-limit</i> [<i>normal-limit</i> [<i>low-limit</i>]]]]	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 22](#).

Table 22 Default Priority Queue Packet Limits

Priority Queue Argument	Packet Limits
<i>high-limit</i>	20
<i>medium-limit</i>	40
<i>normal-limit</i>	60
<i>low-limit</i>	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 <i>interface-type</i> <i>interface-number</i>	Specifies the interface, and then enters interface configuration mode.
Step 2	Router(config-if)# priority-group <i>list-number</i>	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 <i>interface-type interface-number</i>	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](#)
- [Priority Queueing Based on Interface Example](#)
- [Maximum Specified Size of the Priority Queue Example](#)
- [Priority List Assigned to an Interface Example](#)
- [Priority Queueing Using Multiple Rules Example](#)

For information on how to configure PQ, see the section “[Priority Queueing Configuration Task List](#)” in this chapter.

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
```




Part 3: 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 Hop Behavior (PHB). This feature enables customers to implement 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 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.

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

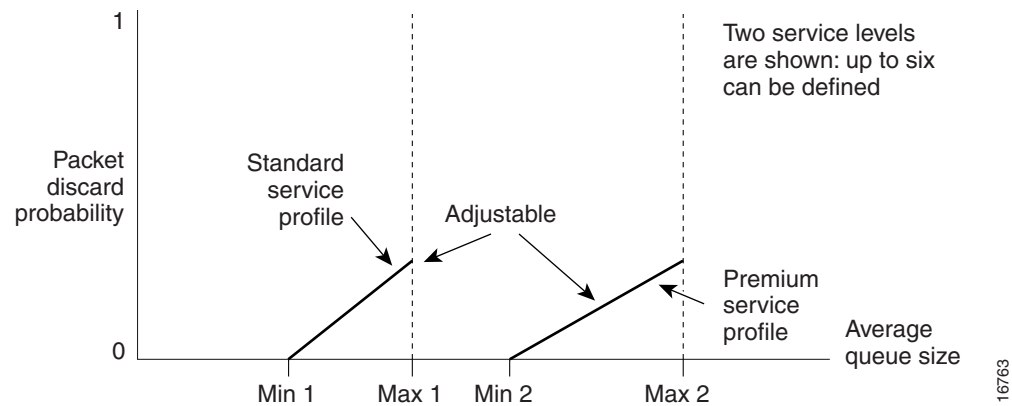
The packet drop probability is based on the minimum threshold, maximum threshold, and mark probability denominator.

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 11](#) summarizes the packet drop probability.

Figure 11 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 retransmit the first ACK, and that a second data segment was received out of order, forcing the receiver to retransmit 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 it 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.

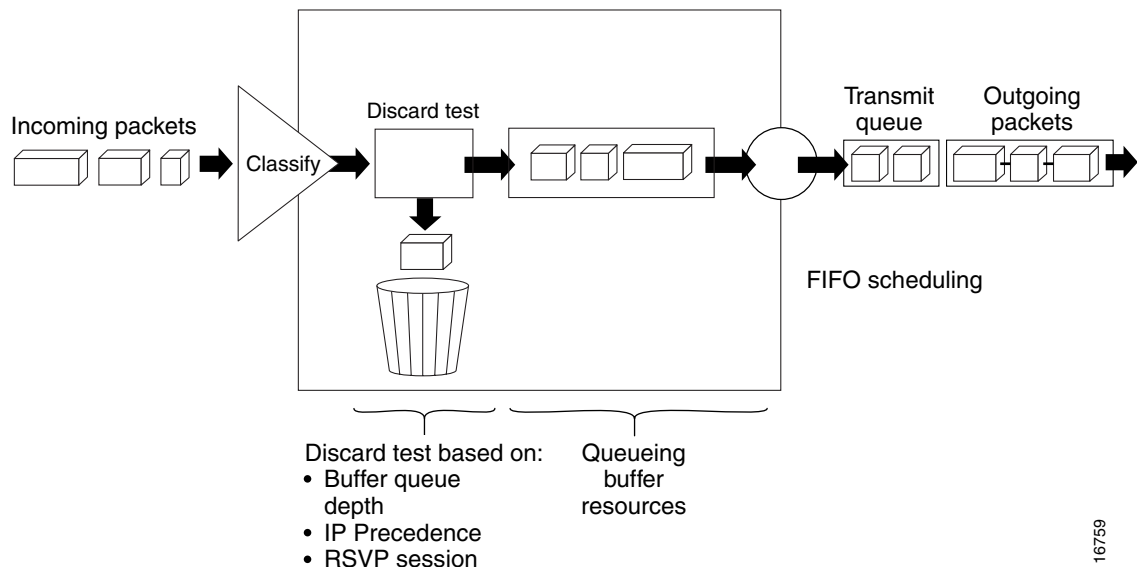
WRED avoids the globalization problems that occur when tail drop is used as the congestion avoidance mechanism. Global synchronization manifests when multiple TCP hosts reduce their transmission rates in response to packet dropping, then increase their transmission rates once again when the congestion 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 12 illustrates how WRED works.

Figure 12 *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:

$$\text{average} = (\text{old_average} * (1 - 2^{-n})) + (\text{current_queue_size} * 2^{-n})$$

where n is the exponential weight factor, a user-configurable 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, refer to the *Cisco IOS Switching Services Configuration Guide* and the *Cisco IOS Switching Services Command Reference*.

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 chapter [Configuring the Modular Quality of Service Command-Line Interface](#) of this book.

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:

$$\text{average} = (\text{old_average} * (1 - 1/2^n)) + (\text{current_queue_size} * 1/2^n)$$

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.

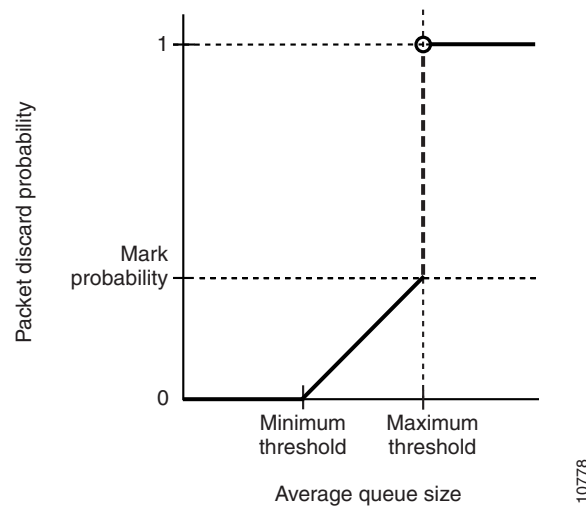
Packet-Drop Probability

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 13](#) summarizes the packet drop probability.

Figure 13 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 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, refer to the *Cisco IOS Switching Services Configuration Guide*.

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 you use 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, refer to the *Cisco IOS Quality of Service Command Reference*.



Weighted Random Early Detection

This part consists of the following:

- [Configuring Weighted Random Early Detection](#)
- [WRED — Explicit Congestion Notification](#)



Configuring Weighted Random Early Detection

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 [“Weighted Random Early Detection” section on page 302](#) in the chapter [Congestion Avoidance Overview](#) in this book.

For a complete description of the WRED and DWRED commands in this chapter, refer to 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 [IP to ATM Class of Service](#) in this book.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the [“Finding Additional Feature Support Information” section on page lxix](#) in the [Using Cisco IOS Software for Release 12.4](#) chapter in this book.



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.

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 “[About WRED](#)” section on page 305 in the chapter [Congestion Avoidance Overview](#) 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 <i>exponent</i>	Configures the weight factor used in calculating the average queue length.
Router(config-if)# random-detect precedence <i>precedence min-threshold max-threshold mark-prob-denominator</i>	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 <i>interface-type interface-number</i>	Displays the header information of the packets inside a queue. This command does not support DWRED.
Router# show queueing interface <i>interface-number [vc [[vpi/] vci]]</i>	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 [<i>type slot port-adaptor port</i>]	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 <i>policy-map</i>	Specifies the name of the traffic policy to be created or modified.
Step 2	Router(config-pmap)# class <i>class-name</i>	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 <i>exponent</i>	Configures the exponential weight factor used in calculating the average queue length.
Step 4	Router(config-pmap-c)# bandwidth <i>bandwidth-kbps</i>	Specifies the amount of bandwidth, in kbps, to be assigned to the traffic class.
Step 5	Router(config-pmap-c)# fair-queue [queue-limit <i>queue-values</i>]	Specifies the number of queues to be reserved for the traffic class.
Step 6	Router(config-pmap-c)# queue-limit <i>number-of-packets</i>	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 chapter [Configuring the Modular Quality of Service Command-Line Interface](#) of this book.

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 <i>policy-map</i>	Specifies the name of the traffic policy to be created or modified.
Step 2	Router(config-pmap)# class <i>class-name</i>	Specifies the name of a traffic class to associate with the traffic policy
Step 3	Router(config-pmap-c)# random-detect exponential-weighting-constant <i>exponent</i>	Configures the exponential weight factor used in calculating the average queue length.
Step 4	Router(config-pmap-c)# random-detect precedence <i>precedence min-threshold max-threshold mark-prob-denominator</i>	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 chapter [Configuring the Modular Quality of Service Command-Line Interface](#) of this book.

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 <i>policy-map-name</i>	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 <i>interface-spec</i>	Displays configuration and statistics of the input and output policies attached to a particular interface.
Router# show policy-map interface <i>interface-spec input</i>	Displays configuration and statistics of the input policy attached to an interface.
Router# show policy-map interface <i>interface-spec output</i>	Displays configuration statistics of the output policy attached to an interface.
Router# show policy-map [interface [<i>interface-spec</i> [<i>input</i> <i>output</i>] [<i>class class-name</i>]]]	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 <i>scaling-factor</i>	Sets the flow threshold multiplier for flow-based WRED.
Step 3	Router(config-if)# random-detect flow count <i>number</i>	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 <i>dscp-based</i>	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 <i>dscpvalue</i> <i>min-threshold max-threshold</i> <i>[mark-probability-denominator]</i>	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 <i>group-name</i> <i>dscp-based</i>	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 <i>dscpvalue</i> <i>min-threshold</i> <i>max-threshold</i> <i>[mark-probability-denominator]</i>	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 <i>group-name</i>]	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 <i>class-map-name</i>	Creates a class map to be used for matching packets to a specified class.
Step 2	Router(config-cmap)# match <i>match criterion</i>	Configures the match criteria for a class map. For more information about match criteria, see the “Creating a Traffic Class” section on page 929 in the chapter Configuring the Modular Quality of Service Command-Line Interface in this book.
Step 3	Router(config-if)# policy-map <i>policy-map</i>	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 <i>class-map-name</i>	Specifies the QoS actions for the default class.

	Command	Purpose
Step 5	Router(config-pmap-c)# bandwidth { <i>bandwidth-kbps</i> percent <i>percent</i> }	Specifies or modifies the bandwidth allocated for a class belonging to a policy map.
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 <i>dscpvalue</i> <i>min-threshold</i> <i>max-threshold</i> [<i>mark-probability-denominator</i>]	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 <i>policy-map</i>	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(WRED)
  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 drop	Tail drop	Minimum threshold	Maximum threshold	Mark 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 drop	Tail drop	Minimum threshold	Maximum threshold	Mark probability	Output 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](#)
- [Modular QoS CLI Example](#)
- [Configuring DWRED in Traffic Policy Example](#)

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 for congestion 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

The following sections provide DiffServ Compliant WRED configuration examples:

- [WRED Configured to Use the DSCP Value Example](#)
- [DSCP Value Configuration Verification Example](#)

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 Serial16/3
```

```
Serial16/3
```

```
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 pkts/bytes	Random drop pkts/bytes	Tail drop pkts/bytes	Minimum thresh	Maximum thresh	Mark 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



WRED — Explicit Congestion Notification

Feature History

Release	Modification
12.2(8)T	This feature was introduced.

This document describes the WRED — Explicit Congestion Notification feature in Cisco IOS Release 12.2(8)T. It includes the following sections:

- [Feature Overview, page 333](#)
- [Supported Platforms, page 336](#)
- [Supported Standards, MIBs, and RFCs, page 337](#)
- [Prerequisites, page 337](#)
- [Configuration Tasks, page 338](#)
- [Configuration Examples, page 338](#)
- [Command Reference, page 340](#)

Feature Overview

Currently, the congestion control and avoidance algorithms for Transmission Control Protocol (TCP) are based on the idea that packet loss is an appropriate indication of congestion on networks transmitting data using the best-effort service model. When a network uses the best-effort service model, the network delivers data if it can, without any assurance of reliability, delay bounds, or throughput. However, these algorithms and the best-effort service model are not suited to applications that are sensitive to delay or packet loss (for instance, interactive traffic including Telnet, web-browsing, and transfer of audio and video data). Weighted Random Early Detection (WRED), and by extension, Explicit Congestion Notification (ECN), helps to solve this problem.

RFC 3168, *The Addition of Explicit Congestion Notification (ECN) to IP*, states that with the addition of active queue management (for example, WRED) to the Internet infrastructure, routers are no longer limited to packet loss as an indication of congestion.

How WRED Works

WRED makes early detection of congestion possible and provides a means for handling multiple classes of traffic. WRED can selectively discard lower priority traffic when the router begins to experience congestion and provide differentiated performance characteristics for different classes of service. It also protects against global synchronization. Global synchronization occurs as waves of congestion crest, only to be followed by periods of time during which the transmission link is not used to capacity. For these reasons, WRED is useful on any output interface or router where congestion is expected to occur.

WRED is implemented at the core routers of a network. Edge routers assign IP precedences to packets as the packets enter the network. With WRED, core routers then use these precedences to determine how to treat different types of traffic. WRED provides separate thresholds and weights for different IP precedences, enabling the network to provide different qualities of service, in regard to packet dropping, for different types of traffic. Standard traffic may be dropped more frequently than premium traffic during periods of congestion.

For more information about WRED, refer to the “Congestion Avoidance Overview” chapter of the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2.

ECN Extends WRED Functionality

WRED drops packets, based on the average queue length exceeding a specific threshold value, to indicate congestion. ECN is an extension to WRED in that ECN marks packets instead of dropping them when the average queue length exceeds a specific threshold value. When configured with the WRED — Explicit Congestion Notification feature, routers and end hosts would use this marking as a signal that the network is congested and slow down sending packets.

As stated in RFC 3168, *The Addition of Explicit Congestion Notification (ECN) to IP*, implementing ECN requires an ECN-specific field that has two bits—the ECN-capable Transport (ECT) bit and the CE (Congestion Experienced) bit—in the IP header. The ECT bit and the CE bit can be used to make four ECN field combinations of 00 to 11. The first number is the ECT bit and the second number is the CE bit. [Table 23](#) lists each of the ECT and CE bit combination settings in the ECN field and what the combinations indicate.

Table 23 **ECN Bit Setting**

ECT Bit	CE Bit	Combination Indicates
0	0	Not ECN-capable
0	1	Endpoints of the transport protocol are ECN-capable
1	0	Endpoints of the transport protocol are ECN-capable
1	1	Congestion experienced

The ECN field combination 00 indicates that a packet is not using ECN.

The ECN field combinations 01 and 10—called ECT(1) and ECT(0), respectively—are set by the data sender to indicate that the endpoints of the transport protocol are ECN-capable. Routers treat these two field combinations identically. Data senders can use either one or both of these two combinations. For more information about these two field combinations, and the implications of using one over the other, refer to RFC 3168, *The Addition of Explicit Congestion Notification (ECN) to IP*.

The ECN field combination 11 indicates congestion to the endpoints. Packets arriving a full queue of a router will be dropped.

How Packets Are Treated When ECN Is Enabled

- If the number of packets in the queue is below the minimum threshold, packets are transmitted. This happens whether or not ECN is enabled, and this treatment is identical to the treatment a packet receives when WRED only is being used on the network.
- If the number of packets in the queue is between the minimum threshold and the maximum threshold, one of the following three scenarios can occur:
 - If the ECN field on the packet indicates that the endpoints are ECN-capable (that is, the ECT bit is set to 1 and the CE bit is set to 0, or the ECT bit is set to 0 and the CE bit is set to 1)—and the WRED algorithm determines that the packet should have been dropped based on the drop probability—the ECT and CE bits for the packet are changed to 1, and the packet is transmitted. This happens because ECN is enabled and the packet gets marked instead of dropped.
 - If the ECN field on the packet indicates that neither endpoint is ECN-capable (that is, the ECT bit is set to 0 and the CE bit is set to 0), the packet may be dropped based on the WRED drop probability. This is the identical treatment that a packet receives when WRED is enabled without ECN configured on the router.
 - If the ECN field on the packet indicates that the network is experiencing congestion (that is, both the ECT bit and the CE bit are set to 1), the packet is transmitted. No further marking is required.
- If the number of packets in the queue is above the maximum threshold, packets are dropped based on the drop probability. This is the identical treatment a packet receives when WRED is enabled without ECN configured on the router.

For More Information

For more information about implementing ECN and about the changes required at the routers and end hosts, refer to the following RFCs:

- RFC 2309, *Internet Performance Recommendations*
- RFC 2884, *Performance Evaluation of Explicit Congestion Notification (ECN) in IP Networks*
- RFC 3168, *The Addition of Explicit Congestion Notification (ECN) to IP*

Benefits

Improved Method for Congestion Avoidance

This feature provides an improved method for congestion avoidance by allowing the network to mark packets for transmission later, rather than dropping them from the queue. Marking the packets for transmission later accommodates applications that are sensitive to delay or packet loss and provides improved throughput and application performance.

Enhanced Queue Management

Currently, dropped packets indicate that a queue is full and the network is experiencing congestion. When a network experiences congestion, this feature allows networks to mark the IP header of a packet with a CE bit. This marking, in turn, triggers the appropriate congestion avoidance mechanism and

allows the network to better manage the data queues. With this feature, ECN-capable routers and end hosts can respond to congestion before a queue overflows and packets are dropped, providing enhanced queue management.

For more information on the benefits associated with ECN, refer to RFC 2309, *Internet Performance Recommendations*.

Related Documents

- *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2
- *Cisco IOS Quality of Service Solutions Command Reference*, Release 12.2
- RFC 2309, *Internet Performance Recommendations*
- RFC 2884, *Performance Evaluation of Explicit Congestion Notification (ECN) in IP Networks*
- RFC 3168, *The Addition of Explicit Congestion Notification (ECN) to IP*

Supported Platforms

- Cisco 805
- Cisco 806
- Cisco 820
- Cisco 828
- Cisco 1400 series
- Cisco 1600 series
- Cisco 1751
- Cisco 2420
- Cisco 3631
- Cisco 3725
- Cisco 3745
- Cisco 7100 series
- Cisco 7200 series
- Cisco 7500
- Cisco 7700
- Cisco CVA120 series
- Cisco MC3810
- Cisco uBR7200 series
- URM (Universal Route Module)

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that support 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 at <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>

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

No new or modified MIBs are supported by this feature.

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

<http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml>

RFCs

- RFC 2309, *Internet Performance Recommendations*
- RFC 2884, *Performance Evaluation of Explicit Congestion Notification (ECN) in IP Networks*
- RFC 3168, *The Addition of Explicit Congestion Notification (ECN) to IP*

Prerequisites

ECN must be configured through the Modular Quality of Service Command-Line Interface (MQC). For more information about MQC, refer to the “Modular Quality of Service Command-Line Interface” part of the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2.

Configuration Tasks

See the following sections for configuration tasks for the WRED — Explicit Congestion Notification feature. Each task in the list is identified as either required or optional.

- [Configuring Explicit Congestion Notification](#) (required)
- [Verifying the Explicit Congestion Notification Configuration](#) (optional)

Configuring Explicit Congestion Notification

To configure ECN, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router (config) # policy-map <i>policy-map-name</i>	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy. Enters QoS policy-map configuration mode.
Step 2	Router (config-pmap) # class class-default	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.
Step 3	Router (config-pmap) # bandwidth { <i>bandwidth-kbps</i> percent <i>percent</i> }	Specifies or modifies the bandwidth (either in kbps or a percentage) allocated for a class belonging to a policy map. Enters policy-map class configuration mode.
Step 4	Router (config-pmap-c) # random-detect	Enables WRED or distributed WRED (dWRED).
Step 5	Router (config-pmap-c) # random-detect ecn	Enables ECN.

Verifying the Explicit Congestion Notification Configuration

To verify the ECN configuration, use the following commands in EXEC or privileged EXEC mode, as needed:

Command	Purpose
Router# show policy-map	If ECN is enabled, displays ECN marking information for a specified policy map.
Router# show policy-map interface	If ECN is enabled, displays ECN marking information for a specified interface.

Configuration Examples

This section provides the following configuration examples:

- [Enabling ECN Example](#)
- [Verifying the ECN Configuration Example](#)

Enabling ECN Example

The following example enables ECN in the policy map called poll:

```
Router(config)# policy-map poll
Router(config-pmap)# class class-default
Router(config-pmap)# bandwidth per 70
Router(config-pmap-c)# random-detect
Router(config-pmap-c)# random-detect ecn
```

Verifying the ECN Configuration Example

The following is sample output from the **show policy-map** command. The words “explicit congestion notification” (along with the ECN marking information) included in the output indicate that ECN has been enabled.

```
Router# show policy-map

Policy Map poll
Class class-default
  Weighted Fair Queueing
    Bandwidth 70 (%)
    exponential weight 9
    explicit congestion notification
    class      min-threshold  max-threshold  mark-probability
    -----
    0          -              -              1/10
    1          -              -              1/10
    2          -              -              1/10
    3          -              -              1/10
    4          -              -              1/10
    5          -              -              1/10
    6          -              -              1/10
    7          -              -              1/10
    rsvp      -              -              1/10
```

The following is sample output from the **show policy-map interface** command. The words “explicit congestion notification” included in the output indicate that ECN has been enabled.

```
Router# show policy-map interface Serial4/1

Serial4/1

Service-policy output:policy_ecn
  Class-map:precl (match-all)
    1000 packets, 125000 bytes
    30 second offered rate 14000 bps, drop rate 5000 bps
  Match:ip precedence 1
  Weighted Fair Queueing
    Output Queue:Conversation 42
    Bandwidth 20 (%)
    Bandwidth 100 (kbps)
    (pkts matched/bytes matched) 989/123625
    (depth/total drops/no-buffer drops) 0/455/0
    exponential weight:9
    explicit congestion notification
    mean queue depth:0
```

class	Transmitted pkts/bytes	Random drop pkts/bytes	Tail drop pkts/bytes	Minimum threshold	Maximum threshold	Mark probability
0	0/0	0/0	0/0	20	40	1/10
1	545/68125	0/0	0/0	22	40	1/10
2	0/0	0/0	0/0	24	40	1/10
3	0/0	0/0	0/0	26	40	1/10
4	0/0	0/0	0/0	28	40	1/10
5	0/0	0/0	0/0	30	40	1/10
6	0/0	0/0	0/0	32	40	1/10
7	0/0	0/0	0/0	34	40	1/10
rsvp	0/0	0/0	0/0	36	40	1/10

class	ECN Mark pkts/bytes
0	0/0
1	43/5375
2	0/0
3	0/0
4	0/0
5	0/0
6	0/0
7	0/0
rsvp	0/0

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at <http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm>.

New Commands

- **random-detect ecn**

Modified Commands

- **show policy-map**
- **show policy-map interface**



Part 4: 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, and Frame Relay Traffic Shaping (FRTS) provide the functionality for shaping traffic.

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 quality of service (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. (For a description of a traffic descriptor, see the “[Classification Overview](#)” chapter in this book.)

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 queuing 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 Traffic Shaping use a weighted fair queue to delay packets in order to shape the flow, and FRTS uses 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.

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:

- [What Is a Token Bucket?](#)
- [Policing with CAR](#)
- [Traffic Policing](#)
- [Traffic Shaping \(Regulating Packet Flow\)](#)

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 (T_c). 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:

$$\text{mean rate} = \text{burst size} / \text{time interval}$$

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 (B_c) 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:

$$(\text{token bucket capacity in bits} / \text{time interval in seconds}) + \text{established rate in bps} = \text{maximum flow speed in bps}$$

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](#)” chapter in this book. 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 a packet, 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, 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 “[Actual and Compounded Debt Example](#)” section.)

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

The following restrictions apply to policing with CAR:

- 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 queuing (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 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 Traffic Policing configured is placed in 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 chapter in this book.

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. However, 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 IOS Switching Services Configuration Guide*.

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 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).

For more information about traffic shaping, see the [“Regulating Packet Flow Using Traffic Shaping”](#) chapter in this book.



Traffic Policing

This part consists of the following:

- [Configuring Traffic Policing](#)
- [Traffic Policing](#)
- [Two-Rate Policer](#)
- [Policer Enhancement — Multiple Actions](#)
- [Percentage-Based Policing and Shaping](#)
- [Modular QoS CLI \(MQC\) Three-Level Hierarchical Policer](#)
- [ATM Policing by Service Category for SVC/SoftPVC](#)
- [Modular QoS CLI \(MQC\) Unconditional Packet Discard](#)
- [Control Plane Policing](#)





Configuring Traffic Policing

This chapter describes the tasks for configuring the Traffic Policing feature.

For complete conceptual information, see the [“Traffic Policing” section on page 349](#) in the [Policing and Shaping Overview](#) chapter of this book.

For a complete description of the Traffic Policing commands mentioned in this chapter, refer to 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.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the [“Finding Additional Feature Support Information” section on page lxix](#) in the [Using Cisco IOS Software for Release 12.4](#) chapter in this book.

Traffic Policing Configuration Task List

To configure the Traffic Policing feature, perform the tasks described in the following sections. The task in the first section is required; the tasks in the remaining section are optional.

- [Configuring Traffic Policing](#) (Required)
- [Verifying the Traffic Policing Configuration](#) (Optional)
- [Monitoring and Maintaining Traffic Policing](#) (Optional)

See the end of this chapter for the section [“Traffic Policing Configuration Examples.”](#)

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 Modular QoS Command-Line Interface (CLI). For information on the Modular QoS CLI, see the chapter [Configuring the Modular Quality of Service Command-Line Interface](#) in this book.

The Traffic Policing feature is configured in the traffic policy. To configure the Traffic Policing feature, use the following command in policy-map class configuration mode:

Command	Purpose
Router(config-pmap-c)# police <i>bps burst-normal burst-max conform-action action exceed-action action violate-action action</i>	Specifies a maximum bandwidth usage by a traffic class. The police command polices traffic based on a token bucket algorithm. The variables in the token bucket algorithm are set in this command line.

The command syntax of the **police** command allows you to specify the action to be taken on a packet when you enable the *action* keyword. The resulting action corresponding to the keyword choices are listed in [Table 24](#).

Table 24 *police* Command Action Keywords

Keyword	Resulting Action
<i>drop</i>	Drops the packet.
set-prec-transmit <i>new-prec</i>	Sets the IP precedence and sends the packet.
set-qos-transmit <i>new-qos</i>	Sets the QoS group and sends the packet.
set-dscp-transmit <i>new-dscp</i>	Sets the differentiated services code point (DSCP) value and sends the packet.
transmit	Sends the packet.

For more information about the **police** command, refer to 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 a description of a single token bucket algorithm and an explanation of how it works, see the “[What Is a Token Bucket?](#)” section on page 344 of the [Policing and Shaping Overview](#) chapter of this book.

Verifying the Traffic Policing Configuration

To verify that the Traffic Policing feature is configured on your interface, use the following command in EXEC mode:

Command	Purpose
Router# show policy-map interface	Displays statistics and configurations of all input and output policies attached to an interface.

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 traffic policy.
Router# show policy-map <i>policy-map-name</i>	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.

Traffic Policing Configuration Examples

The following sections provide Traffic Policing configuration examples:

- [Traffic Policy that Includes Traffic Policing Example](#)
- [Verifying the Configuration Example](#)

For information on how to configure the Traffic Policing feature, see the section “[Traffic Policing Configuration Task List](#)” in this chapter.

Traffic Policy that Includes Traffic Policing Example

The following configuration example 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 chapter [Configuring the Modular Quality of Service Command-Line Interface](#) in this book.

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 sent, packets that exceed are assigned a QoS group value of 4 and are sent, and packets that violate are dropped.

For a description of a token bucket and an explanation of how a token bucket works, see the “[What Is a Token Bucket?](#)” section on page 344 of the [Policing and Shaping Overview](#) chapter of this book.

```
7200-uit(config)# class-map acgroup2
7200-uit(config-cmap)# match access-group 2
7200-uit(config-cmap)# exit
7200-uit(config)# policy-map police
7200-uit(config-pmap)# class acgroup2
7200-uit(config-pmap-c)# police 8000 2000 4000 conform-action transmit exceed-action
set-qos-transmit 4 violate-action drop
7200-uit(config-pmap-c)# exit
7200-uit(config-pmap)# exit
7200-uit(config)# interface fastethernet 0/0
7200-uit(config-if)# service-policy input police
```

Verifying the Configuration Example

The following example verifies that the Traffic 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
```



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.

This document includes the following sections:

- [Feature Overview, page 359](#)
- [Supported Platforms, page 362](#)
- [Supported Standards, MIBs, and RFCs, page 362](#)
- [Prerequisites, page 363](#)
- [Configuration Tasks, page 363](#)
- [Monitoring and Maintaining Traffic Policing, page 364](#)
- [Configuration Examples, page 364](#)
- [Command Reference, page 365](#)
- [Glossary, page 366](#)

Feature Overview

Table 25 **Feature History**

Cisco IOS Release	Enhancement
12.0(5)XE	This feature was introduced.
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.

Table 25 **Feature History (continued)**

Cisco IOS Release	Enhancement
12.2(2)T	<p>The set-clp-transmit option for the <i>action</i> argument was added to the police command.</p> <p>The set-frde-transmit option for the <i>action</i> 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.</p> <p>The set-mpls-exp-transmit option for the <i>action</i> argument was added to the police command.</p>

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 Command-Line Interface (Modular QoS CLI). For information on configuring the Modular QoS CLI, see the *Modular Quality of Service Command-Line Interface Overview* on Cisco Connection Online (CCO) and the Documentation CD-ROM.

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 *Class-Based Marking* feature 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

- *Modular Quality of Service Command-Line Interface* document
- *Committed Access Rate* feature module
- *Class-Based Marking* feature module
- *Class-Based Weighted Fair Queuing* feature module

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, refer to 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 Cisco Express Forwarding, see the *Cisco Express Forwarding and Distributed Cisco Express Forwarding* feature modules.

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 Modular QoS CLI. For information on the Modular QoS CLI, see the *Modular Quality of Service Command-Line Interface* document on CCO or the Documentation CD-ROM.

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 <i>bps burst-normal burst-max conform-action action exceed-action action violate-action action</i>	Specifies a maximum bandwidth usage by a traffic class.

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 a description of a single token bucket algorithm and an explanation of how it works, see the “What is a Token Bucket?” section of the *Policing and Shaping Overview* document. An example of how the single token bucket algorithm works is also given in the “[Command Reference](#)” section of this document.

For a description of the two token bucket algorithm and an explanation of how it works, see the “[Command Reference](#)” section of this document.

Verifying Traffic Policing

Use the **show policy-map interface EXEC** command to verify that the Traffic 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 document.
- 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 <i>policy-map-name</i>	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](#)

Configuring a Service Policy that Includes Traffic Policing

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 *Modular Quality of Service Command-Line Interface* document on CCO and the Documentation CD-ROM.

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 a description of a token bucket and an explanation of how a token bucket works, see the “What is a Token Bucket?” section of the *Policing and Shaping Overview* document. An example of how the token bucket works is also given in the “[Command Reference](#)” section of this document.

```
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 modified command is pertinent to this feature. To see the command pages for this command and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at <http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm>.

- **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.



Two-Rate Policer

Feature History

Release	Modification
12.2(4)T	This feature was introduced.
12.2(4)T3	Support for the Cisco 7500 series routers was added.

This document describes the Two-Rate Policer feature in Cisco IOS Release 12.2(4)T. It includes the following sections:

- [Feature Overview, page 367](#)
- [Supported Platforms, page 370](#)
- [Supported Standards, MIBs, and RFCs, page 371](#)
- [Configuration Tasks, page 371](#)
- [Monitoring and Maintaining the Two-Rate Policer, page 373](#)
- [Configuration Examples, page 373](#)
- [Command Reference, page 374](#)

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. For more information about the **police** command, see the “[Command Reference](#)” section of this document.

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, refer to the “Implementing DiffServ for End-to-End Quality of Service” chapter of the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2.

**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, refer to the “Policing and Shaping Overview” chapter of the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2.

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 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 *Class-Based Marking* feature module available with Cisco IOS Release 12.2(2)T. More information about the Class-Based Marking feature is available from the Cisco documentation website (Cisco.com) or the Cisco documentation CD.

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

Related Features and Technologies

- Modular Quality of Service Command-Line Interface
- Class-based Weighted Fair Queueing (CBWFQ)
- Class-Based Packet Marking
- Traffic Policing

Related Documents

- *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2
- *Cisco IOS Quality of Service Solutions Command Reference*, Release 12.2
- RFC 2698, *A Two Rate Three Color Marker*

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, refer to the documentation for your specific router.

Platform Support Through Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Feature Navigator. Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

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.

To access 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 at <http://www.cisco.com/register>.

Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Feature Navigator home page at the following URL:

<http://www.cisco.com/go/fn>

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

The Two-Rate Policer feature supports the following MIBs:

- CISCO-CLASS-BASED-QOS-MIB
- CISCO-CLASS-BASED-QOS-CAPABILITY-MIB

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

<http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml>

RFCs

This feature supports RFC 2698, *A Two Rate Three Color Marker*.

Prerequisites

- 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, refer to the *Cisco IOS Switching Services Configuration Guide*, Release 12.2.
- 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 QoS CLI. For information on the Modular QoS CLI, see the “Modular Quality of Service Command-Line Interface” chapter of the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2.

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 { <i>cir cir</i> } [bc <i>conform-burst</i>] { pir pir } [be <i>peak-burst</i>]	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 (<i>conform-burst</i> and <i>peak-burst</i> , 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 26 *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 <i>new-dscp</i>	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 <i>new-prec</i>	Sets the IP precedence and sends the packet with the new IP precedence value setting.
set-qos-transmit <i>new-qos</i>	Sets the QoS group value and sends the packet with the new QoS group value setting.
transmit	Sends the packet with no alteration.

For more information about the **police** command, see the “[Command Reference](#)” section of this document.

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 the single-rate Traffic Policing feature, refer to the “Policing and Shaping Overview” chapter in the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2.

For more information about the two token bucket algorithm, see the “[Command Reference](#)” section of this document.

Verifying the Two-Rate Policer Configuration

To verify that the Two-Rate Policer is configured on your interface, use the following command in EXEC or privileged EXEC mode:

Command	Purpose
Router# <code>show policy-map interface</code>	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 document.
- 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 EXEC or privileged EXEC mode commands:

Command	Purpose
Router# <code>show policy-map</code>	Displays all configured policy maps.
Router# <code>show policy-map policy-map-name</code>	Displays the user-specified policy map.
Router# <code>show policy-map interface</code>	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-pmap-c)# interface s3/0
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 s3/0
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.

Command Reference

The following modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at <http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm>.

- **police**
- **show policy-map**
- **show policy-map interface**



Policer Enhancement — Multiple Actions

Feature History

Release	Modification
12.2(8)T	This feature was introduced.

This document describes the Policer Enhancement — Multiple Actions feature in Cisco IOS Release 12.2(8)T. It includes the following sections:

- [Feature Overview, page 375](#)
- [Supported Platforms, page 377](#)
- [Supported Standards, MIBs, and RFCs, page 378](#)
- [Prerequisites, page 379](#)
- [Configuration Tasks, page 379](#)
- [Monitoring and Maintaining the Multiple Policer Actions, page 380](#)
- [Configuration Examples, page 381](#)
- [Command Reference, page 382](#)

Feature Overview

This feature further extends the functionality of the Cisco IOS Traffic Policing feature (a single-rate policer) and the Two-Rate Policer feature. The Traffic Policing and Two-Rate Policer features are traffic policing mechanisms that allow you to control the maximum rate of traffic sent or received on an interface. Both of these traffic policing mechanisms mark packets as either conforming to, exceeding, or violating a specified rate. After a packet is marked, you can specify an action to be taken on the packet based on that marking.

With both the Traffic Policing feature and the Two-Rate Policer feature, you can specify only one conform action, one exceed action, and one violate action. Now with the new Policer Enhancement — Multiple Actions feature, you can specify multiple conform, exceed, and violate actions for the marked packets.

You specify the multiple actions by using the *action* argument of the **police** command. The resulting actions are listed in [Table 27](#).

Table 27 *police Command Action Arguments*

Specified Action	Result
drop	Drops the packet.
set-clp-transmit	Sets the ATM Cell Loss Priority (CLP) bit from 0 to 1 on the ATM cell and transmits the packet.
set-dscp-transmit <i>new-dscp</i>	Sets the IP differentiated services code point (DSCP) value and transmits the packet with the ATM CLP bit set to 1.
set-frde-transmit	Sets the Frame Relay Discard Eligibility (DE) bit from 0 to 1 on the Frame Relay frame and transmits the packet.
set-mpls-exp-transmit	Sets the Multiprotocol Label Switching (MPLS) experimental (EXP) bits from 0 to 7 and transmits the packet.
set-prec-transmit <i>new-prec</i>	Sets the IP Precedence level and transmits the packet.
set-qos-transmit <i>new-qos</i>	Sets the Quality of Service (QoS) group value and transmits the packet.
transmit	Transmits the packet.

For more information about the **police** command and how to use it with the Policer Enhancement — Multiple Actions feature, see the “[Command Reference](#)” section of this document.

For more information about the Cisco IOS Traffic Policing feature, refer to the “Policing and Shaping” section of the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2. For more information about the Two-Rate Policer feature, refer to the new features for Release 12.2(4)T on Cisco.com.

Benefits

Before this feature, you could specify only *one* marking action for a packet, in addition to transmitting the packet. This feature provides enhanced flexibility by allowing you to specify *multiple* marking actions for a packet, as required. For example, if you know the packet will be transmitted through both a TCP/IP and a Frame Relay environment, you can change the DSCP value of the exceeding or violating packet, and also set the Frame Relay Discard Eligibility (DE) bit from 0 to 1 to indicate lower priority.

Restrictions

- 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.
- Multiple policer actions can be configured on an interface, a subinterface, a Frame Relay data-link connection identifier (DLCI), and an ATM permanent virtual circuit (PVC) only.

- When using this feature, you can specify a maximum of four actions at one time.
- Multiple policer actions are not supported on the following interfaces:
 - Fast EtherChannel
 - PRI
 - Any interface on a Cisco 7500 series router that does not support CEF or dCEF

Related Features and Technologies

- Modular Quality of Service Command-Line Interface (Modular QoS CLI)
- Class-Based Weighted Fair Queueing (CBWFQ)
- Class-Based Packet Marking
- Traffic Policing
- Two-Rate Policing

Related Documents

- *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2
- *Cisco IOS Quality of Service Solutions Command Reference*, Release 12.2
- *Cisco IOS Switching Services Configuration Guide*, Release 12.2
- *Two-Rate Policer*, Cisco IOS Release 12.2(4)T feature module
- RFC 2697, *A Single Rate Three Color Marker*
- RFC 2698, *A Two Rate Three Color Marker*

Supported Platforms

- Cisco 1700 series
- Cisco 2600 series
- Cisco 3620
- Cisco 3640
- Cisco 3660
- Cisco 7100 series
- Cisco 7200 series
- Cisco 7500 series (VIP-based platform only)
- Cisco MC3810

**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 and the Cisco 3640 router). For more information, refer to the documentation for your specific router.

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that support 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 at <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>

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

- CISCO-CLASS-BASED-QOS-MIB
- CISCO-CLASS-BASED-QOS-CAPABILITY-MIB

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

<http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml>

RFCs

- RFC 2697, *A Single Rate Three Color Marker*
- RFC 2698, *A Two Rate Three Color Marker*

Prerequisites

- Before configuring the Policer Enhancement — Multiple Actions feature, you should read and understand the following:
 - *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2
Specifically, the “Configuring Traffic Policing” chapter and the “Policing and Shaping Overview” chapter.
 - *Two-Rate Policer*, Cisco IOS Release 12.2(4)T feature module
- On a Cisco 7500 series router, CEF or dCEF must be configured on the interface before you can use the Policer Enhancement — Multiple Actions feature. For additional information on CEF or dCEF, refer to the *Cisco IOS Switching Services Configuration Guide*, Release 12.2.
- To configure the Policer Enhancement — Multiple Actions feature, 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 QoS CLI. For information on the Modular QoS CLI, refer to the “Modular Quality of Service Command-Line Interface Overview” chapter of the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2.

Configuration Tasks

See the following sections for configuration tasks for the Police Enhancement — Multiple Actions feature. Each task in the list is identified as either required or optional.

- [Configuring Multiple Policer Actions](#) (required)
- [Verifying the Multiple Policer Actions Configuration](#) (optional)

Configuring Multiple Policer Actions

To configure multiple policer actions, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# policy-map <i>policy-map-name</i>	Creates a policy map. Enters policy-map configuration mode.
Step 2	Router(config-pmap)# class <i>class-default</i>	Specifies the default traffic class for a service policy. Enters policy-map class configuration mode.
Step 3	Router(config-pmap-c)# police { <i>cir cir</i> } [bc <i>conform-burst</i>] { pir <i>pir</i> } [be <i>peak-burst</i>] [conform-action <i>action</i> [exceed-action <i>action</i> [violate-action <i>action</i>]]]	Configures traffic policing and specifies multiple actions applied to packets marked as conforming to, exceeding, or violating a specific rate. Use one line per action that you want to specify. Enters policy-map class police configuration mode.

Verifying the Multiple Policer Actions Configuration

To verify that the multiple policer actions have been configured on the interface, use the following command in 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 document.
- 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 Multiple Policer Actions

To monitor and maintain the multiple policer actions, use the following EXEC or privileged EXEC mode commands, as needed:

Command	Purpose
Router# show policy-map	Displays all configured policy maps.
Router# show policy-map <i>policy-map-name</i>	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 examples:

- [Multiple Actions in a Two-Rate Policer Example](#)
- [Verifying the Multiple Policer Actions Example](#)

Multiple Actions in a Two-Rate Policer Example

In the following example, a policy map called police is configured to use a two-rate policer to police traffic leaving an interface. Two rates, a committed information rate (CIR) of 1 Mbps and a peak information rate (PIR) of 2 Mbps, have been specified.

```
Router(config)# policy-map police
Router(config-pmap)# class class-default
Router(config-pmap-c)# police cir 1000000 pir 2000000
Router(config-pmap-c-police)# conform-action transmit
Router(config-pmap-c-police)# exceed-action set-prec-transmit 4
Router(config-pmap-c-police)# exceed-action set-frde
Router(config-pmap-c-police)# violate-action set-prec-transmit 2
Router(config-pmap-c-police)# violate-action set-frde-transmit
Router(config-pmap-c-police)# end
```

The following actions will be performed on packets associated with the policy map called police:

- All packets marked as conforming to these rates (that is, packets conforming to the CIR) will be transmitted unaltered.
- All packets marked as exceeding these rates (that is, packets exceeding the CIR but not exceeding the PIR) will be assigned an IP Precedence level of 4, the DE bit will be set to 1, and then transmitted.
- All packets marked as violating the rate (that is, exceeding the PIR) will be assigned an IP Precedence level of 2, the DE bit will be set to 1, and then transmitted.

Verifying the Multiple Policer Actions Example

The following sample output of the **show policy-map** command displays the configuration for a service policy called police. In this service policy, multiple actions for packets marked as exceeding the specified CIR rate have been configured. For those packets, the IP Precedence level is set to 4, the DE bit is set to 1, and the packet is transmitted. Multiple actions for packets marked as violating the specified PIR rate have also been configured. For those packets, the IP Precedence level is set to 2, the DE bit is set to 1, and the packet is transmitted.

```
Router# show policy-map police

Policy Map police
Class class-default
  police cir 1000000 bc 31250 pir 2000000 be 31250
    conform-action transmit
    exceed-action set-prec-transmit 4
    exceed-action set-frde-transmit
    violate-action set-prec-transmit 2
    violate-action set-frde-transmit
```

Command Reference

The following modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at <http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm>.

- **police**
- **show policy-map**
- **show policy-map interface**



Percentage-Based Policing and Shaping

Feature History

Release	Modification
12.2(13)T	This feature was introduced.

Supported Platforms

For platforms supported in Cisco IOS Release 12.2(13)T, consult Cisco Feature Navigator.

This document describes the Percentage-Based Policing and Shaping feature in Cisco IOS Release 12.2(13)T. It includes the following sections:

- [Feature Overview, page 383](#)
- [Supported Platforms, page 385](#)
- [Supported Standards, MIBs, and RFCs, page 385](#)
- [Configuration Tasks, page 386](#)
- [Configuration Examples, page 388](#)
- [Command Reference, page 390](#)

Feature Overview

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 based on 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.

This feature also provides the option of specifying burst sizes in milliseconds (ms) when configuring traffic policing and shaping based on a percentage of bandwidth.

Configuring traffic policing and shaping based on a percentage of bandwidth is accomplished by using the **police** (percent) and **shape** (percent) commands. For more information about these commands, see the “[Command Reference](#)” section later in this document.

For more information on traffic policing and traffic shaping, refer to the chapter “Policing and Shaping Overview” in the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2.

Benefits

Increased Flexibility

This feature provides the ability to configure traffic policing and traffic shaping based on a *percentage* of bandwidth available on an 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.

Restrictions

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.

Related Features and Technologies

- Modular QoS command-line interface (CLI) (Modular QoS CLI)
- Class-based weighted fair queueing (CBWFQ)
- Class-based packet marking
- Traffic policing
- Two-rate policing
- Traffic shaping

Related Documents

- *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2
- *Cisco IOS Quality of Service Solutions Command Reference*, Release 12.2
- *Cisco IOS Switching Services Configuration Guide*, Release 12.2
- *Cisco IOS Switching Services Command Reference*, Release 12.2
- *Class-Based Marking*, Cisco IOS Release 12.2(2)T feature module
- *Two-Rate Policer*, Cisco IOS Release 12.2(4)T feature module
- *Policer Enhancements—Multiple Actions*, Cisco IOS Release 12.2(8)T feature module
- RFC 2697, *A Single Rate Three Color Marker*

- RFC 2698, *A Two Rate Three Color Marker*

Supported Platforms

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that are supported on specific platforms. To obtain updated information about 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 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. In the release section, you can compare releases side by side to display both the features unique to each software release and the features that releases have 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 at <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, refer to the online release notes or, if supported, Cisco Feature Navigator.

Supported Standards, MIBs, and RFCs

Standards

None

MIBs

None

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

- RFC 2697, *A Single Rate Three Color Marker*
- RFC 2698, *A Two Rate Three Color Marker*

Prerequisites

On a Cisco 7500 series router, Distributed Cisco Express Forwarding (dCEF) must be configured on the interface before you can use the Percentage-Based Policing and Shaping feature. For additional information on dCEF, refer to the *Cisco IOS Switching Services Configuration Guide*, Release 12.2.

Configuration Tasks

See the following sections for configuration tasks for the Percentage-Based Policing and Shaping feature. Each task in the list is identified as either required or optional.

- [Configuring Policing and Shaping Based on Bandwidth Percentage](#) (required)
- [Attaching the Policy Map to an Interface or a VC](#) (required)
- [Verifying the Policing and Shaping Bandwidth Percentage Setting](#) (optional)

Configuring Policing and Shaping Based on Bandwidth Percentage

To configure traffic policing and shaping based on a percentage of bandwidth available on an interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router (config)# policy-map <i>policy-name</i>	Specifies the name of the policy map to be created. Enters policy-map configuration mode.
Step 2	Router(config-pmap)# class-map <i>class-map-name</i>	Specifies the name of the class map to be created. Enters policy-map class configuration mode.
Step 3	Router(config-pmap-c)# police cir percent <i>percent</i> [bc <i>conform-burst-in-msec</i>] [pir percent <i>percent</i>] [be <i>peak-burst-in-msec</i>]	Configures traffic policing.
Step 4	Router(config-pmap-c)# shape { average peak } percent <i>percent</i> [bc] [be]	Configures traffic shaping using either an average or peak traffic shaping rate based on a percentage of available bandwidth.

	Command	Purpose
Step 5	Router (config-pmap-c) # service-policy <i>policy-map-name</i>	Specifies the name of a policy map to be used as a child policy map for this class.
Step 6	Router (config-pmap-c) # exit	Exits policy-map class configuration mode.

Attaching the Policy Map to an Interface or a VC

To attach the policy map to an interface or a virtual circuit (VC), use the following command in interface configuration mode. To attach the policy map to a VC, use the following command in ATM VC configuration mode.

Command	Purpose
Router (config-if) # service-policy output ¹ <i>policy-map-name</i>	Specifies the name of the policy map to be attached to the input direction of an interface or VC. The policy map evaluates all traffic entering that interface or VC.
or	
Router (config-if-atm-vc) # service-policy output <i>policy-map-name</i>	

1. Traffic shaping is supported on service policies attached to output interfaces or output VCs only.

Verifying the Policing and Shaping Bandwidth Percentage Setting

To verify the policing and shaping bandwidth percentages in the class map and the associated policy map, use the following commands in EXEC mode, as needed:

Command	Purpose
Router# show class-map	Displays all information about a class map, including the match criterion.
Router# show policy-map	Displays all configured policy maps.
Router# show policy-map interface <i>interface-name</i>	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 permanent virtual circuit (PVC) on the interface.

Troubleshooting Tips

- For input traffic policing on a Cisco 7500 series router, verify that 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.

Configuration Examples

This section provides the following configuration examples:

- [Specifying Traffic Policing Based on a Bandwidth Percentage Example](#)
- [Specifying Traffic Shaping Based on a Bandwidth Percentage Example](#)
- [Verifying That CEF Is Enabled Example](#)

Specifying Traffic Policing Based on a Bandwidth Percentage Example

The following example configures traffic policing using a committed information rate (CIR) and a peak information rate (PIR) based on 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 (config)# policy-map policy1
Router(config-pmap)# class-map class1
Router(config-pmap-c)# police cir percent 20 bc 300 ms pir percent 40 be 400 ms
Router (config-pmap-c)# service-policy child-policy1
Router(config-pmap-c)# exit
Router(config-pmap-c)# interface serial 3/1
Router(config-if)# service-policy output policy1
```

The purpose of the burst parameters (bc and be values) 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.

Specifying Traffic Shaping Based on a Bandwidth Percentage Example

The following example configures traffic shaping using an average shaping rate based on a percentage of bandwidth. In this example, 25 percent of the bandwidth has been specified. Additionally, an optional bc value and be value (300 ms and 400 ms, respectively) have been specified.

```
Router (config)# policy-map policy1
Router(config-pmap)# class-map class1
Router(config-pmap-c)# shape average percent 25 300 ms 400 ms
Router (config-pmap-c)# service-policy child-policy1
Router(config-pmap-c)# exit
Router(config-pmap-c)# interface serial 3/1
Router(config-if)# service-policy output policy1
```

The purpose of the bc and be values is to drop packets gradually, as is done with WRED, and to avoid tail drop. Setting sufficiently high burst values helps to ensure good throughput.

Verifying That CEF Is Enabled Example

As mentioned previously, on a Cisco 7500 series router, dCEF must be configured on the interface before you can use the Percentage-Based Policing and Shaping feature. The **show ip cef summary** command can be used to confirm that dCEF is enabled and is being used for IP switching. In rare instances, this command displays “IP Distributed CEF without switching” in the command output. This indicates that dCEF is disabled. The following sample output of the **show ip cef summary** command indicates that dCEF is disabled:

```
Router# show ip cef summary

IP Distributed CEF with switching (Table Version 36), flags=0x0
  18 routes, 0 reresolve, 0 unresolved (0 old, 0 new), peak 3
  18 leaves, 19 nodes, 22136 bytes, 45 inserts, 27 invalidations
  0 load sharing elements, 0 bytes, 0 references
  universal per-destination load sharing algorithm, id 680E93E2
  3(0) CEF resets, 1 revisions of existing leaves
  Resolution Timer:Exponential (currently 1s, peak 1s)
  0 in-place/0 aborted modifications
  refcounts: 5136 leaf, 5120 node
```

For information on enabling dCEF, refer to the *Cisco IOS Switching Services Configuration Guide*, Release 12.2.

When you configure a feature that requires special handling or is not yet supported in the dCEF switching paths, packets are forwarded to the next switching layer for handling. In this instance, the output of the **show cef interface** command displays “Packets switched to this interface on line card are dropped to next slow path” as shown in the following sample output.

```
Router# show cef interface Serial 10/0/0:28

Serial10/0/0:28 is up (if_number 38)
  Internet address is 90.0.0.1/8
  ICMP redirects are never sent
  Per packet loadbalancing is disabled
  Inbound access list is not set
  Interface is marked as point to point interface
  Packets switched to this interface on linecard are dropped to next slow path
  Hardware idb is Serial10/0/0:28
  Fast switching type 4, interface type 20
  IP Distributed CEF switching enabled
  Fast flags 0x0. ifindex 37(37)
  Slot 10 Slot unit 0 VC 28
  Hardware transmit queue ptr 0x48001AE0 (0x48001AE0)
  Transmit limit accumulator 0x48000102 (0x48000102)
  IP MTU 1500
```

For more information about the **show ip cef interface** command, refer to the *Cisco IOS Switching Services Command Reference*, Release 12.2.

Command Reference

The following modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at <http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm>.

- **police (percent)**
- **shape (percent)**
- **show policy-map**
- **show policy-map interface**



Modular QoS CLI (MQC) Three-Level Hierarchical Policer

The Modular QoS CLI (MQC) Three-Level Hierarchical Policer extends the traffic policing functionality by allowing you to configure traffic policing at *three* levels of policy map hierarchies; a primary level, a secondary level, and a tertiary level. Traffic policing may be configured at any or all of these levels, depending on the needs of your network. Configuring traffic policing in a three-level hierarchical structure provides a high degree of granularity for traffic policing.

Feature Specifications for the Modular QoS CLI (MQC) Three-Level Hierarchical Policer

Feature History

Release	Modification
12.2(13)T	This feature was introduced.

Supported Platforms

For platforms supported in Cisco IOS Release 12.2(13)T, consult Cisco Feature Navigator.

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, refer to the online release notes or, if supported, Cisco Feature Navigator.

Contents

- [Restrictions for the Modular QoS CLI \(MQC\) Three-Level Hierarchical Policer, page 392](#)
- [Information About the Modular QoS CLI \(MQC\) Three-Level Hierarchical Policer, page 393](#)
- [How to Configure the Modular QoS CLI \(MQC\) Three-Level Hierarchical Policer, page 395](#)
- [Configuration Examples for the Modular QoS CLI \(MQC\) Three-Level Hierarchical Policer, page 400](#)
- [Additional References, page 403](#)
- [Command Reference, page 405](#)

Restrictions for the Modular QoS CLI (MQC) Three-Level Hierarchical Policer

If traffic policing is configured at both the top level and secondary levels, note the following caveats:

- When traffic policing is configured at both the primary and secondary levels, the traffic policer at the secondary level acts only on packets sent by the policer at the top level.

However, the packet classification for the policy map at the secondary level occurs before the primary level policer has acted on the classes. When this situation occurs, the class counters for the policy map at the secondary level may not be equal to the number of packets acted upon by the second level policer.

The following output of the **show policy-map interface** command helps to illustrate this point. In this sample output two policy maps (called “primary_level,” and “secondary_level,” respectively) have been configured. The primary_level policy map contains a class map called “c1,” and the secondary_level policy map contains a class map called “c3”.

```
> > > show policy interface serial5/0.1
> > > Serial5/0.1
> > >
> > > Service-policy output: primary_level
> > >
> > > Class-map: c1 (match-all)
> > > 24038 packets, 3004750 bytes
> > > 30 second offered rate 0 bps, drop rate 0 bps
> > > Match: any
> > > police:
> > >   cir 300000 bps, bc 9375 bytes
> > >   conformed 18105 packets, 2263125 bytes; actions:
> > >   transmit
> > >   exceeded 5933 packets, 741625 bytes; actions: (*)
> > >   drop
> > >   conformed 0 bps, exceed 0 bps
> > >
```

```

> > >      Service-policy : secondary_level
> > >
> > >      Class-map: c3 (match-all)
> > >          24038 packets, 3004750 bytes
> > >          30 second offered rate 0 bps, drop rate 0 bps
> > >      Match: any
> > >      police: (<= Indicates traffic policing has been configured)
> > >          cir 200000 bps, bc 3000 bytes
> > >          pir 250000 bps, be 3000 bytes
> > >          conformed 12047 packets, 1505875 bytes; actions:      (**)
> > >          set-frde-transmit
> > >          exceeded 3004 packets, 375500 bytes; actions:      (**)
> > >          set-frde-transmit
> > >          violated 3054 packets, 381750 bytes; actions:      (**)
> > >          set-frde-transmit
> > >          conformed 0 bps, exceed 0 bps, violate 0 bps
> > >
> > >      Class-map: class-default (match-any)
> > >          0 packets, 0 bytes
> > >          30 second offered rate 0 bps, drop rate 0 bps
> > >      Match: any
> > >          0 packets, 0 bytes
> > >          30 second rate 0 bps

```

Note the following about this example:

- The class counter for the class map called “c3” shows 24038 packets (italicized in the example).
- Traffic policing has been configured in the policy map, and the traffic policing feature for class map “c3” shows a total of 18105 packets — 12047 conformed packets, plus 3004 exceeded packets, plus 3054 violated packets (indicated by the double asterisks (“**”) in the example). This total is because 5933 packets have already been dropped in class map “c1” (indicated by the “*” in the example).
- Therefore, only 18105 packets (24038 packets minus 5933 packets) are acted upon by the traffic policing feature configured in the second_level policy map.
- In this implementation of the Modular QoS CLI (MQC) Three-Level Hierarchical Policer, traffic policing at the primary level does not guarantee fairness in sharing bandwidth among the child classes. If packets from two different classes arrive at the same rate and then go through a traffic policer, the output rates of the two classes could be different because this feature acts as an aggregate policer.

In other words, it is possible that the primary-level policer could drop packets in one class in favor of the other class. This situation would happen because the primary-level policer had enough tokens when the packets for one class arrived, but there were not enough tokens left for the other class. This pattern could continue indefinitely, based on the arrival pattern of the packets.

Information About the Modular QoS CLI (MQC) Three-Level Hierarchical Policer

To configure the Modular QoS CLI (MQC) Three-Level Hierarchical Policer, you need to understand the following concepts:

- [Modular Quality of Service Command-Line Interface \(MQC\), page 394](#)
- [Packet Flow in the Modular QoS CLI \(MQC\) Three-Level Hierarchical Policer, page 394](#)
- [Other Traffic Policing-Related Features, page 395](#)

Modular Quality of Service Command-Line Interface (MQC)

The MQC is a command-line interface (CLI) structure that allows you to create traffic policies and attach these 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 Modular quality of service (QoS) CLI structure 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.

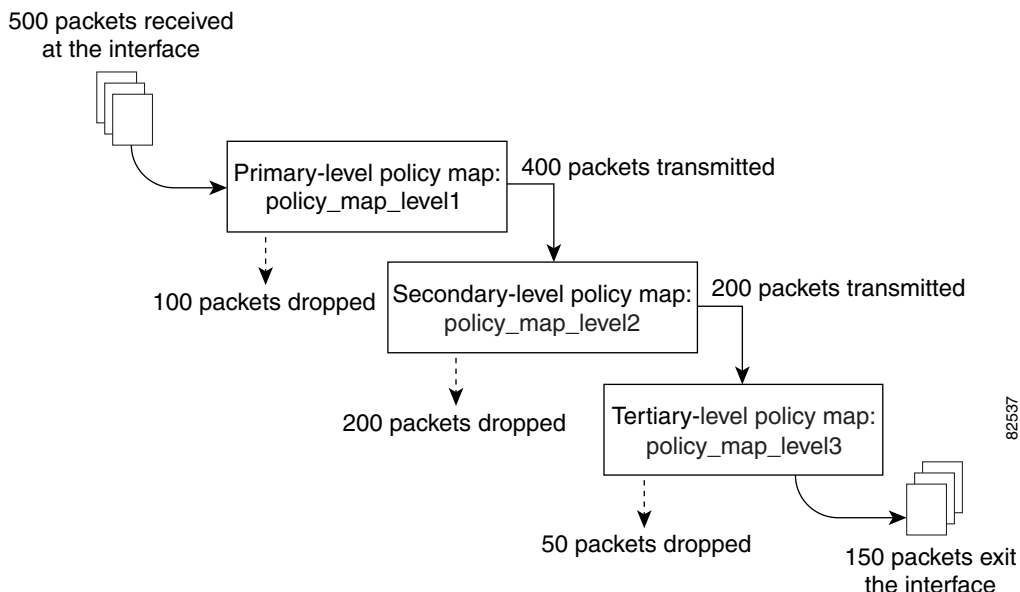
A traffic class contains three major elements: a name, a series of **match** commands, and, if more than one **match** command exists in the traffic class, an instruction on how to evaluate these **match** commands. The traffic class is named in the **class-map** command line; that is, if you enter the **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.

Packet Flow in the Modular QoS CLI (MQC) Three-Level Hierarchical Policer

Figure 14 illustrates the flow of packets among policy maps configured for traffic policing at each level in the hierarchy.

Figure 14 Packet Flow Among Policy Maps



In [Figure 14](#), three policy maps are configured: `policy_map_level1` (the primary-level policy map), `policy_map_level2` (the secondary-level policy map), and `policy_map_level3` (the tertiary-level policy map). Traffic policing is configured in each policy map, and each policy map is attached to a service policy and to an interface.

In this simplified illustration, 500 packets arrive at the interface at which the policy map called “`policy_map_level1`” is attached. Because of the way traffic policing is configured in this policy map, 100 packets are dropped and 400 packets are transmitted.

The traffic policer at the secondary-level policy map (`policy_map_level2`) then evaluates the packets and treats them as determined by the way traffic policing is configured at this level. Of the 400 packets received, 200 are dropped and 200 are transmitted.

The traffic policer at the tertiary-level policy map (`policy_map_level3`), in turn, evaluates the 200 packets it has now received and applies the appropriate treatment as determined by the way the traffic policing is configured at this level.

Other Traffic Policing-Related Features

The Cisco IOS traffic policing software features allow 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 or violates the parameters is dropped or sent with a different priority.

The Cisco IOS software currently includes the following traffic policing features:

- Traffic Policing (a single-rate policer)
- Two-Rate Policer
- Policer Enhancements — Multiple Actions
- Percentage-Based Policing and Shaping

Previously, these features could be configured at two levels of a policy map hierarchy; the top level and one secondary level. With the Modular QoS CLI (MQC) Three-Level Hierarchical Policer, these traffic policing-related features can be configured in three levels of a policy map hierarchy.

The tasks for configuring each of these traffic policing-related features is essentially the same. That is, you use the MQC to create a policy map. Then you use the **police** command to configure traffic policing for a specific class within that policy map. The policy map is then attached to an interface.

Traffic policing can be configured to specify multiple marking actions for the traffic being policed, or to use a percentage of available bandwidth when policing traffic.

How to Configure the Modular QoS CLI (MQC) Three-Level Hierarchical Policer

This section contains the following procedures. Each procedure is identified as either required or optional.

- [Configuring Traffic Policing, page 396](#) (required)
- [Attaching the Policy Map to an Interface, page 397](#) (required)
- [Verifying the Configuration, page 399](#) (optional)

Configuring Traffic Policing

Traffic policing can be configured at any level of the policy map hierarchy, that is, at the primary level, secondary level, or the tertiary level.

Prerequisites

Before configuring traffic policing, you must use the MQC to create a policy map. For information about using the MQC to create a policy map, refer to the “Configuring the Modular Quality of Service Command-Line Interface” chapter in the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2.

After creating a policy map, use the following commands to configure traffic policing:

SUMMARY STEPS

1. **enable**
2. **configure** {**terminal** | **memory** | **network**}
3. **policy-map** *policy-name*
4. **class-map** *class-map-name*
5. **police** *bps burst-normal burst-max conform-action action exceed-action action violate-action action*
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	configure { terminal memory network }	Enters global configuration mode.
Step 3	policy-map <i>policy-name</i> 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 396. Enter policy map name.
Step 4	class-map <i>class-map-name</i> Example: Router(config-pmap)# class-map class1	Specifies the name of the class map created when the policy map was created earlier and enters policy-map class configuration mode. • See the “Prerequisites” section on page 396. Enter the class map name.

	Command or Action	Purpose
Step 5	<pre>police bps burst-normal burst-max conform-action action exceed-action action violate-action action</pre> <p>Example: Router(config-pmap-c)# police 8000 1000 1000 conform-action transmit exceed-action drop violate-action drop</p>	Configures traffic policing according to burst sizes and any optional actions specified.
Step 6	<pre>exit</pre> <p>Example: Router(config-pmap-c)# exit</p>	(Optional) Exits the policy-map class configuration mode.

Attaching the Policy Map to an Interface

After the policy map has been created and traffic policing has been configured, the policy map must be attached to an interface. Policy maps can be attached to either the input or output direction of the interface.

Depending on the needs of your network, you may need to attach the policy map to a subinterface, an ATM permanent virtual circuit (PVC), a Frame Relay data-link connection identifier (DLCI), or other type of interface.

To attach the policy map to an interface, use the following commands:

SUMMARY STEPS

1. **enable**
2. **configure {terminal | memory | network}**
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	<pre>enable</pre> <p>Example: Router> enable</p>	Enables higher privilege levels, such as privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<pre>configure {terminal memory network}</pre> <p>Example: Router# configure terminal</p>	Enters global configuration mode.

	Command or Action	Purpose
Step 3	interface <i>type number</i> Example: Router(config-if)# interface s4/0	Configures an interface (or subinterface) type and enters interface configuration mode. <ul style="list-style-type: none"> Enter the interface type number.
Step 4	pvc [<i>name</i>] <i>vpi/vci</i> [<i>ilmi</i> <i>qsaa1</i> <i>smds</i>] 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 virtual circuit (VC) configuration mode (config-if-atm-vc). 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 .
Step 5	service-policy { <i>input</i> <i>output</i> } <i>policy-map-name</i> Example: Router(config-if)# service-policy input policy1	Specifies the name of the policy map to be attached to the input <i>or</i> 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. <ul style="list-style-type: none"> Enter the policy map name.
Step 6	exit Example: Router(config-if)# exit	(Optional) Exits interface configuration mode.

What to Do Next

If you want to configure traffic policing at another level in the policy map hierarchy, repeat the steps in the [“Configuring Traffic Policing”](#) section and the [“Attaching the Policy Map to an Interface”](#) section.

Verifying the Configuration

This task allows you to verify that you created the configuration you intended and that the feature is functioning correctly. To verify the configuration, use the following commands:

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables higher privilege levels, such as privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	show policy-map or show policy-map interface <i>interface-name</i> Example: Router# show policy-map or Example: Router# show policy-map interface s4/0	Displays all configured policy maps. or 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. <ul style="list-style-type: none"> • Enter the interface name.
Step 3	exit Example: Router(config-if)# exit	(Optional) Exits interface configuration mode.

Troubleshooting Tips

The commands in the “[Verifying the 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, the configuration is not correct or the feature is not functioning as expected, do the following:

If the configuration is not the one you intended, complete the following procedures:

- 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.

If the packets are not being matched correctly (for example, the packet counters are not incrementing correctly), complete the following procedures:

- Use the **show policy-map** command and analyze the output of the command.
- Use the **show running-config** command and analyze the output of the command.
- Run the **show policy-map interface** command and analyze the output of the command. Review the the following:
 - If a policy map applies queueing, and the packets are matching the correct class, but you see unexpected results, compare the number of packets to the number of packets matched.
 - If the interface is congested, and you are only seeing a small number of packets matched, check the tuning of the 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 show output of the command.

Configuration Examples for the Modular QoS CLI (MQC) Three-Level Hierarchical Policer

This section provides the following configuration example:

- [Configuring the Modular QoS CLI \(MQC\) Three-Level Hierarchical Policer Example](#)

Configuring the Modular QoS CLI (MQC) Three-Level Hierarchical Policer Example

In the following example, the Modular QoS CLI (MQC) Three-Level Hierarchical Policer has been configured for three classes within three separate policy maps. The three classes, called “c1,” “c2,” and “c3,” respectively, have been configured using the match criteria specified as follows:

```
class-map c1
  match any

class-map c2
  match ip precedence 1 2 3

class-map c3
  match ip precedence 2
```

Next, the classes are configured in three separate policy maps, called “p_all” (the primary-level policy map), “pmatch_123” (the secondary-level policy map), and “pmatch_2” (the tertiary-level policy map), as shown below.

```

policy p_all
  class c1
    police 100000
    service-policy pmatch_123

policy pmatch_123
  class c2
    police 20000
    service-policy pmatch_2

policy pmatch_2
  class c3
    police 8000

```

The primary goal of this configuration is to limit all traffic to 100 kbps. Within this, the secondary goal is make sure that packets with precedence values of 1, 2, or 3 do not exceed 20 kbps and that packets with precedence value of 2 never exceed 8 kbps.

To verify that the classes have been configured correctly and to confirm the results of the traffic policing configuration in the policy maps, the **show policy-map** command and the **show policy-map interface** command can be used, as shown in the following sections.

The following sample output of the **show policy-map** command verifies the configuration of the classes in the policy maps:

```

Router# show policy map

Policy Map p_all
Class c1
  police cir 100000 bc 3000
    conform-action transmit
    exceed-action drop
  service-policy pmatch_123

Policy Map pmatch_123
Class c2
  police cir 20000 bc 1500
    conform-action transmit
    exceed-action drop
  service-policy pmatch_2

Policy Map pmatch_2
Class c3
  police cir 8000 bc 1500
    conform-action transmit
    exceed-action drop

```

The following sample output of the **show policy-map interface** command confirms the results of this configuration on the attached interface:

```

Router# show policy-map interface Ethernet3/1

Ethernet3/1

Service-policy output:p_all

Class-map:c1 (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps

```

```

Match:any
police:
  cir 100000 bps, bc 3000 bytes
  conformed 0 packets, 0 bytes; actions:
    transmit
  exceeded 0 packets, 0 bytes; actions:
    drop
  conformed 0 bps, exceed 0 bps,

Service-policy :pmatch_123

Class-map:c2 (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match:ip precedence 1 2 3
  police:
    cir 20000 bps, bc 1500 bytes
    conformed 0 packets, 0 bytes; actions:
      transmit
    exceeded 0 packets, 0 bytes; actions:
      drop
    conformed 0 bps, exceed 0 bps,

Service-policy :pmatch_2

Class-map:c3 (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match:ip precedence 2
  police:
    cir 8000 bps, bc 1500 bytes
    conformed 0 packets, 0 bytes; actions:
      transmit
    exceeded 0 packets, 0 bytes; actions:
      drop
    conformed 0 bps, exceed 0 bps,

Class-map:class-default (match-any)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match:any

Class-map:class-default (match-any)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match:any

Class-map:class-default (match-any)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match:any

```


Additional References

The following sections provide additional references related to the Modular QoS CLI (MQC) Three-Level Hierarchical Policer:

- [Related Documents, page 403](#)
- [Standards, page 403](#)
- [MIBs, page 404](#)
- [RFCs, page 404](#)
- [Technical Assistance, page 404](#)

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	<i>Cisco IOS Quality of Service Solutions Command Reference</i> , Release 12.2
Additional information about configuring traffic policing	“Policing and Shaping” section of the <i>Cisco IOS Quality of Service Solutions Configuration Guide</i> , Release 12.2
Modular QoS Command-Line Interface (CLI) (MQC)	“Modular Quality of Service Command-Line Interface” section of the <i>Cisco IOS Quality of Service Solutions Configuration Guide</i> , Release 12.2
Two-rate traffic policing	<i>Two-Rate Policer</i> , Cisco IOS Release 12.2(4)T feature module
Traffic policing using multiple policer actions	<i>Policer Enhancements — Multiple Actions</i> , Cisco IOS Release 12.2(8)T feature module
Percentage-based traffic policing and shaping	<i>Percentage-Based Policing and Shaping</i> , Cisco IOS Release 12.2(13)T feature module
Frame Relay configuration information and information about DLCIs	<i>Cisco IOS Wide-Area Networking Configuration Guide</i> , Release 12.2
Frame Relay commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	<i>Cisco IOS Wide-Area Networking Command Reference</i> , Release 12.2

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 ¹	MIBs Link
<ul style="list-style-type: none"> CISCO-CLASS-BASED-QOS-CAPABILITY-MIB CISCO-CLASS-BASED-QOS-MIB 	<p>To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:</p> <p>http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml</p>

1. Not all supported MIBs are listed.

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

RFCs ¹	Title
RFC 2697	<i>A Single Rate Three Color Marker</i>
RFC 2698	<i>A Two Rate Three Color Marker</i>

1. Not all supported RFCs are listed.

Technical Assistance

Description	Link
<p>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.</p>	<p>http://www.cisco.com/public/support/tac/home.shtml</p>

Command Reference

This feature uses no new or modified commands. To see the command pages for the commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at <http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm>.



ATM Policing by Service Category for SVC/SoftPVC

Feature History

Release	Modification
12.2(4)B	This feature was introduced on the Cisco 6400 NSP.
12.2(13)T	This command was integrated into Cisco IOS Release 12.2(13)T.

This document describes the ATM Policing by Service Category for SVC/SoftPVC feature in Cisco IOS Release 12.2(13)T and includes the following sections:

- [Feature Overview, page 407](#)
- [Supported Platforms, page 408](#)
- [Supported Standards, MIBs, and RFCs, page 408](#)
- [Configuration Tasks, page 409](#)
- [Monitoring and Maintaining ATM Policing by Service Category for SVC/SoftPVC, page 410](#)
- [Configuration Examples, page 411](#)
- [Command Reference, page 412](#)
- [Glossary, page 413](#)

Feature Overview

When configured, an ATM switch at the network side of a user-to-network (UNI) interface polices the flow of cells in the forward (into the network) direction of a virtual connection. These traffic policing mechanisms are known as usage parameter control (UPC). With UPC, the switch determines whether received cells comply with the negotiated traffic management values and takes one of the following actions on violating cells:

- Pass the cell without changing the cell loss priority (CLP) bit in the cell header.
- Tag the cell with a CLP bit value of 1.
- Drop (discard) the cell.

The ATM Policing by Service Category for SVC/SoftPVC feature enables you to specify which traffic to police, based on service category, on switched virtual circuits (SVCs) or terminating VCs on the destination end of a soft VC.

For more information on UPC, see the “Traffic and Resource Management” chapter in the *Guide to ATM Technology*.

Benefits

This feature enables you to select which and how traffic is affected by UPC. For example, you can configure your switch to pass all UBR traffic, but tag all other traffic types.

Related Features and Technologies

- Intelligent early packet discard (EPD)
- Intelligent partial (tail) packet discard

Related Documents

- *ATM Switch Router Software Configuration Guide*
- *ATM and Layer 3 Switch Router Command Reference*
- *Guide to ATM Technology*
- *ATM Forum UNI 3.1 Specification*

Supported Platforms

This feature is supported on the node switch processor (NSP) of the Cisco 6400 carrier-class broadband aggregator.

Supported Standards, MIBs, and RFCs

Standards

None

MIBs

CISCO-ATM-IF-MIB.my—New objects were created for per-service category SVC UPC intent.

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

<http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml>

RFCs

None

Configuration Tasks

See the following sections for configuration tasks for the ATM Policing by Service Category for SVC/SoftPVC feature. Each task in the list is identified as either required or optional:

- [Configuring ATM Policing by Service Category for SVC/SoftPVC](#) (Required)
- [Verifying ATM Policing by Service Category for SVC/SoftPVC](#) (Optional)

Configuring ATM Policing by Service Category for SVC/SoftPVC

To configure the ATM Policing by Service Category for SVC/SoftPVC feature, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Switch(config)# interface atm slot/subslot/port	Selects the ATM interface.
Step 2	Switch(config-if)# atm svc-upc-intent [{abr cbr vbr-rt vbr-nrt ubr}] {tag pass drop} (Repeat this step for each service category and UPC mode combination.)	Specifies the UPC mode. If no service category is specified, then the UPC mode configuration is applied to all traffic types.

Verifying ATM Policing by Service Category for SVC/SoftPVC

- Step 1** Enter the **show atm vc** or **show atm vp EXEC** command to display the UPC mode for a particular VC or VP.

```
Switch# show atm vc int atm 0/0/1 2 120
```

```
Interface:ATM0/0/1, Type:oc3suni
VPI = 2 VCI = 120
Status:DOWN
Time-since-last-status-change:1w1d
Connection-type:PVC
Cast-type:point-to-multipoint-leaf
Packet-discard-option:disabled
Usage-Parameter-Control (UPC):pass
Wrr weight:2
Number of OAM-configured connections:0
OAM-configuration:disabled
OAM-states: Not-applicable
Cross-connect-interface:ATM0/0/1, Type:oc3suni
...
```

- Step 2** Enter the **show atm interface EXEC** command. If the UPC mode is not the same for all service categories, the “Svc Upc Intent” field displays “by sc.”

```
Switch# show atm interface atm 8/0/1

Interface:      ATM8/0/1          Port-type:      oc3suni
IF Status:     UP              Admin Status:   up
Auto-config:   enabled         AutoCfgState:   completed
IF-Side:       Network         IF-type:        NNI
Uni-type:      not applicable  Uni-version:    not applicable
Max-VPI-bits:  8              Max-VCI-bits:   14
Max-VP:        255          Max-VC:         16383
ConfMaxSvpcVpi:255      CurrMaxSvpcVpi:255
ConfMaxSvccVpi:255      CurrMaxSvccVpi:255
ConfMinSvccVci:35       CurrMinSvccVci:35
Svc Upc Intent:by sc      Signalling:     Enabled
ATM Address for Soft VC:47.0091.8100.0000.0002.b9ae.9301.4000.0c84.0010.00
Configured virtual links:
  PVCLs  SoftVCLs  SVCLs  TVCLs  PVPLs  SoftVPLs  SVPLs  Total-Cfgd  Inst-Conns
    3      4      0      0      1      0      0      8      7
Logical ports (VP-tunnels):  0
Input cells:  3036674          Output cells:  3036816
5 minute input rate:         0 bits/sec,    0 cells/sec
5 minute output rate:        0 bits/sec,    0 cells/sec
Input AAL5 pkts:1982638, Output AAL5 pkts:1982687, AAL5 crc errors:0
```

Troubleshooting Tips

If a VC is not configured with the appropriate UPC mode, make sure that the VC was set up after the **atm svc-upc-intent** command was configured. Changes to the UPC mode take affect after the VC is torn down and set up again.

Monitoring and Maintaining ATM Policing by Service Category for SVC/SoftPVC

Use the commands listed below to monitor and maintain ATM Policing by Service Category for SVC/SoftPVC:

Command	Purpose
Switch# show atm interface	Displays ATM-specific information about an ATM interface.
Switch# show controllers atm slot/subslot/port	Displays information about a physical port device. Includes dropped (or discarded) cells.
Switch# show atm vc [interface atm slot/subslot/port]	Displays the configured UPC action and intelligent packet discard mechanisms, as well as the number of cells discarded due to UPC violations.

Example: Monitoring and Maintaining ATM Policing by Service Category for SVC/SoftPVC

```
Switch# show atm vc interface atm 3/0/1.51 51 16

Interface: ATM3/0/1.51, Type: oc3suni
VPI = 51 VCI = 16
Status: DOWN
Time-since-last-status-change: 2w0d
Connection-type: PVC
Cast-type: point-to-point
Packet-discard-option: enabled
Usage-Parameter-Control (UPC): pass
Wrr weight: 32
Number of OAM-configured connections: 0
OAM-configuration: disabled
OAM-states: Not-applicable
Cross-connect-interface: ATM2/0/0, Type: ATM Swi/Proc
Cross-connect-VPI = 0
Cross-connect-VCI = 73
Cross-connect-UPC: pass
Cross-connect OAM-configuration: disabled
Cross-connect OAM-state: Not-applicable
Encapsulation: AAL5ILMI
Threshold Group: 6, Cells queued: 0
Rx cells: 0, Tx cells: 0
Tx Clp0:0, Tx Clp1: 0
Rx Clp0:0, Rx Clp1: 0
Rx Upc Violations:0, Rx cell drops:0
Rx pkts:0, Rx pkt drops:0
Rx connection-traffic-table-index: 6
Rx service-category: UBR (Unspecified Bit Rate)
Rx pcr-clp01: 424
Rx scr-clp01: none
Rx mcr-clp01: none
Rx cdvt: 1024 (from default for interface)
Rx mbs: none
Tx connection-traffic-table-index: 6
Tx service-category: UBR (Unspecified Bit Rate)
Tx pcr-clp01: 424
Tx scr-clp01: none
Tx mcr-clp01: none
Tx cdvt: none
Tx mbs: none
No AAL5 connection registered
```

Configuration Examples

This section provides the following configuration example:

- [Non-UBR Traffic Policing](#)

Non-UBR Traffic Policing

In the following example, the UBR traffic on ATM 3/0/0 is passed while all other traffic is policed:

```
Switch(config)# interface atm 3/0/0
Switch(config-if)# atm svc-upc-intentubr pass
Switch(config-if)# atm svc-upc-intentcbr tag
Switch(config-if)# atm svc-upc-intentvbr-rt tag
Switch(config-if)# atm svc-upc-intentvbr-nrt tag
Switch(config-if)# atm svc-upc-intentabr drop
```

Command Reference

The following modified command is pertinent to this feature. To see the command pages for this command and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at <http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm>.

- **atm svc-upc-intent**

Glossary

ABR—available bit rate. QoS class defined by the ATM Forum for ATM networks. ABR is used for connections that do not require timing relationships between source and destination. ABR provides no guarantees in terms of cell loss or delay, providing only best-effort service. Traffic sources adjust their transmission rate in response to information they receive describing the status of the network and its capability to successfully deliver data. Compare with CBR, UBR, and VBR.

CBR—constant bit rate. QoS class defined by the ATM Forum for ATM networks. CBR is used for connections that depend on precise clocking to ensure undistorted delivery. Compare with ABR, UBR, and VBR.

CLP—cell loss priority. Field in the ATM cell header that determines the probability of a cell being dropped if the network becomes congested. Cells with CLP = 0 are insured traffic, which is unlikely to be dropped. Cells with CLP = 1 are best-effort traffic, which might be dropped in congested conditions to free up resources to handle insured traffic.

PVC—permanent virtual circuit (or connection). 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. Compare with SVC. See also virtual circuit.

soft PVC—A PVC-SVC hybrid in which only the two terminating virtual connection links (VCLs) at either end are permanent and the rest of the VCLs are switched (SVC). Like the PVC, a soft PVC is permanent and the called party cannot drop the connection. Like the SVC, a soft PVC is automatically rerouted if a switch or link in the path fails.

SVC—switched virtual circuit. Virtual circuit that is dynamically established on demand and is torn down when transmission is complete. SVCs are used in situations where data transmission is sporadic. See also virtual circuit. Called a switched virtual connection in ATM terminology. Compare with PVC.

tagged traffic—ATM cells that have their CLP bit set to 1. If the network is congested, tagged traffic can be dropped to ensure the delivery of higher-priority traffic. Sometimes called DE traffic. See also CLP.

traffic policing—Process used to measure the actual traffic flow across a given connection and compare it to the total admissible traffic flow for that connection. Traffic outside of the agreed upon flow can be tagged (where the CLP bit is set to 1) and can be discarded en route if congestion develops. Traffic policing is used in ATM, Frame Relay, and other types of networks. Also known as admission control, permit processing, rate enforcement, and UPC. See also tagged traffic.

UBR—unspecified bit rate. QoS class defined by the ATM Forum for ATM networks. UBR allows any amount of data up to a specified maximum to be sent across the network but there are no guarantees in terms of cell loss rate and delay. Compare with ABR, CBR, and VBR.

UPC—usage parameter control. See traffic policing.

VBR—variable bit rate. QoS class defined by the ATM Forum for ATM networks. VBR is subdivided into a real time (RT) class and non-real time (NRT) class. VBR (RT) is used for connections in which there is a fixed timing relationship between samples. VBR (NRT) is used for connections in which there is no fixed timing relationship between samples but that still need a guaranteed QoS. Compare with ABR, CBR, and UBR.

virtual circuit—Logical circuit created to ensure reliable communication between two network devices. A virtual circuit is defined by a VPI/VCI pair, and can be either permanent (PVC) or switched (SVC). Virtual circuits are used in Frame Relay and X.25. In ATM, a virtual circuit is called a virtual channel. Sometimes abbreviated VC.



Modular QoS CLI (MQC) Unconditional Packet Discard

Feature History

Release	Modification
12.2(13)T	This feature was introduced.

Supported Platforms

For platforms supported in Cisco IOS Release 12.2(13)T, consult Cisco Feature Navigator.

This document describes the Modular QoS CLI (MQC) Unconditional Packet Discard feature in Cisco IOS Release 12.2(13)T. It includes the following sections:

- [Feature Overview, page 415](#)
- [Supported Platforms, page 416](#)
- [Supported Standards, MIBs, and RFCs, page 417](#)
- [Configuration Tasks, page 418](#)
- [Configuration Examples, page 420](#)
- [Command Reference, page 421](#)

Feature Overview

The Modular QoS CLI (MQC) Unconditional Packet Discard feature allows customers to classify traffic matching certain criteria and then configure the system to unconditionally discard any packets matching that criteria. The Modular QoS CLI (MQC) Unconditional Packet Discard feature is configured using the Modular Quality of Service Command-Line Interface (MQC) feature. For more information about the MQC feature, refer to the *Quality of Service Cisco IOS Solutions Configuration Guide*, Release 12.2.

Packets are unconditionally discarded by using the new **drop** command within the MQC. For more information about the **drop** command, see the “[Command Reference](#)” section later in this document.

Benefits

Enhanced System Utilization

This feature allows you to discard (drop), without any further system processing, the packets of a particular class. This function is very useful when you want to discard all the packets for nonessential applications (for instance, Internet browsing applications or unauthorized video applications) and allocate system resources to more essential applications. This feature allows the user to discard those nonessential packets and simultaneously obtain the bit and drop rate statistics for that particular class and the traffic within that class. The statistics are gathered through the CISCO-CLASS-BASED-QOS-MIB.

Restrictions

Packets are unconditionally discarded by configuring the drop action inside a traffic class (inside of a policy map). This drop action is accomplished with the new **drop** command. Note the following restrictions for configuring the drop action within a traffic class:

- The discarding action is the only action that can be configured in a traffic class. That is, no other actions can be configured in the traffic class.
- When a traffic class is configured with the **drop** command, a “child” (nested) policy cannot be configured for this specific traffic class through the **service policy** command.
- The discarding action cannot be configured for the default class known as the class-default class.

Related Features and Technologies

- Modular quality of service command-line interface (MQC)

Related Documents

- *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2
- *Cisco IOS Quality of Service Solutions Command Reference*, Release 12.2

Supported Platforms

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that are supported on specific platforms. To obtain updated information about 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 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. In the release section, you can compare releases side by side to display both the features unique to each software release and the features that releases have 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 at <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, refer to the online release notes or, if supported, Cisco Feature Navigator.

Supported Standards, MIBs, and RFCs

Standards

None

MIBs

- 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://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

None

Configuration Tasks

See the following sections for configuration tasks for the Modular QoS CLI (MQC) Unconditional Packet Discard feature. Each task in the list is identified as either required or optional.

- [Configuring the Class Map](#) (required)
- [Creating a Policy Map](#) (required)
- [Attaching the Policy Map to an Interface or a VC](#) (required)
- [Verifying the Discard Action Configuration in the Traffic Class](#) (optional)

Configuring the Class Map

To configure a class map to discard packets belonging to a specific class, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# class-map <i>class-map-name</i>	Specifies the name of the class map to be created. If match-all or match-any is not specified, traffic must match all the match criteria to be classified as part of the traffic class. Enters class-map configuration mode.
Step 2	Router(config-cmap)# match access-group { <i>access-group</i> name <i>access-group-name</i> }	Specifies that traffic matching the specified access group will be placed in the map class created above. This command provides just an example of the match criterion you can specify. For more information about the additional match criteria available, refer to the “Configuring the Quality of Service Command-Line Interface” chapter of the <i>Cisco IOS Quality of Service Solutions Configuration Guide</i> , Release 12.2.
Step 3	Router(config-cmap)# exit	Exits from the configuration mode.

Creating a Policy Map

To create a policy map (also sometimes referred to as a service policy or a traffic policy), use the following commands beginning in global configuration mode.

A policy map can be created using the MQC feature. For more information about creating a policy map using the MQC feature, refer to the “Configuring the Modular Quality of Service Command-Line Interface” chapter of the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2.

	Command	Purpose
Step 1	Router (config)# policy-map <i>policy-name</i>	Specifies the name of the policy map to be created. Enters policy-map configuration mode.
Step 2	Router (config-pmap)# class <i>class-name</i>	Specifies the name of the traffic class configured earlier in the “Configuring the Class Map” section above. This traffic class is used to classify traffic to the policy map. Enters policy-map class configuration mode.
Step 3	Router (config-pmap)# drop	Discards the packets in the specified traffic class.
Step 4	Router (config-cmap)# exit	Exits policy-map configuration mode.

Attaching the Policy Map to an Interface or a VC

To attach the policy map to an interface or a virtual circuit (VC), use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router (config)# interface <i>type number</i> [<i>name-tag</i>]	Configures the interface type and enters interface configuration mode.
Step 2	Router (config-if)# pvc [<i>name</i>] <i>vpi/vci</i> [ilmi qsaal smcS]	(Optional) Creates or assigns a name to an ATM permanent virtual circuit (PVC), and specifies the encapsulation type on an ATM PVC. Enters ATM VC configuration mode (config-if-atm-vc). This step is required only if you are attaching the policy map to an ATM PVC.
Step 3	Router (config-if)# service-policy input <i>policy-map-name</i> or Router (config-if-atm-vc)# service-policy output <i>policy-map-name</i>	Specifies the name of the policy map to be attached to the input or output direction of an interface or VC. The policy map evaluates all traffic entering or leaving that interface or VC.
Step 4	Router (config-if)# exit	Exits interface configuration mode.

Verifying the Discard Action Configuration in the Traffic Class

To verify that the discard action has been configured in the traffic class and the policy map (and to display the number of packets discarded), use the following commands in EXEC or privileged EXEC mode, as needed:

Command	Purpose
Router# show policy-map	Displays the configuration of all classes for a specified service policy map or all classes for all existing policy maps.
Router# show policy-map interface interface-name	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.

Configuration Examples

This section provides the following configuration examples:

- [Configuring the Discard Action Configuration in a Traffic Class Example](#)
- [Verifying the Discard Action Configuration in the Policy Map Example](#)

Configuring the Discard Action Configuration in a Traffic Class Example

In the following sample configuration, a traffic class called “class1” has been created and configured for use in a policy-map called “policy1.” The policy-map policy1 is attached to an output serial interface 2/0. All packets matching access-group 101 are placed in a class called “c1.” Packets belonging to this class are discarded.

```
Router(config)# class-map class1
Router(config-cmap)# match access-group 101
Router(config-cmap)# policy-map policy1
Router(config-pmap)# class c1
Router(config-pmap-c)# drop
Router(config-pmap-c)# interface s2/0
Router(config-if)# service-policy output policy1
Router(config-if)# exit
```

The following sample output of the **show policy-map** command displays the contents of the policy map called “policy1.” All the packets belonging to the class called “c1” are discarded.

```
Router# show policy-map policy1

Policy Map policy1
  Class c1
    drop
```

Verifying the Discard Action Configuration in the Policy Map Example

The following sample output of the **show policy-map interface** command displays the statistics for the Serial2/0 interface, to which a policy map called “policy1” is attached. The discard action has been specified for all the packets belonging to a class called “c1.” In this example, 32000 bps of traffic is sent (“offered”) to the class and all of them are dropped. Therefore, the drop rate shows 32000 bps.

```
Router# show policy-map interface Serial2/0

Serial2/0

Service-policy output: policy1

Class-map: c1 (match-all)
  10184 packets, 1056436 bytes
  5 minute offered rate 32000 bps, drop rate 32000 bps
  Match: ip precedence 0
  drop
```

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at <http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm>.

New Commands

- **drop**

Modified Commands

- **show policy-map**
- **show policy-map interface**



Control Plane Policing

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. Thus, the control plane (CP) can help maintain packet forwarding and protocol states despite an attack or heavy traffic load on the router or switch.

Feature History for the Control Plane Policing Feature

Release	Modification
12.2(18)S	This feature was introduced.
12.3(4)T	Control Plane Policing was integrated into Cisco IOS Release 12.3(4)T, and the output rate-limiting (silent mode operation) feature was added.
12.3(7)T	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.
12.0(29)S	The Control Plane Policing feature was integrated into Cisco IOS Release 12.0(29)S.
12.2(18)SXD1	The Control Plane Policing feature was integrated into Cisco IOS Release 12.2(18)SXD1.
12.0(30)S	Support for distributed control plane services on the Cisco 12000 series Internet router was added.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- [Prerequisites for Control Plane Policing, page 424](#)
- [Restrictions for Control Plane Policing, page 424](#)

- [Information About Control Plane Policing](#), page 425
- [How to Use the Control Plane Policing Feature](#), page 430
- [Configuration Examples for Control Plane Policing](#), page 437
- [Additional References](#), page 439
- [Command Reference](#), page 440

Prerequisites for Control Plane Policing

- Understanding the concepts and general configuration procedure (class map and policy map) for applying quality-of-service (QoS) policies on a router

For information about Cisco IOS QoS and the procedure for configuring QoS in your network using the modular QoS command-line interface (MQC), refer to [Cisco IOS Quality of Service Solutions Configuration Guide](#), Release 12.2

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, and 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 [Output Rate-Limiting and Silent Mode Operation](#), page 430.

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.

Modular QoS Restrictions

The Control Plane Policing feature requires the modular QoS command-line interface (CLI) (MQC) to configure packet classification and policing. Thus, restrictions that apply to MQC also apply to control plane policing. Also, only two MQC actions are supported in policy maps—**police** and **drop**.

Features that require network-based application recognition (NBAR) classification may not work well at the control plane level. Only the following classification (match) criteria are supported: standard and extended IP access lists and the **match ip dscp** command, the **match ip precedence** command, and the **match protocol arp** command.

The **match protocol arp** command is not supported in Cisco IOS 12.2SX releases.

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, Hardware Control Plane Interface for Control Plane Policing has the following restrictions:

- Supported only with Supervisor Engine 720. Not supported with the Supervisor Engine 2.
- Does not support CoPP output rate limiting (policing).
- Does not support the CoPP silent operation mode.
- Release 12.2(18)SXD1 automatically installs the CoPP service policy on all DFC-equipped switching modules.

Information About Control Plane Policing

To configure the Control Plane Policing feature, you should understand the following concepts:

- [Benefits of Control Plane Policing, page 425](#)
- [Terms to Understand, page 425](#)
- [Control Plane Security and Packet QoS Overview, page 427](#)
- [Aggregate Control Plane Services, page 428](#)
- [Output Rate-Limiting and Silent Mode Operation, page 430](#)

Benefits of Control Plane Policing

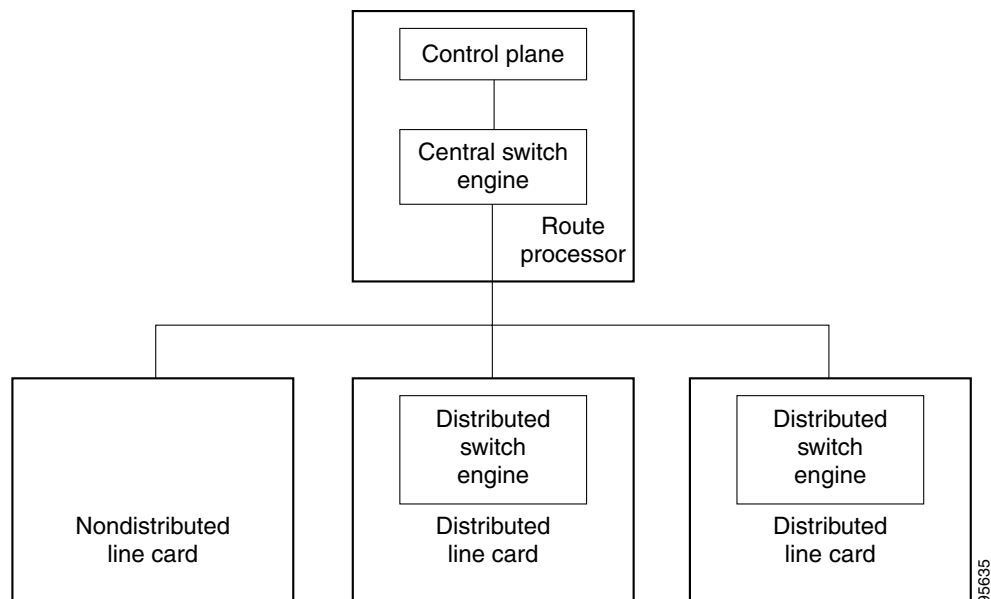
Configuring the Control Plane Policing feature on your Cisco router or switch provides the following benefits:

- 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 15](#) illustrates how control plane policing works.

Figure 15 *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.

- 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 12.0(30)S and later 12.0S releases.

Control Plane Security and Packet QoS Overview

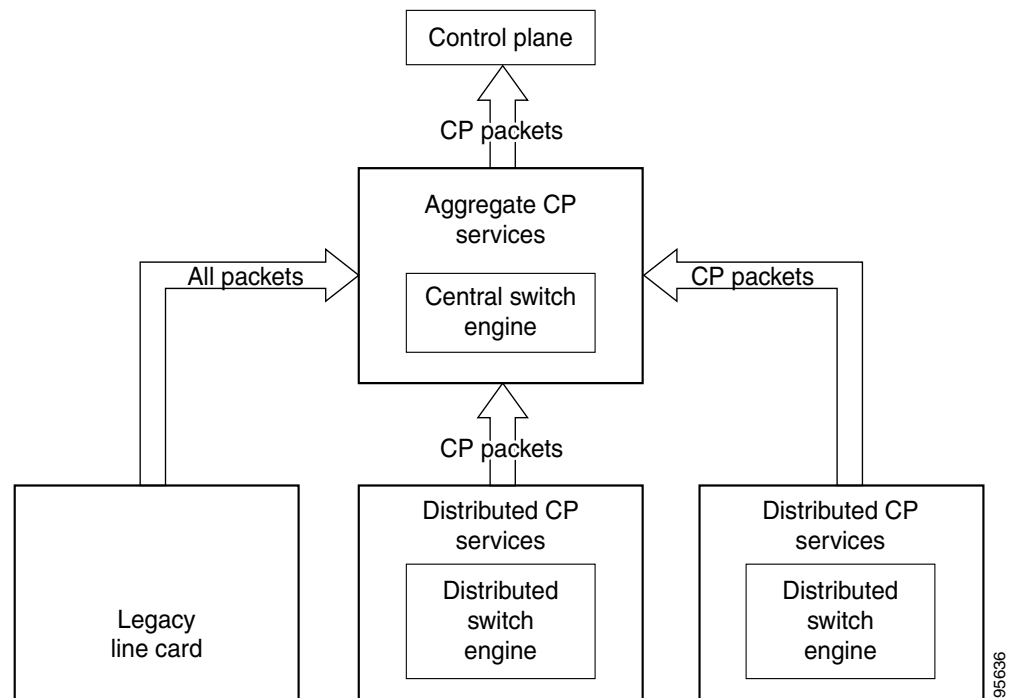
To protect the CP on a router from DoS attacks and to provide packet QoS, the Control Plane Policing feature treats the CP as a separate entity with its own ingress (input) and egress (output) ports, which are like ports on a router and switch. Because the Control Plane Policing feature treats the CP as a separate entity, a set of rules can be established and associated with the ingress and egress port 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 from 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 and a routing decision on the input path have been made. As shown in [Figure 16](#), 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 [Aggregate Control Plane Services](#), page 428)
- 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 [Distributed Control Plane Services](#), page 428)

Figure 16 *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 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 linecard 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:

- Defined for a single input interface, such as the CP, and represents 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 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 or not 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.
- 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 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.
- Modular QoS does not prevent a single 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 amount of CP packets received from all line cards on a router may exceed aggregate CP levels.

When Distributed CP Services Are Necessary

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 modular QoS command-line interface 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 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 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 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 is 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 from 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 the Control Plane Policing Feature

This section documents the following procedures:

- [Defining Aggregate Control Plane Services, page 431](#)
- [Defining Distributed Control Plane Services, page 432](#)

- [Verifying Aggregate CP Services, page 433](#)
- [Verifying Distributed CP Services, page 435](#)

Defining Aggregate Control Plane Services

Perform this task to configure aggregate CP services, such as packet rate control and silent packet discard, for the active route processor.

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 map and policy map for control plane traffic.

For information about how to classify traffic and create a QoS policy, refer to the “[Modular Quality of Service Command-Line Interface](#)” chapter in the *Cisco IOS Quality of Service Solutions Configuration Guide*.

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**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	<code>control-plane</code> Example: Router(config)# control-plane	Enters control-plane configuration mode to attach a QoS policy that manages CP traffic.
Step 4	<code>service-policy {input output} policy-map-name</code> Example: Router(config-cp)# service-policy input control-plane-policy	Attaches a QoS service policy to the control plane. <ul style="list-style-type: none"> 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. <i>policy-map-name</i>—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.

Defining Distributed Control Plane Services

Perform this task 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.

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, refer to the “[Modular Quality of Service Command-Line Interface](#)” chapter in the *Cisco IOS Quality of Service Solutions Configuration Guide*.

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, Supervisor Engine 720 automatically installs the service policy on all DFC-equipped switching modules.

SUMMARY STEPS

- enable**
- configure terminal**
- control-plane slot** *slot-number*
- service-policy input** *policy-map-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>enable</p> <p>Example: Router> enable</p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	<p>configure terminal</p> <p>Example: Router# configure terminal</p>	<p>Enters global configuration mode.</p>
Step 3	<p>control-plane slot <i>slot-number</i></p> <p>Example: Router(config)# control-plane slot 3 Router(config)# control-plane slot</p>	<p>Enters control-plane configuration mode to attach a QoS policy to manage CP traffic on the line card in the specified slot.</p>
Step 4	<p>service-policy input <i>policy-map-name</i></p> <p>Example: Router(config-cp)# service-policy input control-plane-policy</p>	<p>Attaches a QoS service policy to filter and manage CP traffic on a specified line card before the aggregate CP policy is applied.</p> <ul style="list-style-type: none"> input—Applies the specified service policy using the distributed switch engine to CP packets received from all interfaces on the line card. <i>policy-map-name</i>—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. <p>Note The service-policy output <i>policy-map-name</i> command is not supported for applying a QoS policy for distributed control plane services.</p>

Verifying Aggregate CP Services

To display information about the service policy attached to the control plane for aggregate CP services, perform the following optional steps.

SUMMARY STEPS

- enable**
- show policy-map control-plane** [**all**] [**input** [**class** *class-name*] | **output** [**class** *class-name*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>enable</p> <p>Example: Router> enable</p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	<p>show policy-map control-plane [all] [input [class <i>class-name</i>] output [class <i>class-name</i>]]</p> <p>Example: Router# show policy-map control-plane all</p>	<p>Displays information about the control plane.</p> <ul style="list-style-type: none"> all—Service policy information about all QoS policies used in aggregate and distributed CP services. input—Statistics for the attached input policy will be displayed. output—Statistics for the attached output policy will be displayed. class <i>class-name</i>—Name of the class whose configuration and statistics are to be displayed.

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 CP Services

To display information about the service policy attached to the control plane to perform distributed CP services, perform the following optional steps.

SUMMARY STEPS

1. **enable**
2. **show policy-map control-plane** [**all** | **slot** *slot-number*] [**input** [**class** *class-name*] | **output** [**class** *class-name*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>enable</p> <p>Example: Router> enable</p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<p>show policy-map control-plane [all] [slot <i>slot-number</i>] [input [class <i>class-name</i>] output [class <i>class-name</i>]]</p> <p>Example: Router# show policy-map control-plane all</p>	<p>Displays information about the service policy used to apply distributed CP services on the router.</p> <ul style="list-style-type: none"> • all—Service policy information about all QoS policies used in aggregate and distributed CP services. • slot <i>slot-number</i>—Service policy information about the QoS policy used for performing distributed CP services on the specified line card. • input—Statistics for the attached input policy will be displayed. • output—Statistics for the attached output policy will be displayed. • class <i>class-name</i>—Name of the traffic class whose configuration and statistics are to be displayed.

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)
```

**Note**

The **match protocol arp** command is not supported in Cisco IOS 12.2SX releases.

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 Rate Limiting \(Input\) Telnet Traffic: Example, page 437](#)
- [Configuring Rate Limiting \(Output\) Telnet Traffic: Example, page 437](#)
- [Configuring Rate Limiting \(Input\) for Distributed CP Traffic: Example, page 438](#)

Configuring Rate Limiting (Input) Telnet Traffic: Example

The following example shows how to configure trusted hosts with source addresses 10.1.1.1 and 10.1.1.2 to forward Telnet packets to the control plane without constraint, while allowing all remaining Telnet packets 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)# exit
```

Configuring Rate Limiting (Output) Telnet Traffic: Example

The following configuration shows how to configure trusted networks with source addresses 10.1.1.3 and 10.1.1.4 to receive Internet Control Management Protocol (ICMP) port-unreachable responses without constraint, while allowing all remaining ICMP port-unreachable responses to be dropped:

```
! Allow 10.1.1.3 trusted network traffic.
Router(config)# access-list 141 deny icmp 10.1.1.3 10.255.255.255 any port-unreachable
! Allow 10.1.1.4 trusted network traffic.
Router(config)# access-list 141 deny icmp 10.1.1.4 10.255.255.255 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)# exit

```

Configuring Rate Limiting (Input) for Distributed CP Traffic: Example

The following example shows how to attach a QoS policy to perform distributed CP services on packets destined for the CP from the interfaces on the line card in slot 1.

As in the previous example, trusted hosts are configured with source addresses 10.1.1.1 and 10.1.1.2 to forward Telnet packets to the control plane without constraint, while allowing all remaining Telnet packets that enter through slot 1 to be policed at the specified rate:

```

! 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 slot 1
Router(config-cp)# service-policy input control-plane-in
Router(config-cp)# exit

```

Additional References

The following sections provide references related to Control Plane Policing.

Related Documents

Related Topic	Document Title
QoS information and configuration tasks	<i>Cisco IOS Quality of Service Solutions Configuration Guide</i>
Additional QoS commands	<i>Cisco IOS Quality of Service Solutions Command Reference, Release 12.3 T</i>

Standards

Standards	Title
No new or modified standards are supported by this feature.	—

MIBs

MIBs	MIBs Link
<ul style="list-style-type: none"> CISCO-CLASS-BASED-QOS-MIB <p>Note Supported only in Cisco IOS Release 12.3(7)T.</p>	<p>To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator, found at the following URL:</p> <p>http://www.cisco.com/go/mibs</p>

RFCs

RFCs	Title
None	—

Technical Assistance

Description	Link
The Cisco Technical Support website, 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/techsupport

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at <http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm>.

- **control-plane**
- **service-policy (control-plane)**
- **show policy-map control-plane**



Packet Flow

This part consists of the following:

- [Regulating Packet Flow Roadmap](#)
- [Regulating Packet Flow Using Traffic Shaping](#)
- [Regulating Packet Flow on a Per-Class Basis — Using Class-Based Traffic Shaping](#)
- [Regulating Packet Flow on a Per-Interface Basis — Using Generic Traffic Shaping](#)



Regulating Packet Flow Roadmap

This roadmap lists the features documented in the *Cisco IOS Quality of Service Solutions* configuration guide and maps them to the modules in which they appear.

Roadmap History

This roadmap was first published on May 2, 2005, and last updated on May 2, 2005.

Feature and Release Support

[Table 28](#) 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](#)

Only features that were introduced or modified in Cisco IOS Release 12.2(1) or later appear in the table. *Not all features may be supported in your Cisco IOS software release.*

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 Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.



Note

[Table 28](#) 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 28 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, page 445](#)
- [Where to Go Next, page 450](#)
- [Additional References, page 451](#)

Information About Traffic Shaping

Before configuring any of the Cisco traffic shaping mechanisms, you should understand the following concepts:

- [Benefits of Shaping Traffic on a Network, page 446](#)
- [Cisco Traffic Shaping Mechanisms, page 446](#)
- [Token Bucket and Traffic Shaping, page 447](#)
- [Traffic Shaping and Rate of Transfer, page 448](#)
- [How Traffic Shaping Regulates Traffic, page 448](#)
- [Traffic Shaping versus Traffic Policing, page 450](#)

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 447).

Table 29 lists the differences between traffic shaping mechanisms.

Table 29 Differences Between Traffic Shaping Mechanisms

	Traffic Shaping Mechanism		
	Class-Based	GTS	FRTS
Command-Line Interface	<ul style="list-style-type: none"> Applies configuration on a per-class basis 	<ul style="list-style-type: none"> Applies configuration on a per interface or subinterface basis traffic group command supported 	<ul style="list-style-type: none"> Classes of parameters Applies configuration to all virtual circuits (VCs) on an interface through inheritance mechanism No traffic group group
Queues Supported	<ul style="list-style-type: none"> Class-based WFQ (CBWFQ) 	<ul style="list-style-type: none"> Weighted Fair Queueing (WFQ) per interface or subinterface 	<ul style="list-style-type: none"> 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	Cisco IOS Wide-Area Networking Configuration Guide , Release 12.3

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 (T_c). 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:

$$\text{mean rate} = \text{burst size} / \text{time interval}$$

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.

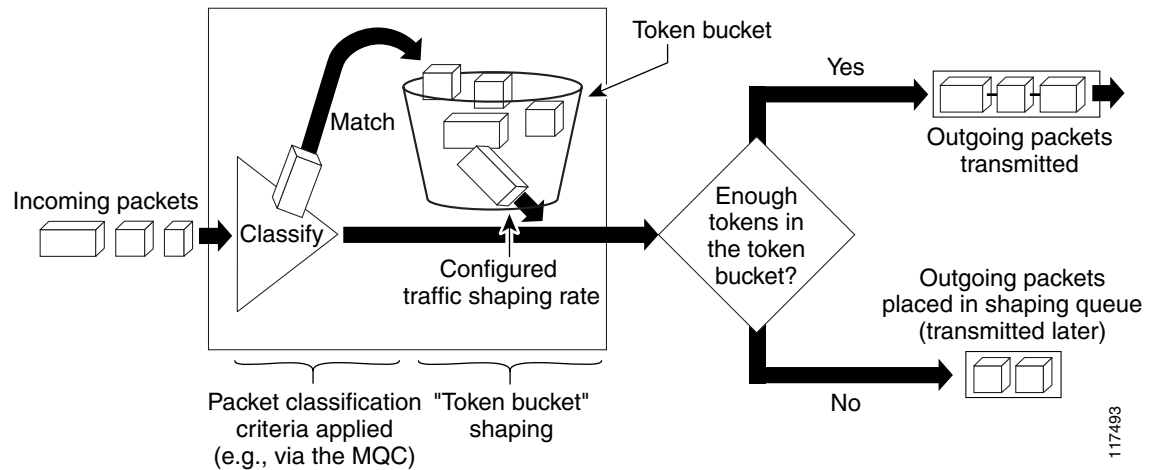
When the Be size equals 0, the interface sends no more than the burst size every interval, achieving an average rate no higher than the mean rate. However, when the Be size is greater than 0, the interface can send as many as Bc plus Be bits in a burst, if in a previous time period the maximum amount 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 17 illustrates how a traffic shaping mechanism regulates traffic.

Figure 17 How a Traffic Shaping Mechanism Regulates Traffic



In [Figure 17](#), 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 B_c size plus the B_e size. The token bucket is filled at a constant rate of B_c worth of tokens at every T_c . 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 T_c , the traffic shaper checks to see if the transmission queue contains enough packets to send (that is, up to either B_c (or B_c plus B_e) 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 30](#).

Table 30 Differences Between Traffic Shaping and Traffic Policing

	Traffic Shaping	Traffic Policing
Triggering Event	<ul style="list-style-type: none"> Occurs automatically at regular intervals (Tc). or Occurs whenever a packet arrives at an interface. 	<ul style="list-style-type: none"> Occurs whenever a packet arrives at an interface.
What it Does	<ul style="list-style-type: none"> 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. 	<ul style="list-style-type: none"> 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:

- [“Regulating Packet Flow — Using Traffic Policing”](#) module
- [Traffic Policing](#), Cisco IOS Release 12.2
- [Two-Rate Policer](#), Cisco IOS Release 12.2(4)T
- [Policer Enhancement — Multiple Actions](#), Cisco IOS Release 12.2(8)T
- [Color-Aware Policer](#), Cisco IOS Release 12.0(26)S

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 [Cisco IOS Wide-Area Networking Configuration Guide](#), Release 12.3.

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 , Release 12.3 T
Packet classification	“Classification” section of the Cisco IOS Quality of Service Solutions Configuration Guide , Release 12.3
MQC	“Modular Quality of Service Command-Line Interface” section of the Cisco IOS Quality of Service Solutions Configuration Guide , Release 12.3
WFQ, CBWFQ, PQ, CQ, FIFO and other queueing mechanisms	“Congestion Management” section of the Cisco IOS Quality of Service Solutions Configuration Guide , Release 12.3
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	Cisco IOS Wide-Area Networking Configuration Guide , Release 12.3

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 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



Regulating Packet Flow on a Per-Class Basis — Using Class-Based 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 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.

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 features. To find information about feature support and configuration, use the [“Feature Information for Class-Based Traffic Shaping”](#) section on page 462.

Contents

- [Prerequisites for Configuring Class-Based Traffic Shaping, page 453](#)
- [Restrictions for Configuring Class-Based Traffic Shaping, page 454](#)
- [Information About Class-Based Traffic Shaping, page 454](#)
- [How to Configure Class-Based Traffic Shaping, page 456](#)
- [Configuration Examples for Class-Based Traffic Shaping, page 460](#)
- [Where to Go Next, page 461](#)
- [Additional References, page 461](#)
- [Feature Information for Class-Based Traffic Shaping, page 462](#)

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 *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.3.

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, page 454](#)
- [Benefits of Class-Based Traffic Shaping, page 455](#)
- [Hierarchical Policy Map Structure of Class-Based Traffic Shaping, page 455](#)

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 higher than expected. It holds and shapes traffic to a particular bit rate by using the token bucket mechanism. See the “[Token Bucket and Traffic Shaping](#)” section in 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 455.

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 [“”](#) 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 queuing mechanism.
- Packets conforming to the specified rate are transmitted.

When packets are placed in a queue, the default queuing mechanism used is weighted fair queuing (WFQ). However, with Class-Based Traffic Shaping, class-based WFQ (CBWFQ) can be configured as an alternative queuing 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 456](#) (required)
- [Configuring the Secondary-Level \(Child\) Policy Map, page 458](#) (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 “Configuring the Modular Quality of Service Command-Line Interface” chapter in the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.3.

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** *interface-name*
10. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	policy-map <i>policy-map-name</i> 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 457 for more information. <ul style="list-style-type: none"> • Enter the policy map name.
Step 4	class { <i>class-name</i> 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. <ul style="list-style-type: none"> • Enter the name of the class or enter the class-default keyword.

	Command or Action	Purpose
Step 5	shape [average peak] <i>mean-rate</i> [[<i>burst-size</i>] [<i>excess-burst-size</i>]] Example: Router(config-pmap-c)# shape average 1000000	Shapes traffic according to the keyword and rate specified. <ul style="list-style-type: none"> Enter the keyword and rate.
Step 6	service-policy <i>policy-map-name</i> 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). <ul style="list-style-type: none"> 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 <i>interface-name</i> Example: Router# show policy-map interface s4/0	(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. <ul style="list-style-type: none"> Enter the interface name.
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 458](#).

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 455](#).

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** *interface-name*
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	policy-map <i>policy-map-name</i> 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 457 for more information. <ul style="list-style-type: none"> • Enter the policy map name.
Step 4	class { <i>class-name</i> class-default }	Specifies the name of the class whose policy you want to create and enters policy-map class configuration mode. <ul style="list-style-type: none"> • Enter the name of the class or enter the class-default keyword.

	Command or Action	Purpose
Step 5	<p>bandwidth { <i>bandwidth-kbps</i> remaining percent <i>percentage</i> percent <i>percentage</i> }</p> <p>Example: Router(config-pmap-c)# bandwidth percent 50</p>	<p>Specifies or modifies the bandwidth allocated for a class belonging to a policy map.</p> <ul style="list-style-type: none"> Enter the amount of bandwidth as a number of kbps, a relative percentage of bandwidth, or an absolute amount of bandwidth. <p>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.</p>
Step 6	<p>end</p> <p>Example: Router(config-pmap-c)# end</p>	Returns to privileged EXEC mode.
Step 7	<p>show policy-map</p> <p>Example: Router# show policy-map</p>	(Optional) Displays all configured policy maps.
Step 8	<p>show policy-map interface <i>interface-name</i></p> <p>Example: Router# show policy-map interface s4/0</p>	<p>(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.</p> <ul style="list-style-type: none"> Enter the interface name.
Step 9	<p>exit</p> <p>Example: Router# exit</p>	(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 460](#)

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 *Cisco IOS Wide-Area Networking Configuration Guide*, Release 12.3.

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	<i>Cisco IOS Quality of Service Solutions Command Reference</i> , Release 12.3 T
Packet classification	“Classification” section of the <i>Cisco IOS Quality of Service Solutions Configuration Guide</i> , Release 12.3
MQC, policy maps, class maps, and hierarchical policy maps	“Modular Quality of Service Command-Line Interface” section of the <i>Cisco IOS Quality of Service Solutions Configuration Guide</i> , Release 12.3
CBWFQ and other queueing mechanisms	“Congestion Management” section of the <i>Cisco IOS Quality of Service Solutions Configuration Guide</i> , Release 12.3
dCEF	<i>Cisco IOS Switching Services Configuration Guide</i> , Release 12.3
Overview information about using traffic shaping to regulate packet flow on a network	“” module
GTS	“ Regulating Packet Flow on a Per-Interface Basis — Using Generic Traffic Shaping ” module
FRTS	<i>Cisco IOS Wide-Area Networking Configuration Guide</i> , Release 12.3

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 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

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 Cisco IOS Release 12.2(1) 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.

For information on a feature in this technology that is not documented here, see the “[Regulating Packet Flow Roadmap](#).”

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 Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

**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: <ul style="list-style-type: none"> • Information About Class-Based Traffic Shaping, page 454 • How to Configure Class-Based Traffic Shaping, page 456



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 features. To find information about feature support and configuration, use the [“Feature Information for Generic Traffic Shaping”](#) section on page 474.

Contents

- [Prerequisites for Configuring Generic Traffic Shaping, page 465](#)
- [Restrictions for Configuring Generic Traffic Shaping, page 466](#)
- [Information About Configuring Generic Traffic Shaping, page 466](#)
- [How to Configure Generic Traffic Shaping, page 467](#)
- [Configuration Examples for Generic Traffic Shaping, page 472](#)
- [Where to Go Next, page 473](#)
- [Additional References, page 473](#)
- [Feature Information for Generic Traffic Shaping, page 474](#)

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, page 466](#)
- [Adaptive Generic Traffic Shaping on Frame Relay Networks, page 467](#)
- [Benefits of Generic Traffic Shaping, page 467](#)

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 “[Token Bucket and Traffic Shaping](#)” section in 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 470.

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 [“”](#) 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, page 467](#) (optional)
- [Configuring Generic Traffic Shaping Using an Access Control List, page 469](#) (optional)
- [Configuring Adaptive Generic Traffic Shaping for Frame Relay Networks, page 470](#) (optional)

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. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface s4/0	Configures an interface (or subinterface) type and enters interface configuration mode. <ul style="list-style-type: none">• Enter the interface type number.
Step 4	traffic-shape rate <i>bit-rate [burst-size] [excess-burst-size] [buffer-limit]</i> Example: Router(config-if)# traffic-shape rate 128000	Enables traffic shaping for outbound traffic on an interface based on the bit rate specified. <ul style="list-style-type: none">• Enter the bit rate.
Step 5	end Example: Router(config-if)# end	Returns to privileged EXEC mode.
Step 6	show traffic-shape [<i>interface-type interface-number</i>] Example: Router# show traffic-shape serial4/0	(Optional) Displays the current traffic-shaping configuration.
	show traffic-shape statistics [<i>interface-type interface-number</i>] 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. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	access-list <i>access-list-number</i> { deny permit } <i>source</i> [<i>source-wildcard</i>] Example: Router(config)# access-list 1 permit 192.5.34.0 0.0.0.255	Shapes traffic according to specified access list. <ul style="list-style-type: none">• Enter the access list number, one of the required keywords, and the source information.
Step 4	interface <i>type number</i> Example: Router(config)# interface s4/0	Configures an interface (or subinterface) type and enters interface configuration mode. <ul style="list-style-type: none">• Enter the interface type number.

	Command or Action	Purpose
Step 5	traffic-shape group <i>access-list bit-rate</i> [<i>burst-size</i> [<i>excess-burst-size</i>]] Example: Router(config-if)# traffic-shape group 101 128000	Enables traffic shaping based on a specific access list for outbound traffic on an interface. <ul style="list-style-type: none"> 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 [<i>interface-type</i> <i>interface-number</i>] Example: Router# show traffic-shape serial4/0	(Optional) Displays the current traffic-shaping configuration.
Step 8	show traffic-shape statistics [<i>interface-type</i> <i>interface-number</i>] 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. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface s4/0	Configures an interface (or subinterface) type and enters interface configuration mode. <ul style="list-style-type: none"> Enter the interface type number.
Step 4	traffic-shape rate <i>bit-rate</i> [<i>burst-size</i>] [<i>excess-burst-size</i>] [<i>buffer-limit</i>] Example: Router(config-if)# traffic-shape rate 128000	Enables traffic shaping for outbound traffic on an interface based on the bit rate specified. <ul style="list-style-type: none"> Enter the bit rate.
Step 5	traffic-shape adaptive <i>bit-rate</i> Example: Router(config-if)# traffic-shape adaptive 64000	Configures a Frame Relay subinterface to estimate the available bandwidth when BECNs are received. <ul style="list-style-type: none"> Enter the bit rate.
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 [<i>interface-type interface-number</i>] Example: Router# show traffic-shape serial4/0	(Optional) Displays the current traffic-shaping configuration.

	Command or Action	Purpose
Step 9	show traffic-shape statistics [<i>interface-type</i> <i>interface-number</i>] 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, page 472](#)
- [Generic Traffic Shaping Using an Access Control List Configuration: Example, page 472](#)
- [Adaptive Generic Traffic Shaping for a Frame Relay Network Configuration: Example, page 473](#)

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 [Cisco IOS Wide-Area Networking Configuration Guide](#), Release 12.3.

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 , Release 12.3 T
Overview information about using traffic shaping to regulate packet flow on a network	“” module
Class-Based Traffic Shaping	“Regulating Packet Flow on a Per-Class Basis — Using Class-Based Traffic Shaping” module
FRTS	Cisco IOS Wide-Area Networking Configuration Guide , Release 12.3

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 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

Feature Information for Generic 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 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.

For information on a feature in this technology that is not documented here, see the “[Regulating Packet Flow Roadmap](#).”

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 Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

**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.	—	—



Part 5: 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.

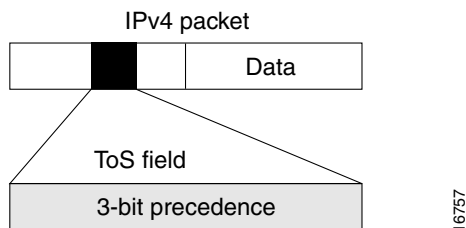
Cisco IOS QoS software takes advantage of IP to meet the challenge of finding a robust QoS signalling solution that can operate over heterogeneous network infrastructures. It overlays Layer 2 technology-specific QoS signalling solutions with Layer 3 IP QoS signalling methods of the Resource Reservation 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 18](#), 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 queuing technologies throughout the network can then use this signal to provide the appropriate expedited handling.

Figure 18 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 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.

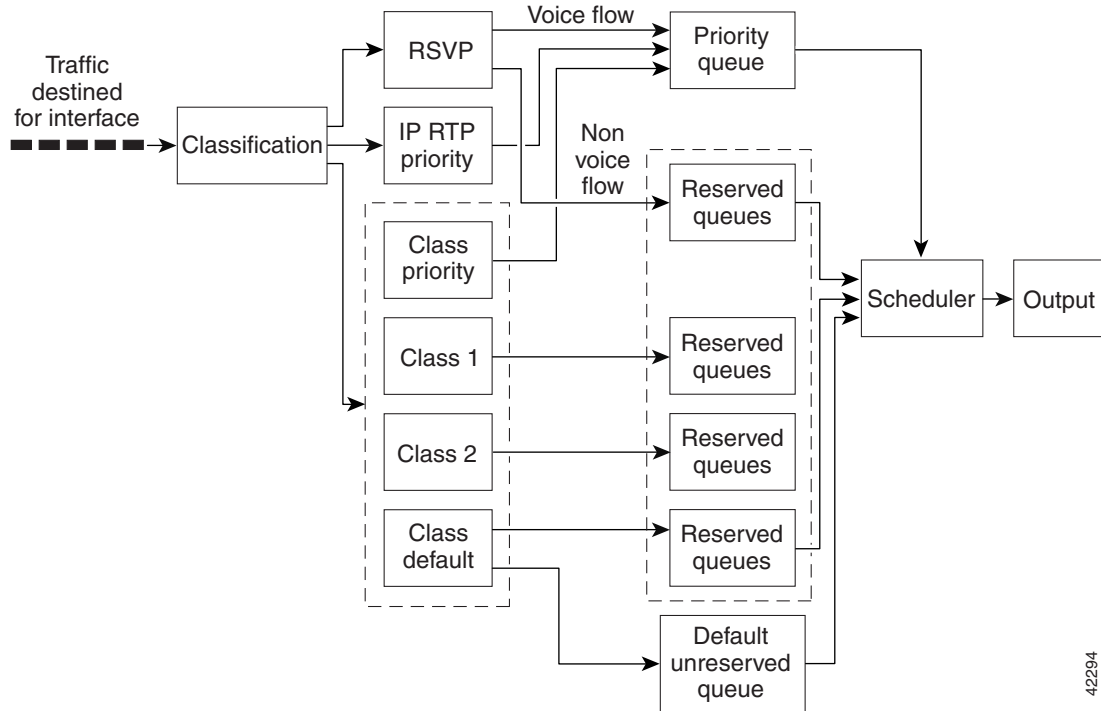
Voice traffic has stringent delay and jitter requirements. It must have very low delay and minimal jitter per hop to avoid degradation of end-to-end QoS. This requirement calls for an efficient queueing implementation, such as low latency queueing (LLQ), that can service voice traffic at almost strict 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 19 shows how RSVP operates with other Voice over IP (VoIP) features, such as **ip rtp priority**, using the same queuing mechanism, LLQ.

Figure 19 *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:

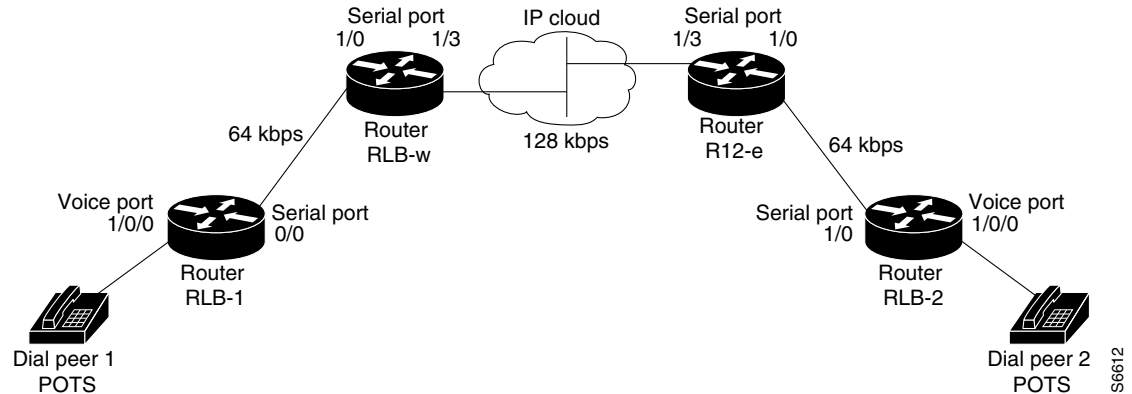
- RFC 2205, *Resource Reservation Protocol*
- RFC 2210, *RSVP with IETF Integrated Services*
- RFC 2211, *Controlled-Load Network Element Service*
- RFC 2212, *Specification of Guaranteed Quality of Service*
- RFC 2215, *General Characterization Parameters for Integrated Service Network Elements*

Figure 20 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 20 Topology Showing LLQ on Each Interface

For information on how to configure the RSVP Support for LLQ feature, see the chapter [Configuring RSVP Support for LLQ](#) in this book.

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.
- RSVP support for LLQ on Frame Relay permanent virtual circuits (PVCs) and ATM PVCs is currently not available. Support is planned for future releases.

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 queuing 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 a timely 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 queuing 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)) queuing 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) queuing 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 rsdp 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 queuing 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 RSVP Support for Frame Relay, see the chapter [Configuring RSVP Support for Frame Relay](#) in this book.

Restrictions

The following restrictions apply to the RSVP Support for Frame Relay feature:

- Interface-level Generic Traffic Shaping (GTS) is not supported.
- VC-level queuing and interface-level queuing 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.

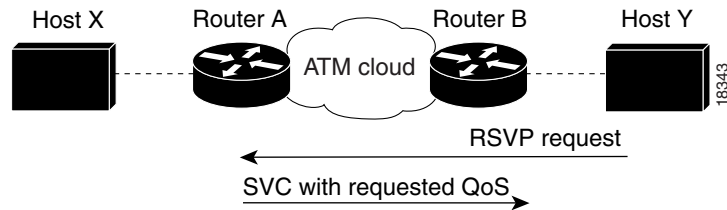
When a packet belonging to a reserved flow arrives on the interface or subinterface, the RSVP-ATM QoS Interworking software uses a token bucket to manage bandwidth guarantees. It measures actual traffic rates against the reservation flowspec to determine if the packet conforms to or exceeds the flowspec. 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 21 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 21 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 21 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 chapter [Configuring RSVP-ATM QoS Interworking](#) in this book.

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:

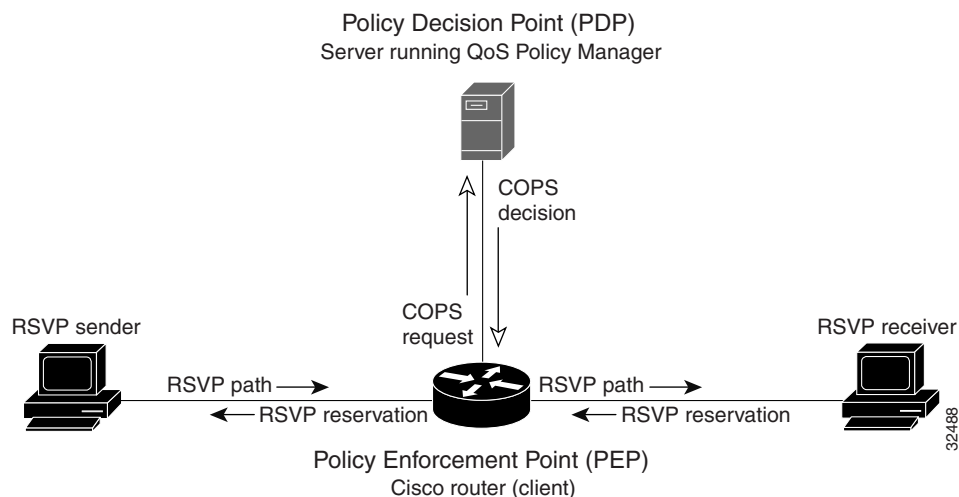
- RFC 2749, *COPS Usage for RSVP*
- RFC 2205, *Resource ReSerVation Protocol (RSVP)*
- RFC 2748, *The COPS (Common Open Policy Service) Protocol*

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 22 is a sample arrangement of COPS with RSVP.

Figure 22 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:

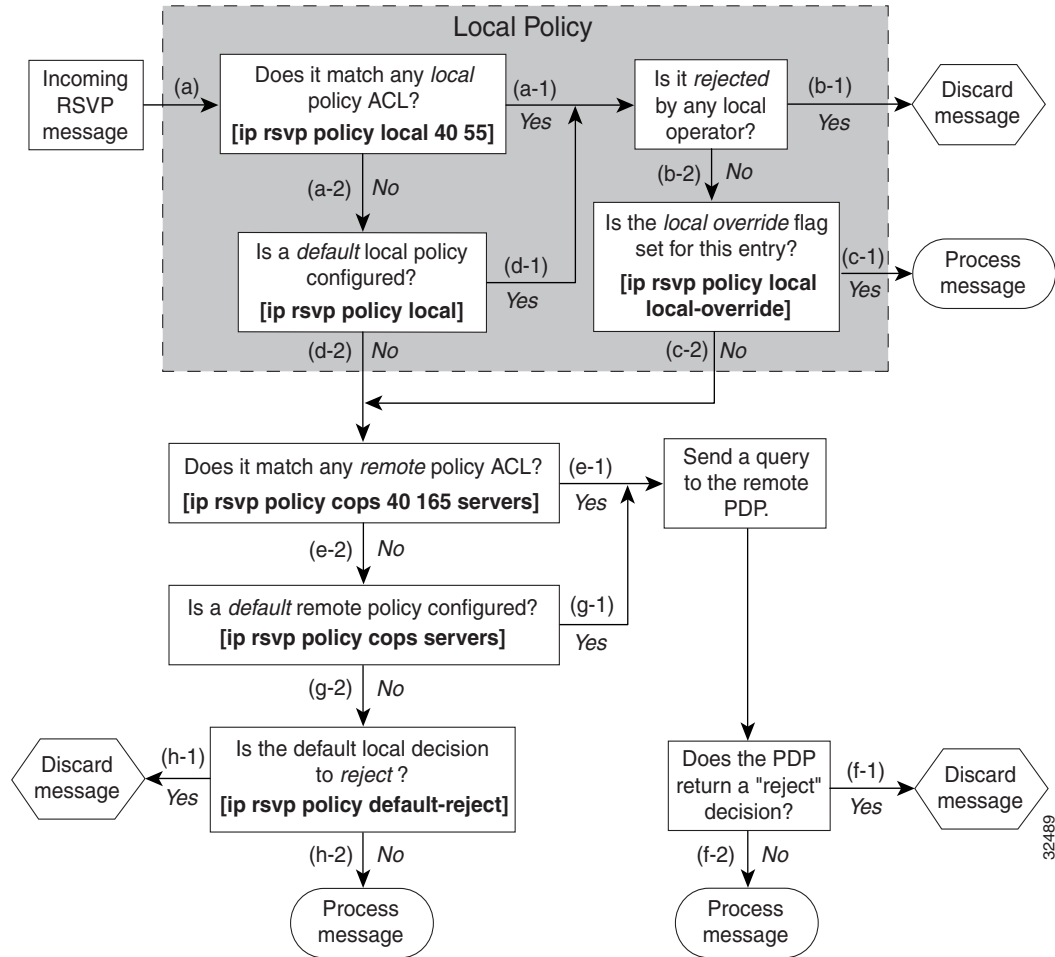
- a. When an RSVP signalling message arrives at the router, the router asks the PDP server how to process the message, either to accept, reject, forward, or install the message.
 - b. The PDP server sends its decision to the router, which then processes the message as instructed.
3. Alternatively, you 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 23 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 23 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 the sender 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 23](#)).
- 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 2](#) 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 2 Matrix for Objects the PDP Can Replace Within RSVP Messages

Message Context	Objects				Items Affected
	Policy	TSpec	Flowspec	Errorspec	
Path In	X	X	—	—	<ul style="list-style-type: none"> • 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	—	<ul style="list-style-type: none"> • 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.

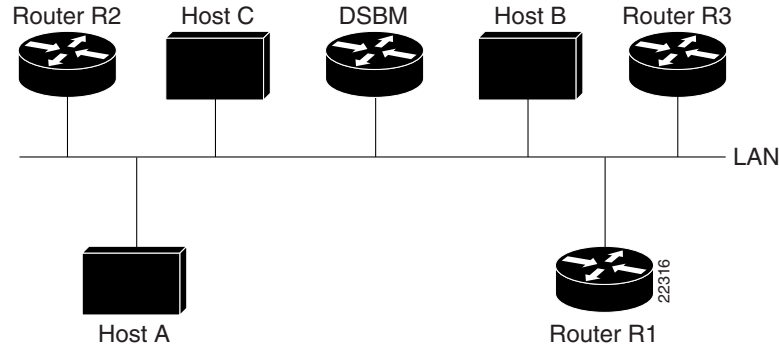


Note

SBM is not supported currently on Token Ring LANs.

[Figure 24](#) shows a managed segment in a Layer 2 domain that interconnects a set of hosts and routers.

Figure 24 **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 chapter [Configuring Subnetwork Bandwidth Manager](#) in this book.



RSVP

This part consists of the following:

- [Configuring RSVP](#)
- [Control Plane DSCP Support for RSVP](#)
- [RSVP Scalability Enhancements](#)
- [RSVP Support for ATM/PVCs](#)
- [RSVP Local Policy Support](#)
- [RSVP Refresh Reduction and Reliable Messaging](#)
- [RSVP Support for RTP Header Compression, Phase 1](#)
- [RSVP Message Authentication](#)



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 chapter, refer to 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.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the [“Finding Additional Feature Support Information” section on page lxix](#) in the [Using Cisco IOS Software for Release 12.4](#) chapter in this book.

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.

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 “[Resource Reservation Protocol](#)” section on page 480 in the chapter [Signalling Overview](#) in this book.

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 queuing 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 queuing 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, RSVP 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 [<i>interface-kbps</i>] [<i>single-flow-kbps</i>]	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 <i>session-ip-address</i> <i>sender-ip-address</i> [tcp udp <i>ip-protocol</i>] <i>session-dport</i> <i>sender-sport</i> <i>previous-hop-ip-address</i> <i>previous-hop-interface</i> <i>bandwidth</i> <i>burst-size</i>	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 <i>session-ip-address</i> <i>sender-ip-address</i> [tcp udp <i>ip-protocol</i>] <i>session-dport</i> <i>sender-sport</i> <i>next-hop-ip-address</i> <i>next-hop-interface</i> { ff se wf } { rate load } <i>bandwidth</i> <i>burst-size</i>	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 [<i>multicast-address</i>]	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 <i>access-list-number</i>	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 <i>precedence-value</i> exceed <i>precedence-value</i> }	Sets the IP Precedence conform and exceed values.
Step 2	Router(config-if)# ip rsvp tos { conform <i>tos-value</i> exceed <i>tos-value</i> }	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

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, page 513](#)
- [Supported Platforms, page 515](#)
- [Supported Standards, MIBs, and RFCs, page 515](#)
- [Prerequisites, page 515](#)
- [Configuration Tasks, page 515](#)
- [Monitoring and Maintaining Control Plane DSCP Support for RSVP, page 517](#)
- [Configuration Examples, page 517](#)
- [Command Reference, page 517](#)
- [Glossary, page 518](#)

Feature Overview

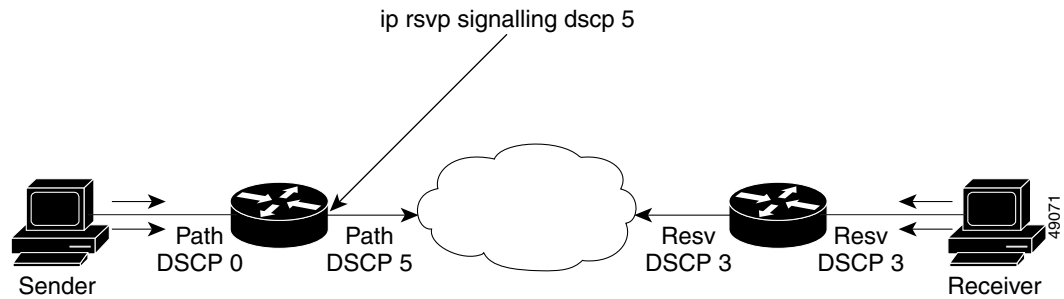
Typically, networks operate on a best-effort delivery basis, which means that all traffic has equal priority and an equal chance of being delivered in a timely manner. When congestion occurs, all traffic has an equal chance of being dropped.

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 25](#) 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 25 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:

- *Cisco IOS Quality of Service Solutions Guide*
- *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](#) (Required)

- [Specifying the DSCP](#) (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 [<i>interface-kbps</i>] [<i>single-flow-kbps</i>]	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 new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at <http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm>.

- **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.



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, page 519](#)
- [Supported Platforms, page 521](#)
- [Supported Standards, MIBs, and RFCs, page 521](#)
- [Prerequisites, page 521](#)
- [Configuration Tasks, page 522](#)
- [Monitoring and Maintaining RSVP Scalability Enhancements, page 526](#)
- [Configuration Examples, page 526](#)
- [Command Reference, page 531](#)
- [Glossary, page 532](#)

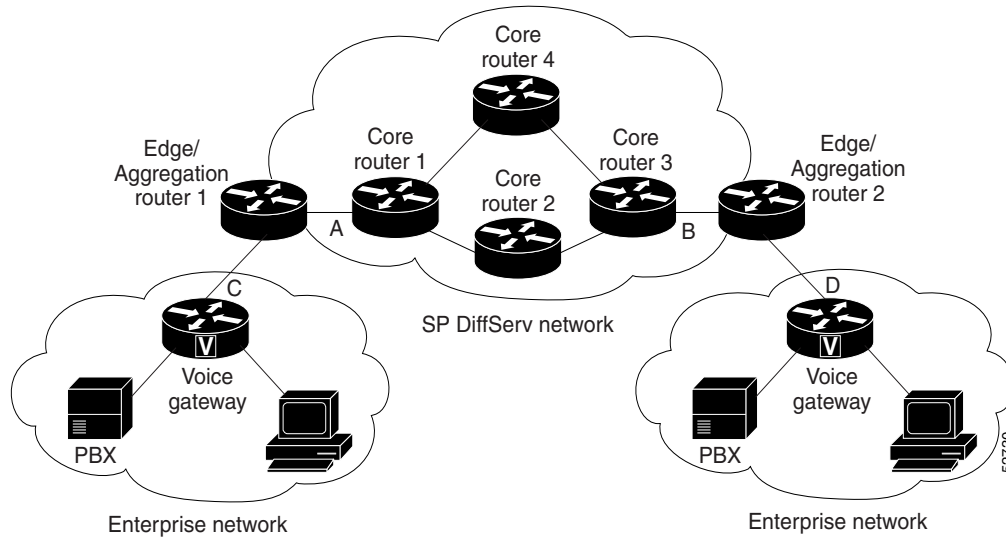
Feature Overview

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.

Figure 26 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 26 *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 of 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:

- *Cisco IOS Quality of Service Solutions Guide*
- *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 [<i>interface-kbps</i>] [<i>single-flow-kbps</i>]	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 <i>class-map-name</i>	Specifies the name of the class for which you want to create or modify class map match criteria.
Step 2	Router(config)# policy-map <i>policy-map-name</i>	Specifies the name of the policy map to be created, 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} <i>policy-map-name</i>	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
ATM6/0:
  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, 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 86 seconds
Long-term average bitrate (bits/sec): 0M 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](#)
- [Configuring the Resource Provider as None with Data Classification Turned Off](#)

Configuring CBWFQ to Accommodate Reserved Traffic

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

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 new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at <http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm>.

- **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 Support for ATM/PVCs

This document describes Cisco Resource Reservation Protocol (RSVP) support for the Asynchronous Transfer Mode/permanent virtual circuits (ATM/PVCs) 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, page 533](#)
- [Supported Platforms, page 536](#)
- [Supported Standards, MIBs, and RFCs, page 536](#)
- [Prerequisites, page 536](#)
- [Configuration Tasks, page 536](#)
- [Monitoring and Maintaining RSVP Support for ATM/PVCs, page 542](#)
- [Configuration Examples, page 542](#)
- [Command Reference, page 545](#)
- [Glossary, page 546](#)

Feature Overview

Network administrators use queueing to manage congestion on a router interface or a permanent virtual circuit (PVC). In an ATM environment, the congestion point might not be the interface itself, but the PVC because of the traffic parameters, including the available bit rate (ABR), the constant bit rate (CBR), and the variable bit rate (VBR) associated with the PVC. For real-time traffic, such as voice flows, to be transmitted in a timely manner, the data rate must not exceed the traffic parameters, or packets might be dropped, thereby affecting voice quality. Fancy queueing such as class-based weighted fair queueing (CBWFQ), low latency queueing (LLQ), or weighted fair queueing (WFQ), can run on the PVC to provide the quality of service (QoS) guarantees for the traffic.

In previous releases, RSVP reservations were not constrained by the traffic parameters of the flow's outbound PVC. As a result, oversubscription could occur when the sum of the RSVP traffic and other traffic exceeded the PVC's capacity.

The RSVP support for ATM/PVCs feature allows RSVP to function with per-PVC queueing for voice-like flows. Specifically, RSVP can install reservations on PVCs defined at the interface and subinterface levels. There is no limit to the number of PVCs 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) queuing 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 configurable (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 queuing 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.

If you do **not** want RSVP to perform per-flow classification, but prefer DiffServ classification instead, then you can configure RSVP to exclude itself from data packet classification, and configure LLQ for classification. For more information, see the "RSVP Scalability Enhancements" feature regarding DiffServ integration.

Benefits

Accurate Admission Control

RSVP performs admission control based on the PVC's average cell rate, sustainable cell rate, or minimum cell rate, depending on the type of PVC that is configured, instead of the amount of bandwidth available on the interface.

Recognition of Layer 2 Overhead

RSVP automatically takes the Layer 2 overhead into account when admitting a flow. For each flow, RSVP determines the total amount of bandwidth required, including Layer 2 overhead, and uses this value for admission control with the WFQ bandwidth manager.

Improved QoS

RSVP provides QoS guarantees for high-priority traffic by reserving resources at the point of congestion (that is, the ATM PVC instead of the interface).

Flexible Configurations

RSVP provides support for point-to-point and multipoint interface configurations, thus enabling deployment of services such as voice over IP (VoIP) in ATM environments with QoS guarantees.

Prevention of Bandwidth Oversubscription

RSVP, CBWFQ, and ip rtp priority do not oversubscribe the amount of bandwidth available on the interface or the PVC even when they are running simultaneously. Prior to admitting a reservation, these features check an internal bandwidth manager to avoid oversubscription.

IP QoS Features Integration into ATM Environments

IP QoS features can now be integrated seamlessly from IP into ATM environments with RSVP providing admission control on a per PVC basis.

Restrictions

- Interface-level generic traffic shaping (GTS) is not supported.
- VC-level queuing and interface-level queuing on the same interface are not supported.
- Nonvoice RSVP flows are not supported.
- Multicast flows are not supported.
- ATM/PVCs must be preconfigured in the network.

Related Features and Technologies

The RSVP support for ATM/PVCs feature is related to QoS features such as low latency queuing and policing. (See the section on [“Related Documents”](#).)

Related Documents

The following documents provide additional information:

- *Cisco IOS Quality of Service Solutions Guide*
- *Cisco IOS Quality of Service Solutions Command Reference*

Supported Platforms

- Cisco 3600 series (Cisco 3620, 3640, and 3660)
- Cisco 3810 multiservice access concentrator
- Cisco 7200 series

Supported Standards, MIBs, and RFCs

Standards

The RSVP support for ATM/PVCs 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 features before RSVP support for ATM/PVCs is enabled:

- Resource Reservation Protocol (RSVP)
- Weighted fair queueing (WFQ)

Configuration Tasks

See the following sections for configuration tasks for the RSVP support for ATM/PVCs feature. Each task in the list indicates whether the task is optional or required.

- [Creating a PVC](#) (Required)
- [Defining ATM QoS Traffic Parameters for a PVC](#) (Required)
- [Defining a Policy Map for WFQ](#) (Required)

- [Applying a Policy Map to a PVC](#) (Required)
- [Enabling RSVP on an Interface](#) (Required)
- [Configuring a Path](#) (Optional)
- [Configuring a Reservation](#) (Optional)

Creating a PVC

To create a PVC, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# pvc [name] vpi/vci [ilmi qsaal smds]	Assigns a name and identifier to a PVC.

Defining ATM QoS Traffic Parameters for a PVC



Note

In order for RSVP to reserve bandwidth, the ATM/PVC traffic parameters must be available bit rate (ABR), variable bit rate non real-time (VBR-NRT), or real-time variable bit rate (VBR). You can specify only one of these parameters per PVC connection; therefore, if you enter a new parameter, it will replace the existing one.

To configure ATM PVC traffic parameters, use *one* of the following commands beginning in interface-ATM-VC configuration mode:

Command	Purpose
Router(config-if-atm-vc)# abr output-pcr output-mcr	Configures the available bit rate (ABR). (ATM-CES port adapter and multiport T1/E1 ATM network module only.)
Router(config-if-atm-vc)# vbr-nrt output-pcr output-scr output-mbs	Configures the variable bit rate-non real time (VBR-NRT) QoS.
Router(config-if-atm-vc)# vbr-rt peak-rate average-rate burst	Configures the real-time variable bit rate (VBR). (Cisco MC3810 and multiport T1/E1 ATM network module only.)

The arguments used here are as follows:

- *-pcr*—peak cell rate
- *-mcr*—minimum cell rate
- *-scr*—sustainable cell rate
- *-mbs*—maximum burst size
- *output-mcr*, *output-scr*, and *average-rate*—reservable bandwidth pool on the PVC

All features running on the PVC, including RSVP, CBWFQ, and LLQ, can use up to 75 percent of the reservable bandwidth pool.

Defining a Policy Map for WFQ

To define a policy map for WFQ, use the following commands, beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# policy-map <i>policy-name</i>	Specifies the policy map name; for example, wfq-voip.
Step 2	Router(config-pmap)# class <i>class-name</i>	Specifies the name of a previously defined class map, such as class-default.
Step 3	Router(config-pmap-c) fair-queue <i>number-of-queues</i>	Specifies the number of queues to be reserved for the default class.

Applying a Policy Map to a PVC

To apply a policy map to a PVC, use the following command, beginning in interface-ATM-VC configuration mode:

Command	Purpose
Router(config-if-atm-vc)# service-policy output <i>policy-name</i>	Applies a policy map to the output direction of the interface.

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 [<i>interface-kbps</i>] [<i>single-flow-kbps</i>]	Enables RSVP on an interface.

Configuring a Path

To configure an RSVP path, use the following command in global configuration mode:

Command	Purpose
Router(config)# ip rsvp sender <i>session-ip-address</i> <i>sender-ip-address</i> [tcp udp <i>ip-protocol</i>] <i>session-dport</i> <i>sender-sport</i> <i>previous-hop-ip-address</i> <i>previous-hop-interface</i> [<i>bandwidth</i>] [<i>burst-size</i>]	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 an RSVP reservation, use the following command in global configuration mode:

Command	Purpose
<pre>Router(config)# ip rsvp reservation session-ip-address sender-ip-address [tcp udp ip-protocol] session-dport sender-sport next-hop-ip address nexthop-interface {ff se wf} {rate load} [bandwidth] [burst-size]</pre>	<p>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.</p>

Verifying RSVP Support for ATM/PVCs Configuration

Multipoint Configuration

To verify RSVP support for ATM/PVCs multipoint configuration, use this procedure:

- Step 1** Enter the **show ip rsvp installed** command to display information about interfaces, subinterfaces, PVCs, and their admitted reservations. The output in the following example shows that the ATM 6/0.1 subinterface has four reservations:

```
Router# show ip rsvp installed
```

```
RSVP:ATM6/0.1
BPS    To           From           Protoc DPort   Sport   Weight Conversation
10K    145.30.30.213 145.40.40.214 UDP    101    101    0         40
15K    145.20.20.212 145.40.40.214 UDP    100    100    6         41
15K    145.30.30.213 145.40.40.214 UDP    100    100    6         41
10K    145.20.20.212 145.40.40.214 UDP    101    101    0         40
```



Note Weight 0 is assigned to voice-like flows, which proceed to the priority queue (PQ).

- Step 2** Enter the **show ip rsvp installed detail** command to display additional information about interfaces, subinterfaces, PVCs, and their current reservations.



Note In the following output, the first flow has a weight = 0 and gets the PQ; the second flow has a weight > 0 and gets a reserved queue.

```
Router# show ip rsvp installed detail
```

```
RSVP:ATM6/0 has the following installed reservations
```

```
RSVP:ATM6/0.1 has the following installed reservations
```

```
RSVP Reservation. Destination is 145.30.30.213, Source is 145.40.40.214,
Protocol is UDP, Destination port is 101, Source port is 101
Reserved bandwidth:10K bits/sec, Maximum burst:1K bytes, Peak rate:10K bits/sec
Min Policed Unit: 0 bytes, Max Pkt Size: 1514
Resource provider for this flow:
WFQ on ATM PVC 100/101 on AT6/0: PRIORITY queue 40. Weight:0, BW 10 kbps
Conversation supports 1 reservations
Data given reserved service:0 packets (0M bytes)
Data given best-effort service:0 packets (0 bytes)
Reserved traffic classified for 48 seconds
Long-term average bitrate (bits/sec):0M reserved, 0M best-effort
RSVP Reservation. Destination is 145.20.20.212, Source is 145.40.40.214,
```

```
Protocol is UDP, Destination port is 100, Source port is 100
Reserved bandwidth:15K bits/sec, Maximum burst:1K bytes, Peak rate:15K bits/sec
Min Policed Unit: 0 bytes, Max Pkt Size: 1514
Resource provider for this flow:
  WFQ on ATM PVC 100/201 on AT6/0: RESERVED queue 41.  Weight:6, BW 15 kbps
Conversation supports 1 reservations
Data given reserved service:0 packets (0M bytes)
Data given best-effort service:0 packets (0 bytes)
Reserved traffic classified for 200 seconds
Long-term average bitrate (bits/sec):0M reserved, 0M best-effort
RSVP Reservation. Destination is 145.30.30.213, Source is 145.40.40.214,
Protocol is UDP, Destination port is 100, Source port is 100
Reserved bandwidth:15K bits/sec, Maximum burst:1K bytes, Peak rate:15K bits/sec
Min Policed Unit: 0 bytes, Max Pkt Size: 1514
Resource provider for this flow:
  WFQ on ATM PVC 100/101 on AT6/0: RESERVED queue 41.  Weight:6, BW 15 kbps
Conversation supports 1 reservations
Data given reserved service:0 packets (0M bytes)
Data given best-effort service:0 packets (0 bytes)
Reserved traffic classified for 60 seconds
Long-term average bitrate (bits/sec):0M reserved, 0M best-effort
RSVP Reservation. Destination is 145.20.20.212, Source is 145.40.40.214,
Protocol is UDP, Destination port is 101, Source port is 101
Reserved bandwidth:10K bits/sec, Maximum burst:1K bytes, Peak rate:10K bits/sec
Min Policed Unit: 0 bytes, Max Pkt Size: 1514
Resource provider for this flow:
  WFQ on ATM PVC 100/201 on AT6/0: PRIORITY queue 40.  Weight:0, BW 10 kbps
Conversation supports 1 reservations
Data given reserved service:0 packets (0M bytes)
Data given best-effort service:0 packets (0 bytes)
Reserved traffic classified for 163 seconds
Long-term average bitrate (bits/sec):0M reserved, 0M best-effort
```

Point-to-Point Configuration

To verify RSVP support for ATM/PVCs point-to-point configuration, use this procedure:

- Step 1** Enter the **show ip rsvp installed** command to display information about interfaces, subinterfaces, PVCs, and their admitted reservations. The output in the following example shows that the ATM 6/0.1 subinterface has two reservations, and the ATM 6/0.2 subinterface has one reservation:

```
Router# show ip rsvp installed

RSVP:ATM6/0.1
BPS    To                From                Protoc DPort   Sport   Weight Conversation
15K    145.30.30.213        145.40.40.214     UDP    100    100    0         40
20K    145.30.30.213        145.40.40.214     UDP    101    101    6         41

RSVP:ATM6/0.2
BPS    To                From                Protoc DPort   Sport   Weight Conversation
150K   145.20.20.212        145.40.40.214     UDP    12     12     6         42
Router#
```



Note Weight 0 is assigned to voice-like flows, which proceed to the PQ.

- Step 2** Enter the **show ip rsvp installed detail** command to display additional information about interfaces, subinterfaces, PVCs, and their current reservations.



Note In the following output, the first flow with a weight = 0 gets the PQ, and the second flow with a weight > 0 gets a reserved queue.

```
Router# show ip rsvp installed detail

RSVP:ATM6/0 has the following installed reservations

RSVP:ATM6/0.1 has the following installed reservations
RSVP Reservation. Destination is 145.30.30.213, Source is 145.40.40.214,
  Protocol is UDP, Destination port is 101, Source port is 101
  Reserved bandwidth:15K bits/sec, Maximum burst:1K bytes, Peak rate:15K bits/sec
  Min Policed Unit: 0 bytes, Max Pkt Size: 1514 bytes
  Resource provider for this flow:
    WFQ on ATM PVC 100/101 on AT6/0: PRIORITY queue 40.  Weight:0, BW 15 kbps
  Conversation supports 1 reservations
  Data given reserved service:0 packets (0M bytes)
  Data given best-effort service:0 packets (0 bytes)
  Reserved traffic classified for 48 seconds
  Long-term average bitrate (bits/sec):0M reserved, 0M best-effort
RSVP Reservation. Destination is 145.20.20.212, Source is 145.40.40.214,
  Protocol is UDP, Destination port is 100, Source port is 100
  Reserved bandwidth:15K bits/sec, Maximum burst:1K bytes, Peak rate:15K bits/sec
  Min Policed Unit: 0 bytes, Max Pkt Size: 1514 bytes
  Resource provider for this flow:
    WFQ on ATM PVC 100/201 on AT6/0: RESERVED queue 41.  Weight:6, BW 15 kbps
  Conversation supports 1 reservations
  Data given reserved service:0 packets (0M bytes)
  Data given best-effort service:0 packets (0 bytes)
  Reserved traffic classified for 200 seconds
  Long-term average bitrate (bits/sec):0M reserved, 0M best-effort
RSVP Reservation. Destination is 145.30.30.213, Source is 145.40.40.214,
  Protocol is UDP, Destination port is 100, Source port is 100
  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:
  WFQ on ATM PVC 100/101 on AT6/0: RESERVED queue 41.  Weight:6, BW 20 kbps
Conversation supports 1 reservations
Data given reserved service:0 packets (0M bytes)
Data given best-effort service:0 packets (0 bytes)
Reserved traffic classified for 60 seconds
Long-term average bitrate (bits/sec):0M reserved, 0M best-effort

RSVP:ATM6/0.2 has the following installed reservations
RSVP Reservation. Destination is 145.20.20.212, Source is 145.40.40.214,
  Protocol is UDP, Destination port is 101, Source port is 101
Reserved bandwidth:150K bits/sec, Maximum burst:1K bytes, Peak rate:150K bits/sec
Min Policed Unit: 0 bytes, Max Pkt Size: 1514 bytes
Resource provider for this flow:
  WFQ on ATM PVC 100/201 on AT6/0: PRIORITY queue 40.  Weight:0, BW 150 kbps
Conversation supports 1 reservations
Data given reserved service:0 packets (0M bytes)
Data given best-effort service:0 packets (0 bytes)
Reserved traffic classified for 163 seconds
Long-term average bitrate (bits/sec):0M reserved, 0M best-effort

```

Monitoring and Maintaining RSVP Support for ATM/PVCs

To monitor and maintain RSVP support for ATM/PVCs, 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, PVCs, and their admitted reservations.
Router# show queueing [custom fair priority random-detect [interface <i>serial-number</i>]]	Displays all or selected configured queueing strategies and available bandwidth for RSVP reservations.
Router# show atm pvc [<i>vpi/vci</i> <i>name</i> interface atm <i>interface-number</i>]	Displays all ATM PVCs and related traffic information.

Configuration Examples

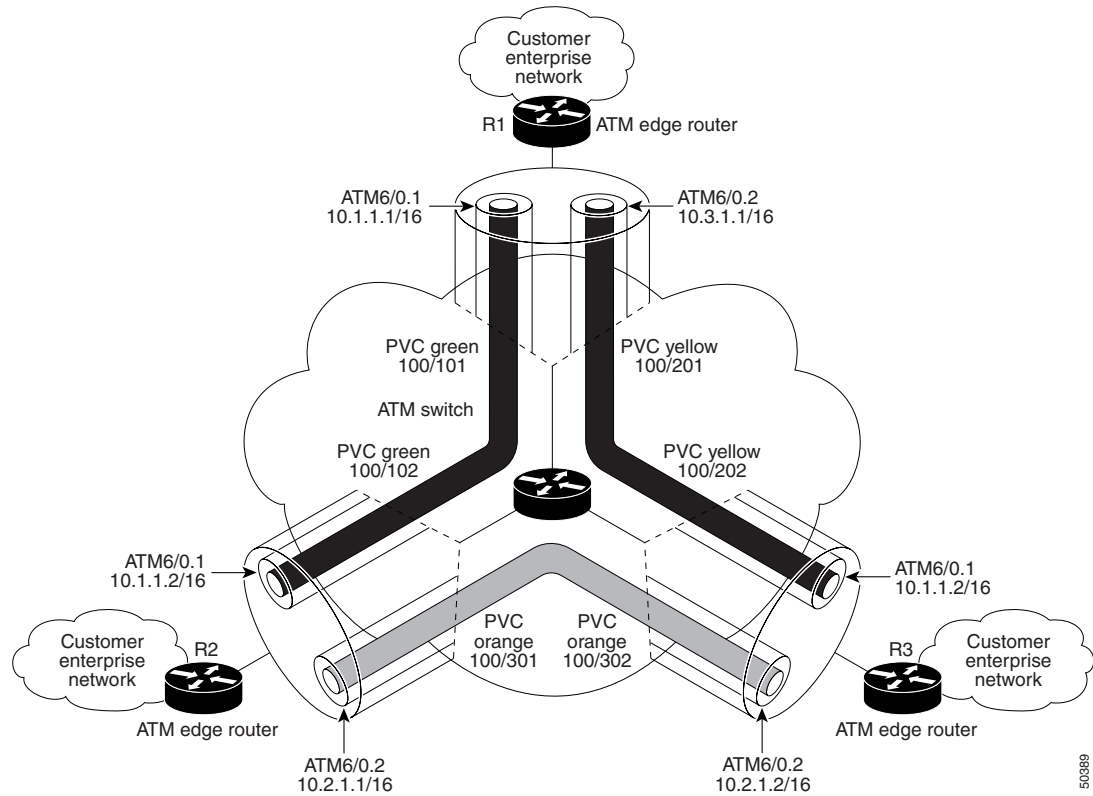
This section provides point-to-point and multipoint configuration examples for the RSVP support for ATM/PVCs feature.

Point-to-Point Configuration

Figure 27 shows a sample point-to-point interface configuration commonly used in ATM environments in which one PVC per subinterface is configured at router R1.

Three small clouds represent office branches that are connected through PVCs over an ATM network.

Figure 27 Point-to-Point Interface Configuration



Here is sample output for a point-to-point configuration:

```

Router#
policy-map wfq-voip
 class class-default
  fair-queue

interface ATM6/0
 no ip address
 ip rsvp bandwidth 112320 112320

interface ATM6/0.1 point-to-point
 ip address 10.1.1.1 255.0.0.0
 pvc green 100/101
  vbr-rt 400 300 200
  inarp 1
 broadcast
 service-policy output wfq-voip
 ip rsvp bandwidth 1250 1250
 ip rsvp resource-provider wfq pvc

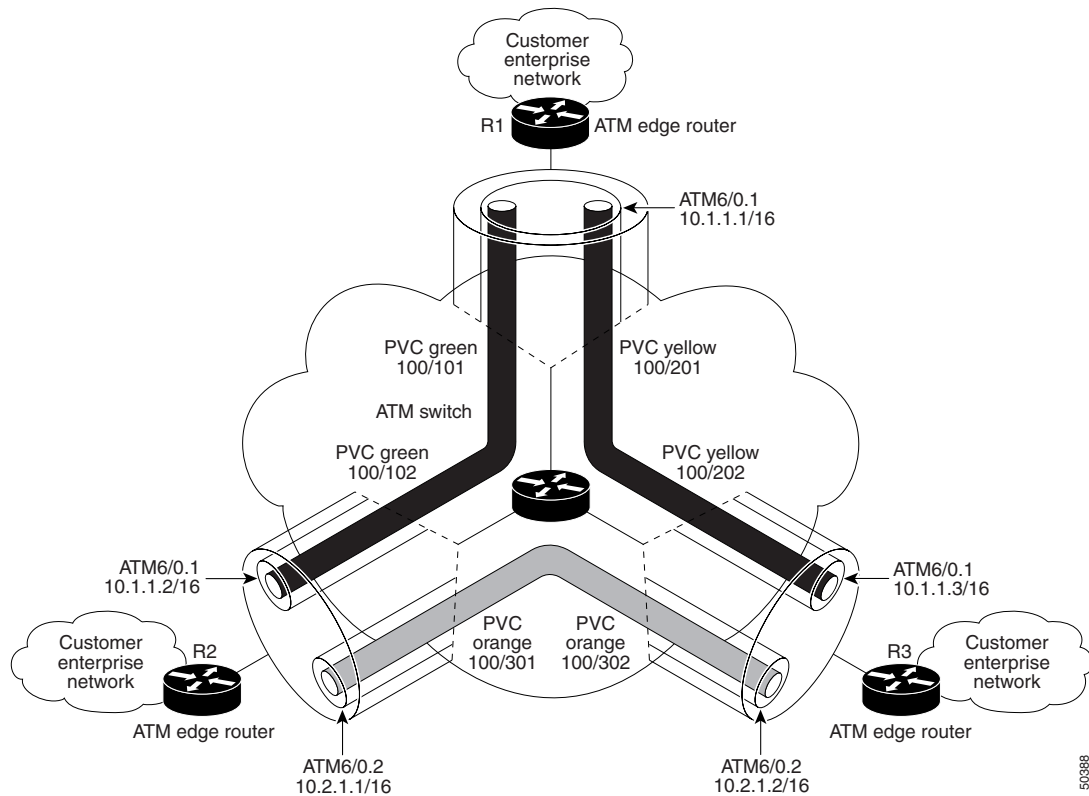
interface ATM6/0.2 point-to-point
 ip address 10.3.1.1 255.0.0.0
 pvc yellow 100/201
  vbr-nrt 500 400 1000
  inarp 1
 broadcast
 service-policy output wfq-voip
 ip rsvp bandwidth 1250 1250
 ip rsvp resource-provider wfq pvc
    
```

Multipoint Configuration

Figure 28 shows a multipoint interface configuration commonly used in ATM environments in which multiple PVCs are configured on the same subinterface at router R1.

The customer enterprise network that includes R1 is the headquarters of a company with PVC connections to each remote office.

Figure 28 Multipoint Interface Configuration



Here is sample output for a multipoint configuration:

```
Router#
policy-map wfq-voip
  class class-default
    fair-queue

interface ATM6/0
  no ip address
  ip rsvp bandwidth 112320 112320

interface ATM6/0.1 multipoint
  ip address 10.1.1.1 255.0.0.0
  pvc green 100/101
    vbr-rt 400 300 200
    inarp 1
    broadcast
    service-policy output wfq-voip

  pvc yellow 100/201
```

```
vbr-nrt 500 400 1000
inarp 1
broadcast
service-policy output wfq-voip

ip rsvp bandwidth 1250 1250
ip rsvp resource-provider wfq pvc
```

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at <http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm>.

- **debug ip rsvp traffic-control**
- **debug ip rsvp wfq**
- **ip rsvp layer2 overhead**
- **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

AAL—ATM adaptation layer. AAL defines the conversion of user information into cells. AAL1 and AAL2 handle isochronous traffic, such as voice and video; AAL3/4 and AAL5 pertain to data communications through the segmentation and reassembly of packets.

ABR—Available bit rate. A QoS class defined by the ATM Forum for ATM networks. ABR is used for connections that do not require timing relationships between source and destination. ABR provides no guarantees in terms of cell loss or delay, providing only best-effort service. Traffic sources adjust their transmission rate in response to information they receive describing the status of the network and its capability to successfully deliver data.

admission control—The process in which an RSVP reservation is accepted or rejected based on end-to-end available network resources.

Asynchronous Transfer Mode—See ATM.

ATM—Asynchronous Transfer Mode. A cell-based data transfer technique in which channel demand determines packet allocation. This is an 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.

available bit rate—See ABR.

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.

CBR—Constant bit rate. A QoS class defined by the ATM Forum for ATM networks. CBR is used for connections that depend on precise clocking to ensure undistorted delivery.

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.

constant bit rate—See CBR.

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.

ILMI—Interim Local Management Interface. Described in the ATM Forum's UNI specification, ILMI allows end users to retrieve basic information, such as status and configuration about virtual connections and addresses, for a particular UNI.

Interim Local Management Interface—See ILMI.

latency—The delay between the time a device receives a packet and the time that the packet is forwarded out the destination port.

MUX—A multiplexing device that combines multiple signals for transmission over a single line. The signals are demultiplexed, or separated, at the receiving end.

payload—The portion of a cell, frame, or packet that contains upper-layer information (data).

permanent virtual circuit—See PVC.

point-to-multipoint connection—One of two fundamental connection types. It is a unidirectional connection in which a single source end system (known as a root node) connects to multiple destination end systems (known as leaves).

point-to-point connection—One of two fundamental connection types. It is a unidirectional or bidirectional connection between two end systems.

PQ—Priority queue. A routing feature in which frames in an output queue are assigned priority based on various characteristics such as packet size and interface type.

priority queue—See PQ.

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.

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.

reservable bandwidth pool—The amount of bandwidth on a link that features can set aside in order to provide QoS guarantees.

Resource Reservation Protocol—See RSVP.

RSVP—Resource Reservation Protocol. A protocol for reserving network resources to provide quality of service guarantees to application flows.

SNAP—Subnetwork Access Protocol. An Internet protocol that operates between a network entity in the subnetwork and a network entity in the end system. SNAP specifies a standard method of encapsulating IP datagrams and ARP messages on IEEE networks. The SNAP entity in the end system makes use of the services of the subnetwork and performs three key functions: data transfer, connection management, and QoS selection.

subnetwork access protocol—See SNAP.

SVC—Switched virtual circuit or connection. A virtual circuit that is dynamically established on demand and is torn down when transmission is complete. SVCs are used in situations where data transmission is sporadic.

switched virtual circuit—See SVC.

variable bit rate—See VBR.

VBR—Variable bit rate. A QoS class defined by the ATM Forum for ATM networks. VBR is subdivided into a real time (RT) class and a non-real time (NRT) class. VBR (RT) is used for connections in which there is a fixed timing relationship between samples. VBR (NRT) is used for connections where there is no fixed timing relationship between samples, but where a guaranteed QoS is still needed.

VC—Virtual circuit. A logical circuit created to ensure reliable communication between two network devices. A virtual circuit can be either permanent (PVC) or switched (SVC).

virtual circuit—See VC.

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 Local Policy Support

Feature History

Release	Modification
12.2(13)T	This feature was introduced.

This document describes the Resource Reservation Protocol (RSVP) Local Policy Support feature in Cisco IOS Release 12.2(13)T. It identifies the supported platforms, provides configuration examples, and lists related Cisco IOS command line interface (CLI) commands.

This document includes the following sections:

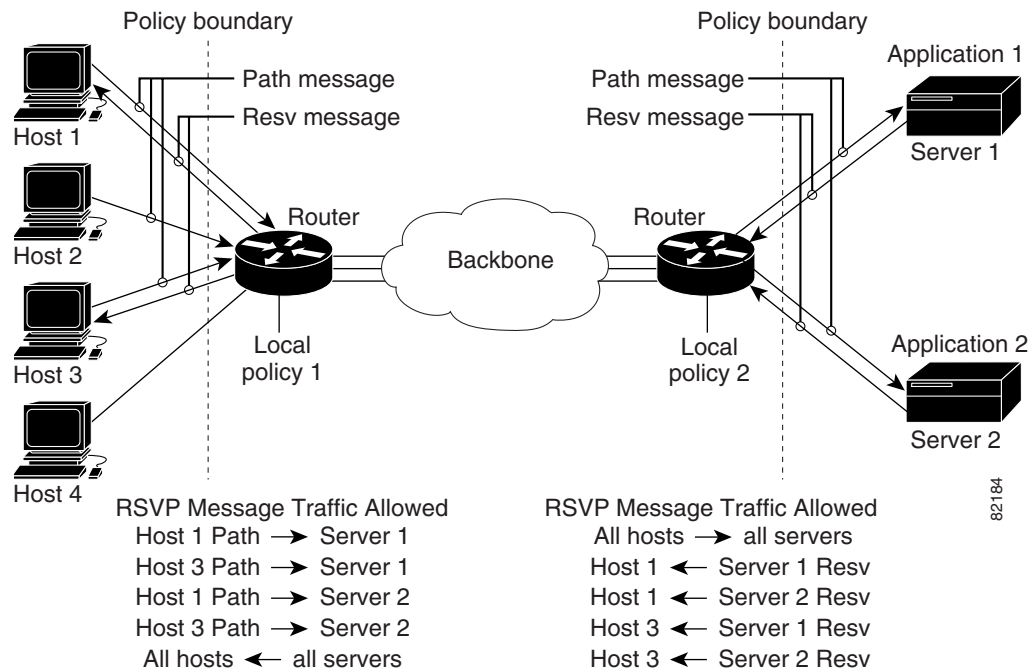
- [Feature Overview, page 549](#)
- [Supported Platforms, page 551](#)
- [Supported Standards, MIBs, and RFCs, page 551](#)
- [Prerequisites, page 552](#)
- [Configuration Tasks, page 552](#)
- [Monitoring and Maintaining RSVP Local Policy Support, page 554](#)
- [Configuration Examples, page 554](#)
- [Command Reference, page 555](#)
- [Glossary, page 556](#)

Feature Overview

Network administrators need the ability to control the resources that RSVP reservations are allowed to use. For example, they may want to restrict RSVP reservations to certain subnets or from specific network servers.

The RSVP Local Policy Support feature allows network administrators to create default and access control list (ACL)-based policies. These policies, in turn, control how RSVP filters its signalling messages to allow or deny quality of service (QoS), as shown in [Figure 29](#), to networking applications based on the IP addresses of the requesting hosts.

Figure 29 RSVP Local Policy Configuration



Benefits

RSVP Reservation Control

Network administrators can restrict the source of RSVP reservations to specific endpoints.

RSVP Reservation Preemption

High priority reservations can preempt existing reservations if there is otherwise no bandwidth available for the new, high priority reservation.

Related Features and Technologies

The RSVP Local Policy Support feature is related to QoS features such as signalling, classification, and congestion management. (See the “[Related Documents](#)” section.)

Related Documents

The following documents provide additional information:

- [Cisco IOS Quality of Service Solutions Configuration Guide](#)
- [Cisco IOS Quality of Service Solutions Command Reference](#)

Supported Platforms

For supported platforms in Cisco IOS Release 12.2(13)T, consult Cisco Feature Navigator.

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 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, refer to 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 MIBs 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

RSVP must be configured on two or more routers or on one router and one host within the network before you can use the RSVP Local Policy Support feature.

Configuration Tasks

See the following section for configuration tasks for the RSVP Local Policy Support feature. Each task in the list indicates whether the task is optional or required.

- [Creating an RSVP Local Policy](#) (required)
- [Specifying Command Line Interface \(CLI\) Submodes](#) (required)

Creating an RSVP Local Policy

To create an RSVP local policy, use the following command beginning in global configuration mode:

Command	Purpose
Router(config)# ip rsvp policy local {default acl acl [acl1...acl8]}	Creates a local policy to determine how RSVP resources are used in a network.

Specifying Command Line Interface (CLI) Submodes

To specify CLI submodes, use the following command beginning in local policy mode:

Command	Purpose
Router(config-rsvp-policy-local)# {accept forward} {all path path-error resv resv-error}	Defines the properties of the default or ACL-based local policy that you are creating.

See the **ip rsvp policy local** command for more detailed information on submodes.

Verifying RSVP Local Policy Configuration

To verify RSVP local policy configuration, use this procedure:

- Step 1** Enter the **show ip rsvp policy** command to display policy-related information including local and default policies configured, Common Open Policy Service (COPS) servers configured, and the preemption parameter configured—enabled or disabled.



Note There are no COPS servers configured in the following output.

```
Router# show ip rsvp policy

Local policy:

    A=Accept    F=Forward

    Path:-- Resv:-- PathErr:-- ResvErr:-- ACL:104
    Path:-- Resv:-- PathErr:-- ResvErr:-- ACL:None [Default policy]

COPS:

Generic policy settings:
    Default policy: Accept all
    Preemption:     Disabled
```

- Step 2** Enter the **show ip rsvp policy local detail** command to display information about the (selected) local policies currently configured.

```
Router# show ip rsvp policy local detail

Local policy for ACL(s): 104
    Preemption Priority: Start at 0, Hold at 0.
    Local Override: Disabled.

            Accept    Forward
Path:        No       No
Resv:        No       No
PathError:   No       No
ResvError:   No       No

Default local policy:
    Preemption Priority: Start at 0, Hold at 0.
    Local Override: Disabled.

            Accept    Forward
Path:        No       No
Resv:        No       No
PathError:   No       No
ResvError:   No       No

Generic policy settings:
    Default policy: Accept all
    Preemption:     Disabled
```

Monitoring and Maintaining RSVP Local Policy Support

To monitor and maintain the RSVP Local Policy Support feature, use the following commands in EXEC mode:

Command	Purpose
Router# show ip rsvp policy	Displays either the configured COPS servers or the local policies.
Router# show ip rsvp policy local	Displays selected local policies that have been configured.
Router# show ip rsvp reservation detail	Displays detailed RSVP-related receiver information currently in the database.
Router# show ip rsvp sender detail	Displays detailed RSVP-related sender information currently in the database.

Configuration Examples

This section provides a configuration example for the RSVP Local Policy Support feature.

RSVP Local Policy Support Example

In the following example, any RSVP nodes in the 192.168.101.0 subnet can initiate or respond to reservation requests, but all other nodes can respond only to reservation requests. This means that any 192.168.101.x node can send and receive Path, PathError, Resv, or ResvError messages. All other nodes can send only Resv or ResvError messages.

In the following example, ACL 104 is configured for a local policy:

```
Router# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Router(config)# access-list 104 permit ip 192.168.101.0 0.0.0.255 any

Router(config)# ip rsvp policy local acl 104

Router(config-rsvp-policy-local)# forward all

Router(config-rsvp-policy-local)# end
```

In the following example, a default local policy is configured:

```
Router(config)# ip rsvp policy local default

Router(config-rsvp-policy-local)# forward resv

Router(config-rsvp-policy-local)# forward resverror

Router(config-rsvp-policy-local)# end
```


Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at <http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm>.

New Commands

- **ip rsvp policy local**
- **ip rsvp policy preempt**
- **show ip rsvp policy local**

Modified Commands

- **show ip rsvp policy**

Glossary

access control list—See ACL.

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, though it may be used to provide a generic packet classification facility.

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.

latency—The delay between the time a device receives a packet and the time that packet is forwarded out the destination port.

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.

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.

port scanning—The act of systematically checking a computer's ports to find an access point.

Resource Reservation Protocol—See RSVP.

RSVP—Resource Reservation Protocol. A protocol for reserving network resources to provide quality of service guarantees to application flows.

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.

tunnel—A secure communications path between two peers, such as routers.

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.



RSVP Refresh Reduction and Reliable Messaging

The RSVP Refresh Reduction and Reliable Messaging feature includes refresh reduction, which improves the scalability, latency, and reliability of Resource Reservation Protocol (RSVP) signalling to enhance network performance and message delivery.

Feature Specifications for RSVP Refresh Reduction and Reliable Messaging

Feature History

Release	Modification
12.2(13)T	This feature was introduced.

Supported Platforms

For supported platforms in Cisco IOS Release 12.2(13)T, consult Cisco Feature Navigator.

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 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, refer to the online release notes or, if supported, Cisco Feature Navigator.

Contents

- [Prerequisites for RSVP Refresh Reduction and Reliable Messaging, page 558](#)
- [Restrictions for RSVP Refresh Reduction and Reliable Messaging, page 558](#)
- [Information About RSVP Refresh Reduction and Reliable Messaging, page 558](#)
- [How to Configure RSVP Refresh Reduction and Reliable Messaging, page 561](#)
- [Configuration Examples for RSVP Refresh Reduction and Reliable Messaging, page 565](#)
- [Additional References, page 567](#)
- [Command Reference, page 569](#)
- [Glossary, page 571](#)

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 RSVP Refresh Reduction and Reliable Messaging, you need to understand the following concepts:

- [Feature Design of RSVP Refresh Reduction and Reliable Messaging, page 559](#)
- [Types of Messages in RSVP Refresh Reduction and Reliable Messaging, page 559](#)
- [Benefits of RSVP Refresh Reduction and Reliable Messaging, page 561](#)

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 using RSVP signalling, 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**

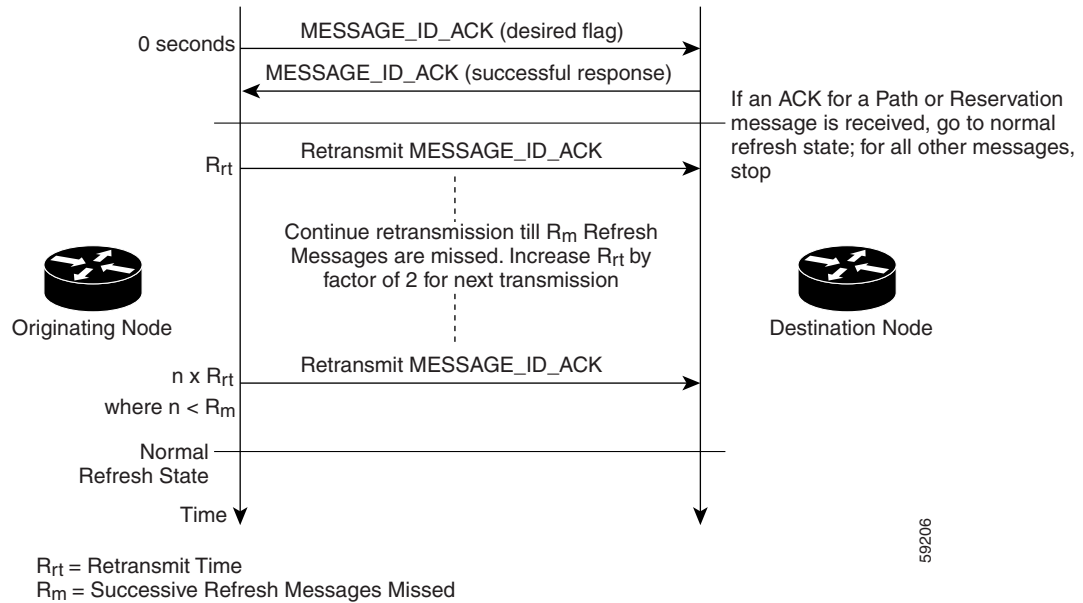
RSVP packets consist of headers that identify the types of messages, and object fields that contain attributes and properties describing how to interpret and act on the content.

Types of Messages in RSVP Refresh Reduction and Reliable Messaging

The RSVP Refresh Reduction and Reliable Messaging feature (Figure 30) includes refresh reduction, which improves the scalability, latency, and reliability of RSVP signalling 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 30 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, then 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 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

In this release, 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 are 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:

- [Enable RSVP on an Interface, page 561](#) (required)
- [Enable RSVP Refresh Reduction, page 562](#) (required)
- [Verify RSVP Refresh Reduction and Reliable Messaging, page 564](#) (optional)

Enable RSVP on an Interface

Perform these tasks to enable RSVP on an interface.

SUMMARY STEPS

1. **enable**
2. **configure** { **terminal** | **memory** | **network** }

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 higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	configure { terminal memory network } Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface [<i>type number</i>] Example: Router(config-if)# interface Ethernet1	Enters interface configuration mode. <ul style="list-style-type: none"> The <i>type number</i> argument identifies the interface to be configured.
Step 4	ip rsvp bandwidth [<i>interface-kbps</i> [<i>sub-pool</i>]] Example: Router(config-if)# ip rsvp bandwidth 7500 7500	Enables RSVP on an interface. <ul style="list-style-type: none"> The optional <i>interface-kbps</i> and <i>sub-pool</i> 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, and 0 to 10,000,000, respectively.
Step 5	end Example: Router(config-if)# end	Exits to privileged EXEC mode.

Enable RSVP Refresh Reduction

Perform these tasks to enable RSVP refresh reduction.

SUMMARY STEPS

1. **enable**
2. **configure** {**terminal** | **memory** | **network**}
3. **ip rsvp signalling refresh reduction**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	configure {terminal memory network} 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	Exits to privileged EXEC mode.

Verify RSVP Refresh Reduction and Reliable Messaging

Perform these tasks 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 higher privilege levels, such as 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 <i>interface_unit</i> summary neighbor] Example: Router# show ip rsvp counters summary	(Optional) Displays the number of RSVP messages that were sent and received on each interface. <ul style="list-style-type: none"> • 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 [<i>interface-type interface-number</i>] [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. <ul style="list-style-type: none"> • The optional detail keyword displays the bandwidth and signalling parameters.
Step 6	show ip rsvp neighbor [detail] Example: Router# show ip rsvp neighbor detail	(Optional) Displays RSVP-neighbor information including IP addresses. <ul style="list-style-type: none"> • 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, page 565](#)

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 additional references related to the RSVP Refresh Reduction and Reliable Messaging feature:

- [Related Documents, page 567](#)
- [Standards, page 567](#)
- [MIBs, page 567](#)
- [RFCs, page 568](#)
- [Technical Assistance, page 568](#)

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, Release 12.2
QoS features including signalling, classification, and congestion management	Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.2

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 ¹	MIBs Link
<ul style="list-style-type: none"> • RFC 2206, <i>RSVP Management Information Base using SMIPv2</i> • RFC 2702, <i>Requirements for Traffic Engineering over MPLS</i> 	<p>To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:</p> <p>http://www.cisco.com/public/sw-center/netmgmt/ctmk/mibs.shtml</p>

1. Not all supported MIBs are listed.

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/ctmk/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

RFCs ¹	Title
RFC 2205	<i>Resource Reservation Protocol</i>
RFC 2209	<i>RSVP—Version 1 Message Processing Rules</i>
RFC 2210	<i>The Use of RSVP with IETF Integrated Services</i>
RFC 2211/2212	<i>Specification of the Controlled-Load Network Element Service</i>
RFC 2749	<i>Common Open Policy Service (COPS) Usage for RSVP</i>
RFC 2750	<i>RSVP Extensions for Policy Control</i>
RFC 2814	<i>SBM Subnet Bandwidth Manager: A Protocol for RSVP-based Admission Control over IEEE 802-style networks</i>
RFC 2961	<i>RSVP Refresh Overhead Reduction Extensions</i>
RFC 2996	<i>Format of the RSVP DCLASS Object</i>

1. Not all supported RFCs are listed.

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, tools, and lots more. 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 new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at <http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm>.

New Commands

- **clear ip rsvp signalling refresh reduction**
- **debug ip rsvp dump-messages**
- **debug ip rsvp rate-limit**
- **debug ip rsvp reliable-msg**
- **debug ip rsvp summary-refresh**
- **ip rsvp listener**
- **ip rsvp signalling initial-retransmit-delay**
- **ip rsvp signalling patherr state-removal**
- **ip rsvp signalling refresh reduction**
- **ip rsvp signalling refresh reduction ack-delay**
- **show ip rsvp listeners**
- **show ip rsvp signalling**
- **show ip rsvp signalling blockade**
- **show ip rsvp signalling rate-limit**
- **show ip rsvp signalling refresh reduction**

Modified Commands

- **clear ip rsvp counters**
- **clear ip rsvp msg-pacing**
- **clear ip rsvp signalling rate-limit**
- **debug ip rsvp**
- **ip rsvp msg-pacing**
- **ip rsvp signalling rate-limit**
- **show ip rsvp**
- **show ip rsvp counters**
- **show ip rsvp interface**
- **show ip rsvp neighbor**

Obsolete and Replaced Commands

Table 3 lists those commands that have been replaced in Cisco IOS Release 12.2(13)T:

Table 3 *Replaced IOS Commands*

Command in Cisco IOS Release 12.0(14)ST	Replacement Command in Cisco IOS Release 12.2(13)T
<code>clear ip rsvp msg-pacing</code>	<code>clear ip rsvp signalling rate-limit</code>
<code>ip rsvp msg-pacing</code>	<code>ip rsvp signalling rate-limit</code>

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.

latency—The delay between the time a device receives a packet and the time that packet is forwarded out the destination port.

LSP—Label-switched path. A sequence of hops in which a packet travels from one router to another router by means of label-switching mechanisms. A label-switched path can be established dynamically, based on normal routing mechanisms, or through configuration.

MPLS—Multiprotocol Label Switching (formerly known as tag switching). A method for directing packets primarily through Layer 2 switching rather than Layer 3 routing. In MPLS, packets are assigned short, fixed-length labels at the ingress to an MPLS cloud by using the concept of forwarding equivalence classes. Within the MPLS domain, the labels are used to make forwarding decisions mostly without recourse to the original packet headers.

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.

refresh message—A message that represents a previously advertised state, contains the same objects and information as a previously transmitted message, and is sent over the same path.

Resource Reservation Protocol—See RSVP.

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 for reserving network resources to provide quality of service guarantees to application flows.

soft state—The status that RSVP maintains in routers and end nodes so that they can be updated by certain RSVP messages. The soft state characteristic permits an RSVP network to support dynamic group membership changes and adapt to changes in routing.

sub pool—A division of bandwidth such that no one tunnel dominates.

tunnel—A secure communications path between two peers, such as routers.

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.



RSVP Support for RTP Header Compression, Phase 1

The Resource Reservation Protocol (RSVP) Support for Real-Time Transport Protocol (RTP) Header Compression, Phase 1 feature provides a method for decreasing a flow's reserved bandwidth requirements so that a physical link can accommodate more voice calls.

Feature Specifications for RSVP Support for RTP Header Compression, Phase 1

Feature History

Release	Modification
12.2(15)T	This feature was introduced.

Supported Platforms

For platforms supported in Cisco IOS Release 12.2(15)T, consult Cisco Feature Navigator.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- [Prerequisites for RSVP Support for RTP Header Compression, Phase 1, page 574](#)
- [Restrictions for RSVP Support for RTP Header Compression, Phase 1, page 574](#)
- [Information About RSVP Support for RTP Header Compression, Phase 1, page 574](#)
- [How to Configure RSVP Support for RTP Header Compression, Phase 1, page 576](#)
- [Configuration Examples for RSVP Support for RTP Header Compression, Phase 1, page 580](#)
- [Additional References, page 581](#)
- [Command Reference, page 583](#)
- [Glossary, page 584](#)

Prerequisites for RSVP Support for RTP Header Compression, Phase 1

- Ensure that Real-Time Transport Protocol (RTP) or User Data Protocol (UDP) header compression is configured in the network.
- Ensure that RSVP is configured on two or more routers within the network before you can use this feature.

Restrictions for RSVP Support for RTP Header Compression, Phase 1

- Routers do not generate compression hints, as described in RFC 3006, in this release.
- Signalled compression hints are not supported.
- Admission control with compression is limited to reservations with one sender per session.

Information About RSVP Support for RTP Header Compression, Phase 1

To configure RSVP Support for RTP Header Compression, Phase 1, you need to understand the following concepts:

- [Feature Design of RSVP Support for RTP Header Compression, Phase 1, page 574](#)
- [Benefits of RSVP Support for RTP Header Compression, Phase 1, page 575](#)

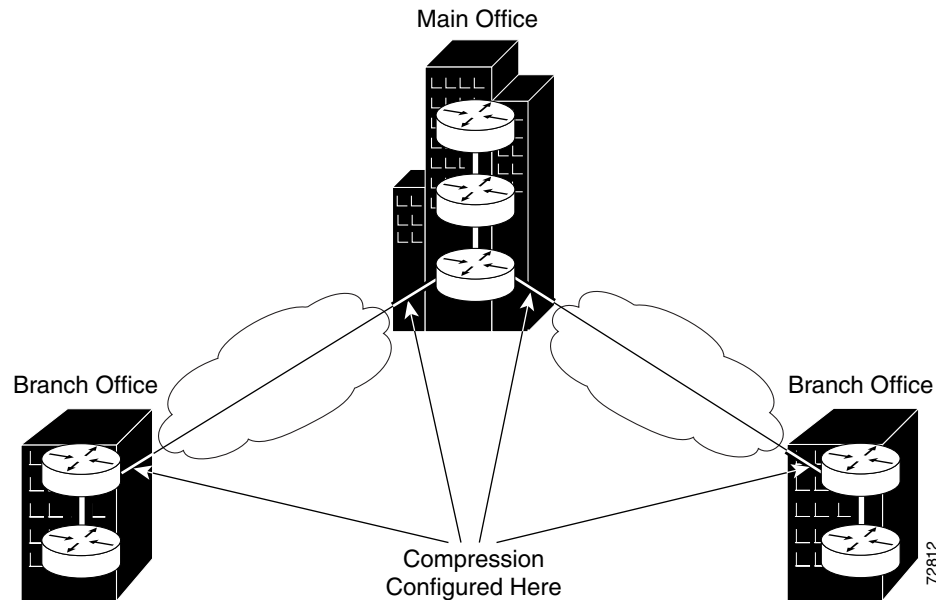
Feature Design of RSVP Support for RTP Header Compression, Phase 1

Network administrators use RSVP with Voice over IP (VoIP) to provide quality of service (QoS) for voice traffic in a network. Because VoIP is a real-time application, network administrators often configure compression within the network to decrease bandwidth requirements. Typically, compression is configured on slow serial lines ([Figure 31](#)), where the savings from reduced bandwidth requirements outweigh the additional costs associated with the compression and decompression processes.

**Note**

RTP header compression is supported by Cisco routers.

Figure 31 **Configuring Compression**



Originating applications know if their traffic is considered compressible, but not whether the network can actually compress the data. Additionally, compression may be enabled on some links along the call's path, but not on others. Consequently, the originating applications must advertise their traffic's uncompressed bandwidth requirements, and receiving applications must request reservation of the full amount of bandwidth. This causes routers whose RSVP implementations do not take compression into consideration to admit the same number of flows on a link running compression as on one that is not.

Predicting Compression within Admission Control

Network administrators, especially those whose networks have very low speed links, may want RSVP to use their links as fully as possible. Such links typically have minimum acceptable outgoing committed information rate (minCIR) values between 19 and 30 kbps. Without accounting for compression, RSVP can admit (at most) one G.723 voice call onto the link, despite the link's capacity for two compressed calls. Under these circumstances, network administrators may be willing to sacrifice a QoS guarantee for the last call, if the flow is less compressible than predicted, in exchange for the ability to admit it.

In order to account for compression during admission control, routers use signalled Tspec information, as well as their awareness of the compression schemes running on the flow's outbound interfaces, to make local decisions as to how much bandwidth should actually be reserved for a flow. By reserving fewer resources than signalled by the receiver, RSVP can allow links to be more fully used.

Benefits of RSVP Support for RTP Header Compression, Phase 1

Additional Calls Accommodated on the Same Link

The RSVP Support for RTP Header Compression, Phase 1 feature performs admission control based on compressed bandwidth so that additional voice calls can be accommodated on the same physical link.

How to Configure RSVP Support for RTP Header Compression, Phase 1

This section contains the following procedure:

- [Configuring RSVP Admission-Control Compression, page 576](#) (optional)

Configuring RSVP Admission-Control Compression



Note

RSVP predicted compression is enabled by default.

Perform this task to configure RSVP admission-control compression.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** [*type number*]
4. **ip rsvp admission-control compression predict** [method {**rtp** | **udp**}] [bytes-saved *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 [<i>type number</i>] Example: Router(config-if)# interface Serial3/0	Enters interface configuration mode. • The <i>type number</i> argument identifies the interface to be configured.

	Command or Action	Purpose
Step 4	<pre>ip rsvp admission-control compression predict [method {rtp udp} [bytes-saved N]]</pre> <p>Example: Router(config-if)# ip rsvp admission-control compression predict method udp bytes-saved 16</p>	<p>Configures RSVP admission-control compression prediction.</p> <ul style="list-style-type: none"> The optional method keyword allows you to select Real-Time Transport Protocol (rtp) or User Data Protocol (udp) for your compression scheme. The optional bytes-saved N keyword allows you to configure the predicted number of bytes saved per packet when RSVP predicts that compression will occur using the specified method.
Step 5	<pre>end</pre> <p>Example: Router(config-if)# end</p>	<p>Exits to privileged EXEC mode.</p>

Verifying RSVP Support for RTP Header Compression, Phase 1 Configuration

Perform this task to verify that the RSVP Support for RTP Header Compression, Phase 1 feature is functioning.

SUMMARY STEPS

1. **enable**
2. **show ip rsvp installed [detail]**
3. **show ip rsvp interface [interface-type interface-number] [detail]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	<pre>enable</pre> <p>Example: Router> enable</p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> Enter your password if prompted.

	Command or Action	Purpose
Step 2	show ip rsvp installed [detail] Example: Router# show ip rsvp installed detail	Displays information about interfaces and their admitted reservations and the resources needed for a traffic control state block (TCSB) after taking compression into account. <ul style="list-style-type: none"> The optional detail keyword displays the reservation's traffic parameters, downstream hop, compression, and resources used by RSVP to ensure QoS for this reservation.
Step 3	show ip rsvp interface [<i>interface-type interface-number</i>] [detail] 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 and the RSVP bandwidth limit counter, taking compression into account. <ul style="list-style-type: none"> The optional detail keyword displays RSVP parameters associated with an interface including bandwidth, admission control, and compression methods.

Examples

This section provides the following example output:

- [Sample Output for the show ip rsvp installed detail Command, page 578](#)
- [Sample Output for the show ip rsvp interface detail Command, page 579](#)

Sample Output for the show ip rsvp installed detail Command

In this example, the **show ip rsvp installed detail** command displays information, including the predicted compression method, its reserved context ID, and the observed bytes saved per packet average, for the admitted flowspec.

```
Router# show ip rsvp installed detail
```

```
RSVP: Ethernet2/1 has no installed reservations
```

```
RSVP: Serial3/0 has the following installed reservations
```

```
RSVP Reservation. Destination is 10.1.1.2. Source is 10.1.1.1,  
Protocol is UDP, Destination port is 18054, Source port is 19156  
Compression: (method rtp, context ID = 1, 37.98 bytes-saved/pkt avg)
```

```
Admitted flowspec:
```

```
Reserved bandwidth: 65600 bits/sec, Maximum burst: 328 bytes, Peak rate: 80K bits/sec  
Min Policed Unit: 164 bytes, Max Pkt Size: 164 bytes
```

```
Admitted flowspec (as required if compression were not applied):
```

```
Reserved bandwidth: 80K bits/sec, Maximum burst: 400 bytes, Peak rate: 80K bits/sec  
Min Policed Unit: 200 bytes, Max Pkt Size: 200 bytes
```

```
Resource provider for this flow:
```

```
WFQ on FR PVC dlci 101 on Se3/0: PRIORITY queue 24. Weight: 0, BW 66 kbps
```

```
Conversation supports 1 reservations [0x1000405]
```

```
Data given reserved service: 3963 packets (642085 bytes)
```

```
Data given best-effort service: 0 packets (0 bytes)
```

```
Reserved traffic classified for 80 seconds
```

```
Long-term average bitrate (bits/sec): 64901 reserved, 0 best-effort
```

```
Policy: INSTALL. Policy source(s): Default
```


Sample Output for the show ip rsvp interface detail Command

In this example, the **show ip rsvp interface detail** command displays the current interfaces and their configured compression parameters.

```
Router# show ip rsvp interface detail

Et2/1:
  Bandwidth:
    Curr allocated: 0 bits/sec
    Max. allowed (total): 1158K bits/sec
    Max. allowed (per flow): 128K bits/sec
    Max. allowed for LSP tunnels using sub-pools: 0 bits/sec
    Set aside by policy (total): 0 bits/sec
  Admission Control:
    Header Compression methods supported:
      rtp (36 bytes-saved), udp (20 bytes-saved)
  Neighbors:
    Using IP encap: 0. Using UDP encap: 0
  Signalling:
    Refresh reduction: disabled
    Authentication: disabled

Se3/0:
  Bandwidth:
    Curr allocated: 0 bits/sec
    Max. allowed (total): 1158K bits/sec
    Max. allowed (per flow): 128K bits/sec
    Max. allowed for LSP tunnels using sub-pools: 0 bits/sec
    Set aside by policy (total): 0 bits/sec
  Admission Control:
    Header Compression methods supported:
      rtp (36 bytes-saved), udp (20 bytes-saved)
  Neighbors:
    Using IP encap: 1. Using UDP encap: 0
  Signalling:
    Refresh reduction: disabled
    Authentication: disabled
```

Troubleshooting Tips

The observed bytes-saved per packet value should not be less than the configured or default value. Otherwise, the flow may be experiencing degraded QoS. To avoid any QoS degradation for future flows, configure a lower bytes-saved per packet value.

Flows may achieve less compressibility than the default RSVP assumes for many reasons, including packets arriving out of order or having different differentiated services code point (DSCP) or precedence values, for example, due to policing upstream within the network.

If compression is enabled on a flow's interface, but the compression prediction was unsuccessful, the reason appears in the output instead of the reserved compression ID and the observed bytes-saved per packet.

Configuration Examples for RSVP Support for RTP Header Compression, Phase 1

This section provides the following configuration example:

- [RSVP Support for RTP Header Compression, Phase 1 Example, page 580](#)

RSVP Support for RTP Header Compression, Phase 1 Example

The following sample configuration shows the compression prediction enabled for flows using UDP and disabled for flows using RTP:

```
Router# configure terminal

Enter configuration commands, one per line.  End with CNTL/Z.

Router(config)# interface Serial3/0

Router(config-if)# ip rsvp admission-control compression predict method udp bytes-saved 16

Router(config-if)# no ip rsvp admission-control compression predict method rtp
```

Use the **show run** command to display all the RSVP configured parameters:

```
Router# show run

2d18h: %SYS-5-CONFIG_I: Configured from console by console

Router# show run int se3/0

Building configuration...

Current configuration : 339 bytes
!
interface Serial3/0
 ip address 10.2.1.1 255.255.0.0
 max-reserved-bandwidth 80
 fair-queue 64 256 8
 serial restart_delay 0
 clock rate 128000
 ip rtp header-compression
 ip rsvp bandwidth
 no ip rsvp admission-control compression predict method rtp
 ip rsvp admission-control compression predict method udp bytes-saved 16
end
```

Additional References

For additional information related to RSVP Support for RTP Header Compression, Phase 1, refer to the following references:

- [Related Documents, page 581](#)
- [Standards, page 581](#)
- [MIBs, page 581](#)
- [RFCs, page 582](#)
- [Technical Assistance, page 582](#)

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, Release 12.2
QoS features—specifically, signalling	Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.2

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 ¹	MIBs Link
<ul style="list-style-type: none"> • RFC 2206, <i>RSVP Management Information Base using SMIPv2</i> 	<p>To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:</p> <p>http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml</p>

1. Not all supported MIBs are listed.

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>

Additional References

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

RFCs ¹	Title
RFC 2205	<i>Resource Reservation Protocol (RSVP)</i>
RFC 2508	<i>Compressing IP/UDP/RTP Headers for Low-Speed Serial Links</i>
RFC 3006	<i>Integrated Services in the Presence of Compressible Flows</i>

1. Not all supported RFCs are listed.

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 new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at <http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm>.

New Commands

- **ip rsvp admission-control compression predict**

Modified Commands

- **debug ip rsvp traffic-control**
- **show ip rsvp installed**
- **show ip rsvp interface**

Glossary

admission control—The process in which a Resource Reservation Protocol (RSVP) reservation is accepted or rejected based on end-to-end available network resources.

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.

compression—The running of a data set through an algorithm that reduces the space required to store or the bandwidth required to transmit the data set.

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.

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.

flowspec—In IPv6, the traffic parameters of a stream of IP packets between two applications.

G.723—A compression technique that can be used for compressing speech or audio signal components at a very low bit rate as part of the H.324 family of standards. This codec has two bit rates associated with it: 5.3 and 6.3 kbps. The higher bit rate is based on ML-MLQ technology and provides a somewhat higher quality of sound. The lower bit rate is based on code excited linear prediction (CELP) compression and provides system designers with additional flexibility. Described in the ITU-T standard in its G-series recommendations.

minCIR—The minimum acceptable incoming or outgoing committed information rate (CIR) for a Frame Relay virtual 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.

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.

RTP—Real-Time Transport Protocol. A protocol that is designed to provide end-to-end network transport functions for applications transmitting real-time data, such as audio, video, or simulation data, over multicast or unicast network services. RTP provides such services as payload type identification, sequence numbering, timestamping, and delivery monitoring to real-time applications.

TCSB—traffic control state block. A Resource Reservation Protocol (RSVP) state that associates reservations with their reserved resources required for admission control.

Tspec—Traffic specification. The traffic characteristics of a data stream from a sender or receiver (included in a Path message).

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.

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.



RSVP Message Authentication

The Resource Reservation Protocol (RSVP) Message Authentication feature provides a secure method to control quality of service (QoS) access to a network.

Feature Specifications for RSVP Message Authentication

Feature History

Release	Modification
12.2(15)T	This feature was introduced.

Supported Platforms

For platforms supported in Cisco IOS Release 12.2(15)T, consult Cisco Feature Navigator.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- [Prerequisites for RSVP Message Authentication, page 585](#)
- [Restrictions for RSVP Message Authentication, page 586](#)
- [Information About RSVP Message Authentication, page 586](#)
- [How to Configure RSVP Message Authentication, page 589](#)
- [Configuration Examples for RSVP Message Authentication, page 600](#)
- [Additional References, page 602](#)
- [Command Reference, page 604](#)
- [Glossary, page 605](#)

Prerequisites for RSVP Message Authentication

Ensure that RSVP is configured on two or more routers within the network before you can use the RSVP Message Authentication feature.

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.

Information About RSVP Message Authentication

To configure RSVP Message Authentication, you need to understand the following concepts:

- [Feature Design of RSVP Message Authentication, page 586](#)
- [Special Considerations for RSVP Message Authentication, page 588](#)
- [Benefits of RSVP Message Authentication, page 588](#)

Feature Design of RSVP Message Authentication

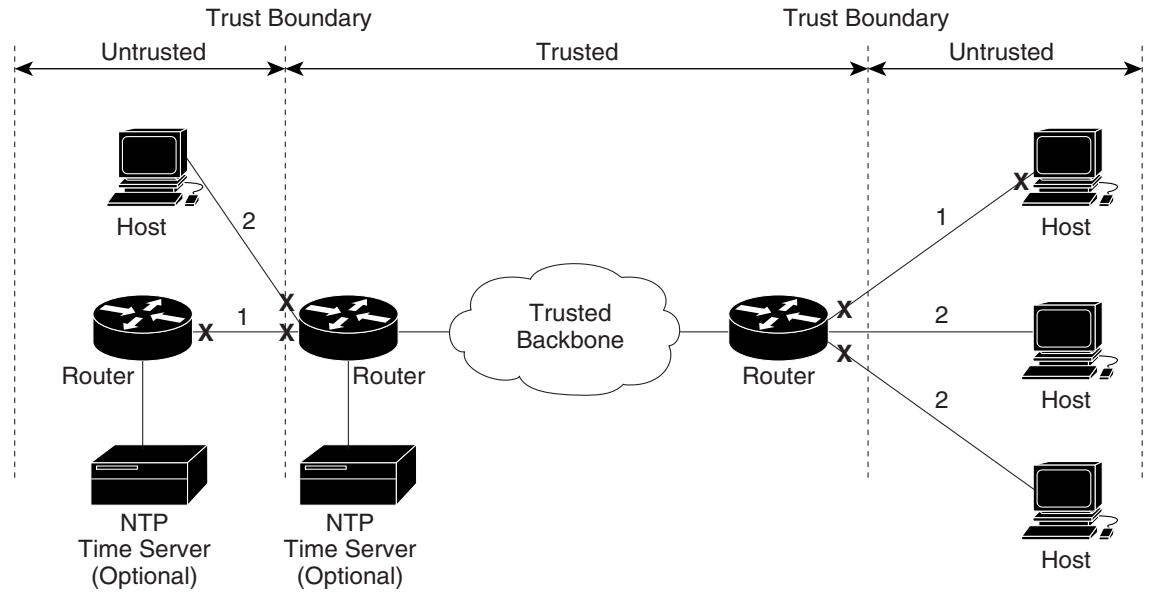
Network administrators need the ability to establish a security domain to control the set of systems that initiate RSVP requests.

The RSVP Message Authentication feature permits neighbors in an RSVP network to use a secure hash to sign all RSVP signalling 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 access control list (ACL).

The signature is accomplished on a per-RSVP-hop basis with an RSVP integrity object in the RSVP message. 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 32](#).

Figure 32 *RSVP Message Authentication Configuration*

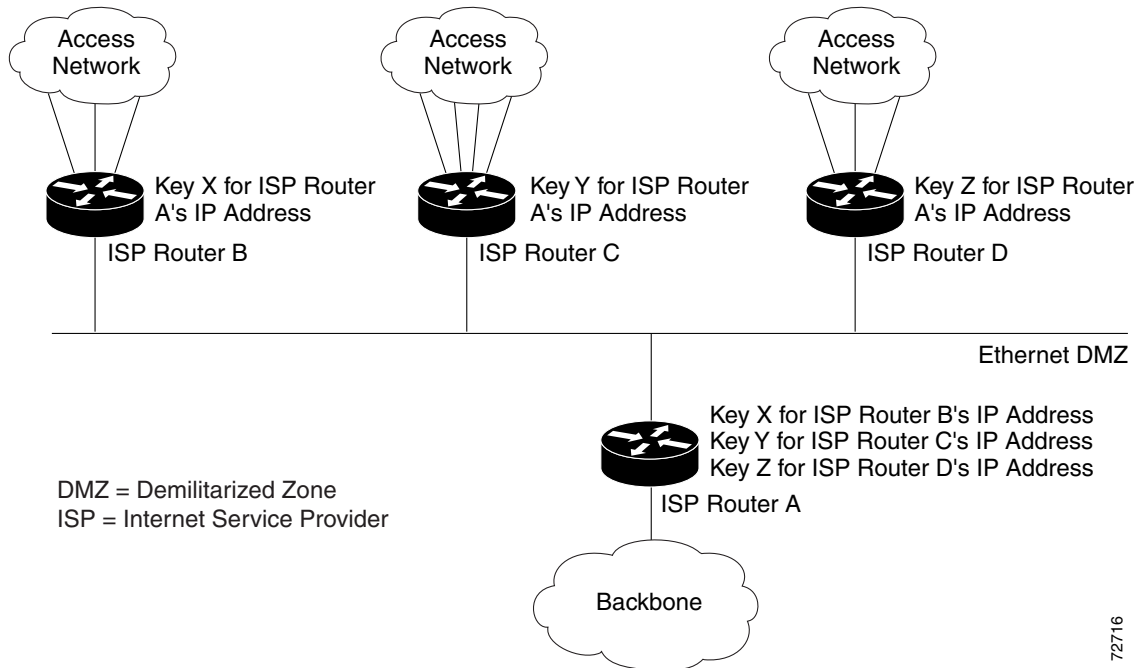


1 = Knows Authentication Key; Accepted
 2 = Does Not Know Key; Rejected
 X = RSVP Authentication Enabled

72677

Special Considerations for RSVP Message Authentication

Figure 33 RSVP Message Authentication in an Ethernet Configuration



In [Figure 33](#), to enable authentication between Internet service providers (ISPs) 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 ISPs 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.

This release does not support the above topology.

You need separate Ethernet networks for A to B, B to A, A to C, C to A, A to D, and D to A. Then configure unique interface keys for them.

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 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.

This section contains the following procedures for a full configuration:

- [Enabling RSVP on an Interface, page 589](#) (required)
- [Configuring an RSVP Authentication Type, page 590](#) (optional)
- [Configuring an RSVP Authentication Key, page 591](#) (required)
- [Enabling RSVP Key Encryption, page 592](#) (optional)
- [Enabling RSVP Authentication Challenge, page 593](#) (optional)
- [Configuring RSVP Authentication Lifetime, page 594](#) (optional)
- [Configuring RSVP Authentication Window Size, page 595](#) (optional)
- [Activating RSVP Authentication, page 596](#) (required)
- [Verifying RSVP Message Authentication, page 597](#) (optional)

This section contains the following tasks for a minimal configuration:

- [Enabling RSVP on an Interface, page 589](#) (required)
- [Configuring an RSVP Authentication Key, page 591](#) (required)
- [Activating RSVP Authentication, page 596](#) (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. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface [<i>type number</i>] Example: Router(config)# interface Ethernet0/0	Enters interface configuration mode. <ul style="list-style-type: none"> The <i>type number</i> argument identifies the interface to be configured.
Step 4	ip rsvp bandwidth [<i>interface-kbps</i>] [<i>single-flow-kbps</i>] Example: Router(config-if)# ip rsvp bandwidth 7500 7500	Enables RSVP on an interface. <ul style="list-style-type: none"> The optional <i>interface-kbps</i> and <i>single-flow-kbps</i> 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. <p>Note Repeat this command for each interface that you want to enable.</p>
Step 5	end Example: Router(config-if)# end	Exits to privileged EXEC mode.

Configuring an RSVP Authentication Type

Perform this task to configure an RSVP authentication type.

SUMMARY STEPS

- enable**
- configure terminal**
- interface** [*type number*]
- ip rsvp authentication type** {md5 | sha-1}
- end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface [<i>type number</i>] Example: Router(config)# interface Ethernet0/0	Enters interface configuration mode. <ul style="list-style-type: none"> The <i>type number</i> argument identifies the interface to be configured.
Step 4	ip rsvp authentication type { md5 sha-1 } Example: Router(config-if)# ip rsvp authentication type sha-1	Specifies the algorithm used to generate cryptographic signatures in RSVP messages. <ul style="list-style-type: none"> The algorithms are md5, the default, and sha-1, which is newer and more secure than md5.
Step 5	end Example: Router(config-if)# end	Exits 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. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface [<i>type number</i>] Example: Router(config)# interface Ethernet0/0	Enters interface configuration mode. <ul style="list-style-type: none"> The <i>type number</i> argument identifies the interface to be configured.
Step 4	ip rsvp authentication key <i>passphrase</i> Example: Router(config-if)# ip rsvp authentication key 11223344	Specifies the data string (key) for the authentication algorithm. <ul style="list-style-type: none"> 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.
Step 5	end Example: Router(config-if)# end	Exits 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

- enable**
- configure terminal**
- key config-key 1** *string*
- end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	key config-key 1 string Example: Router(config)# key config-key 1 11223344	Enables key encryption in the configuration file. <ul style="list-style-type: none"> The <i>string</i> argument can contain up to eight alphanumeric characters.
Step 4	end Example: Router(config)# end	Exits 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. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	interface [<i>type number</i>] Example: Router(config)# interface Ethernet0/0	Enters interface configuration mode. • The <i>type number</i> argument identifies the interface to be configured.
Step 4	ip rsvp authentication challenge Example: Router(config-if)# ip rsvp authentication challenge	Makes RSVP perform a challenge-response handshake when RSVP learns about any new challenge-capable neighbors on a network.
Step 5	end Example: Router(config-if)# end	Exits 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 [<i>type number</i>] Example: Router(config)# interface Ethernet0/0	Enters interface configuration mode. • The <i>type number</i> argument identifies the interface to be configured.

	Command or Action	Purpose
Step 4	<pre>ip rsvp authentication lifetime hh:mm:ss</pre> <p>Example: Router(config-if)# ip rsvp authentication 00:05:00</p>	Controls how long RSVP maintains security associations with RSVP neighbors. <ul style="list-style-type: none"> The default security association for hh:mm:ss is 30 minutes; the range is 1 second to 24 hours.
Step 5	<pre>end</pre> <p>Example: Router(config-if)# end</p>	Exits to privileged EXEC mode.

Configuring RSVP Authentication Window Size

Perform this task to configure 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	<pre>enable</pre> <p>Example: Router> enable</p>	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	<pre>configure terminal</pre> <p>Example: Router# configure terminal</p>	Enters global configuration mode.
Step 3	<pre>interface [type number]</pre> <p>Example: Router(config)# interface Ethernet0/0</p>	Enters interface configuration mode. <ul style="list-style-type: none"> The <i>type number</i> argument identifies the interface to be configured.

	Command or Action	Purpose
Step 4	ip rsvp authentication window-size [n] Example: Router(config-if)# ip rsvp authentication window-size 2	Specifies the maximum number of authenticated messages that can be received out of order. <ul style="list-style-type: none"> The default value is one message; the range is 1 to 64 messages.
Step 5	end Example: Router(config-if)# end	Exits 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. <ul style="list-style-type: none"> 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. <ul style="list-style-type: none"> The <i>type number</i> argument identifies the interface to be configured.

	Command or Action	Purpose
Step 4	<code>ip rsvp authentication</code> Example: Router(config-if)# ip rsvp authentication	Activates RSVP cryptographic authentication.
Step 5	<code>end</code> Example: Router(config-if)# end	Exits 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 [interface-type interface-number] [detail]`
3. `show ip rsvp authentication [detail] [ip-address | hostname]`
4. `show ip rsvp counters [interface interface_unit | summary | neighbor]`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>enable</code> Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<code>show ip rsvp interface [interface-type interface-number] [detail]</code> 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. <ul style="list-style-type: none"> • The optional detail keyword displays the bandwidth, signalling, and authentication parameters.

	Command or Action	Purpose
Step 3	<pre>show ip rsvp authentication [detail] [ip-address hostname]</pre> <p>Example: Router# show ip rsvp authentication detail</p>	<p>Displays the security associations that RSVP has established with other RSVP neighbors.</p> <ul style="list-style-type: none"> 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	<pre>show ip rsvp counters [interface interface_unit summary neighbor]</pre> <p>Example: Router# show ip rsvp counters summary</p>	<p>(Optional) Displays the number of RSVP messages that were sent and received on each interface; shows error counter incrementing whenever an RSVP message is received on an interface with RSVP authentication enabled, but authentication checks failed on that message.</p> <p>Note The error counter can also increment when it receives an error not related to authentication.</p> <ul style="list-style-type: none"> The optional summary keyword shows the cumulative number of RSVP messages sent and received by the platform.

Examples

This section provides the following example output:

- [Sample Output for the show ip rsvp authentication detail Command, page 598](#)
- [Sample Output for the show ip rsvp interface detail Command, page 598](#)

Sample Output for the show ip rsvp authentication detail Command

In this example, the **show ip rsvp authentication detail** command displays information, including IP addresses, interfaces enabled, and configured cryptographic authentication parameters about security associations that RSVP has established with neighbors.

```
Router# show ip rsvp authentication detail

Neighbor: 192.168.101.2  Key ID (hex): 62d0b1140000
Interface: Ethernet0/0  Key type:   Static
Direction: Send        Expiration: 000d 00h 29m 39s
Last seq # sent:
13851245224380071944

Neighbor: 192.168.101.2  Key ID (hex): 62d164fc00000
Interface: Ethernet0/0  Key type:   Static
Direction: Receive     Expiration: 000d 00h 29m 39s
Last valid seq # rcvd:  Challenge:  Not configured
13851246177862811649
```

Sample Output for the show ip rsvp interface detail Command

In this example, the **show ip rsvp interface detail** command displays detailed information, including the cryptographic authentication parameters, for all RSVP-configured interfaces on a router.

**Note**

The authentication key in the following example appears encrypted (<encrypted>). That is because the **key config-key 1 string** command was issued prior to the **show ip rsvp interface detail** command.

```
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
```

Troubleshooting Tips

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 signalling (Path and Resv) messages.

Configuration Examples for RSVP Message Authentication

This section provides the following configuration example:

- [RSVP Message Authentication Example, page 600](#)

RSVP Message Authentication Example

In the following output, the cryptographic authentication parameters, including type, key, challenge, lifetime, 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 the authentication key changes from encrypted to clear text after the **no key config-key 1** command is issued:

```

Router# show run int 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 run
*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

```

Additional References

For additional information related to the RSVP Message Authentication feature, refer to the following references:

- [Related Documents, page 602](#)
- [Standards, page 602](#)
- [MIBs, page 602](#)
- [RFCs, page 603](#)
- [Technical Assistance, page 603](#)

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, Release 12.2
QoS features including signalling, classification, and congestion management	Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.2
Error messages	Cisco IOS System Error Messages

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 standards has not been modified by this feature.	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

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

RFCs ¹	Title
RFC 1321	<i>The MD5 Message Digest Algorithm</i>
RFC 2104	<i>HMAC: Keyed-Hashing for Messaging Authentication</i>
RFC 2205	<i>Resource Reservation Protocol</i>
RFC 2209	<i>RSVP—Version 1 Message Processing Rules</i>
RFC 2401	<i>Security Architecture for the Internet Protocol</i>
RFC 2747	<i>RSVP Cryptographic Authentication</i>
RFC 3174	<i>US Secure Hash Algorithm 1 (SHA1)</i>

1. Not all supported RFCs are listed.

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 new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at <http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm>.

New Commands

- **clear ip rsvp authentication**
- **debug ip rsvp authentication**
- **ip rsvp authentication**
- **ip rsvp authentication challenge**
- **ip rsvp authentication key**
- **ip rsvp authentication lifetime hh:mm:ss**
- **ip rsvp authentication type**
- **ip rsvp authentication window-size**
- **show ip rsvp authentication**

Modified Commands

- **show ip rsvp counters**
- **show ip rsvp interface**

Glossary

admission control—The process in which an RSVP reservation is accepted or rejected based on end-to-end available network resources.

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 signalling 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.

trusted neighbor—A router with authorized access to information.

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.



Configuring RSVP Support for LLQ

This chapter describes the tasks for configuring the RSVP Support for Low Latency Queueing (LLQ) feature.

For complete conceptual information, see the [“RSVP Support for Low Latency Queueing”](#) section on page 481 in the chapter [Signalling Overview](#) in this book.

For a complete description of the RSVP Support for LLQ commands in this chapter, refer to 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.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the [“Finding Additional Feature Support Information”](#) section on page lxix in the [Using Cisco IOS Software for Release 12.4](#) chapter in this book.

RSVP Support for LLQ Configuration Task List

To configure RSVP support for 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 Flow Classification](#) (Required)
- [Enabling RSVP and WFQ](#) (Required)
- [Configuring a Burst Factor](#) (Optional)
- [Configuring a Path](#) (Optional)
- [Configuring a Reservation](#) (Optional)
- [Verifying RSVP Support for LLQ Configuration](#) (Optional)
- [Monitoring and Maintaining RSVP Support for LLQ](#) (Optional)

See the end of this chapter for the section [“RSVP Support for LLQ Configuration Examples.”](#)

Configuring Flow Classification

To configure flow classification, use the following command in global configuration mode:

Command	Purpose
Router#(config)# ip rsvp pq-profile	Specifies the criteria for determining which flows go into the priority queue.

Enabling RSVP and WFQ

To enable RSVP and weighted fair queueing (WFQ), use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# interface s2/0	Enables an interface; for example, serial interface 2/0.
Step 2	Router(config-if)# ip rsvp bandwidth	Enables RSVP on an interface.
Step 3	Router(config-if)# fair-queue	Enables WFQ on an interface with priority queueing (PQ) support.

Configuring a Burst Factor

To configure a burst factor, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# ip rsvp burst policing	Specifies a burst factor on a per-interface basis.

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 input interface, the service type, the average bit rate, and the burst size.

Verifying RSVP Support for LLQ Configuration

To verify RSVP support for LLQ configuration, perform the following steps:

- Step 1** Enter the **show ip rsvp installed** command to display information about interfaces and their admitted reservations. A sample output is shown.

This output shows that Ethernet interface 2/1 has four reservations and serial interface 3/0 has none.

```
Router# show ip rsvp installed
```

```
RSVP:Ethernet2/1
BPS    To           From           Protoc DPort  Sport  Weight Conversation
44K    145.20.0.202 145.10.0.201  UDP   1000  1000  0       264
44K    145.20.0.202 145.10.0.201  UDP   1001  1001  13      266
98K    145.20.0.202 145.10.0.201  UDP   1002  1002  6       265
1K     145.20.0.202 145.10.0.201  UDP   10    10    0       264
RSVP:Serial3/0 has no installed reservations
Router#
```



Note In the sample output, 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 and their current reservations. A sample output is shown.

```
Router# show ip rsvp installed detail
```

```
RSVP:Ethernet2/1 has the following installed reservations
RSVP Reservation. Destination is 145.20.0.202, Source is 145.10.0.201,
  Protocol is UDP, Destination port is 1000, Source port is 1000
  Reserved bandwidth:44K bits/sec, Maximum burst:1K bytes, Peak rate:44K bits/sec
  Resource provider for this flow:
    WFQ on hw idb Se3/0: PRIORITY queue 264. Weight:0, BW 44 kbps
  Conversation supports 1 reservations
  Data given reserved service:316 packets (15800 bytes)
  Data given best-effort service:0 packets (0 bytes)
  Reserved traffic classified for 104 seconds
  Long-term average bitrate (bits/sec):1212 reserved, 0M best-effort
RSVP Reservation. Destination is 145.20.0.202, Source is 145.10.0.201,
  Protocol is UDP, Destination port is 1001, Source port is 1001
  Reserved bandwidth:44K bits/sec, Maximum burst:3K bytes, Peak rate:44K bits/sec
```

```

Resource provider for this flow:
  WFQ on hw idb Se3/0: RESERVED queue 266.  Weight:13, BW 44 kbps
Conversation supports 1 reservations
Data given reserved service:9 packets (450 bytes)
Data given best-effort service:0 packets (0 bytes)
Reserved traffic classified for 107 seconds
Long-term average bitrate (bits/sec):33 reserved, 0M best-effort
RSVP Reservation. Destination is 145.20.0.202, Source is 145.10.0.201,
  Protocol is UDP, Destination port is 1002, Source port is 1002
Router#

```



Note In the sample output, the first flow gets the priority queue (weight = 0) while the second flow does not.

Monitoring and Maintaining RSVP Support for LLQ

To monitor and maintain the RSVP Support for LLQ feature, 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 and their admitted reservations.
Router# show queue interface-type interface-number	Displays queuing configuration and statistics for a particular interface.

RSVP Support for LLQ Configuration Examples

This section provides a configuration example for the RSVP Support for LLQ feature.

For information about configuring RSVP support for LLQ, see the section [“RSVP Support for LLQ Configuration Task List”](#) in this chapter.

In the following example, PQ parameters, including flow rate and burst factor, are defined:

```

Router(config)# ip rsvp pq-profile ?

<1-1048576>  Max Flow Rate (bytes/second)
voice-like  Voice-like flows
<cr>

Router(config)# ip rsvp pq-profile 11000 1500 ?

<100-4000>      Max Peak to Average Ratio (in %)
ignore-peak-value  Ignore the flow's p/r ratio
<cr>

Router(config)# ip rsvp pq-profile 11000 1500 ignore-peak-value
Router(config)# end

```



```
Router# sh run | include pq-profile
ip rsvp pq-profile 11000 1500 ignore-peak-value
In the following example, RSVP is enabled:
```

```
Router# configure terminal

Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface loopback 40
Router(config-if)# ip rsvp bandwidth ?
    <1-10000000> Reservable Bandwidth(KBPS)
    <cr>

Router(config-if)# ip rsvp bandwidth 300 ?
    <1-10000000> Largest Reservable Flow(KBPS)
    <cr>

Router(config-if)# ip rsvp bandwidth 300 30 ?
    <cr>
Router(config-if)# ip rsvp bandwidth 300 30
Router(config-if)# end
```

In the following example, WFQ is enabled:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface e0/1
Router(config-if)# fair-queue
Router(config-if)# fair-queue 64
```

In the following example, a burst factor is configured:

```
Router(config)# interface e3/0
Router(config-if)# ip rsvp burst policing 200
```

In the following example, a path is defined:

```
Router(config)# ip rsvp sender 145.20.20.202 145.10.10.201 udp 10 20
145.10.10.201 loopback 10 80 10
```

In the following example, a reservation is defined:

```
Router(config)# ip rsvp reservation 145.20.20.202 145.10.10.201 udp
10 20 145.20.20.202 1o20 ff load 80 10
```




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 [“RSVP Support for Frame Relay”](#) section on page 483 in the chapter [Signalling Overview](#) in this book.

For a complete description of the RSVP Support for Frame Relay commands in this chapter, refer to 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.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the [“Finding Additional Feature Support Information”](#) section on page lxix in the [Using Cisco IOS Software for Release 12.4](#) chapter in this book.

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)
- [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 <i>name</i>	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 <i>map-class-name</i>	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 { <i>in</i> <i>out</i> } <i>bps</i>	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} <i>bps</i>	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 <i>fragment-size</i>	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# <code>show ip rsvp installed</code>	Displays information about interfaces and their admitted reservations.
Router# <code>show ip rsvp installed detail</code>	Displays additional information about interfaces, DLCIs, and their admitted reservations.
Router# <code>show queueing</code>	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:

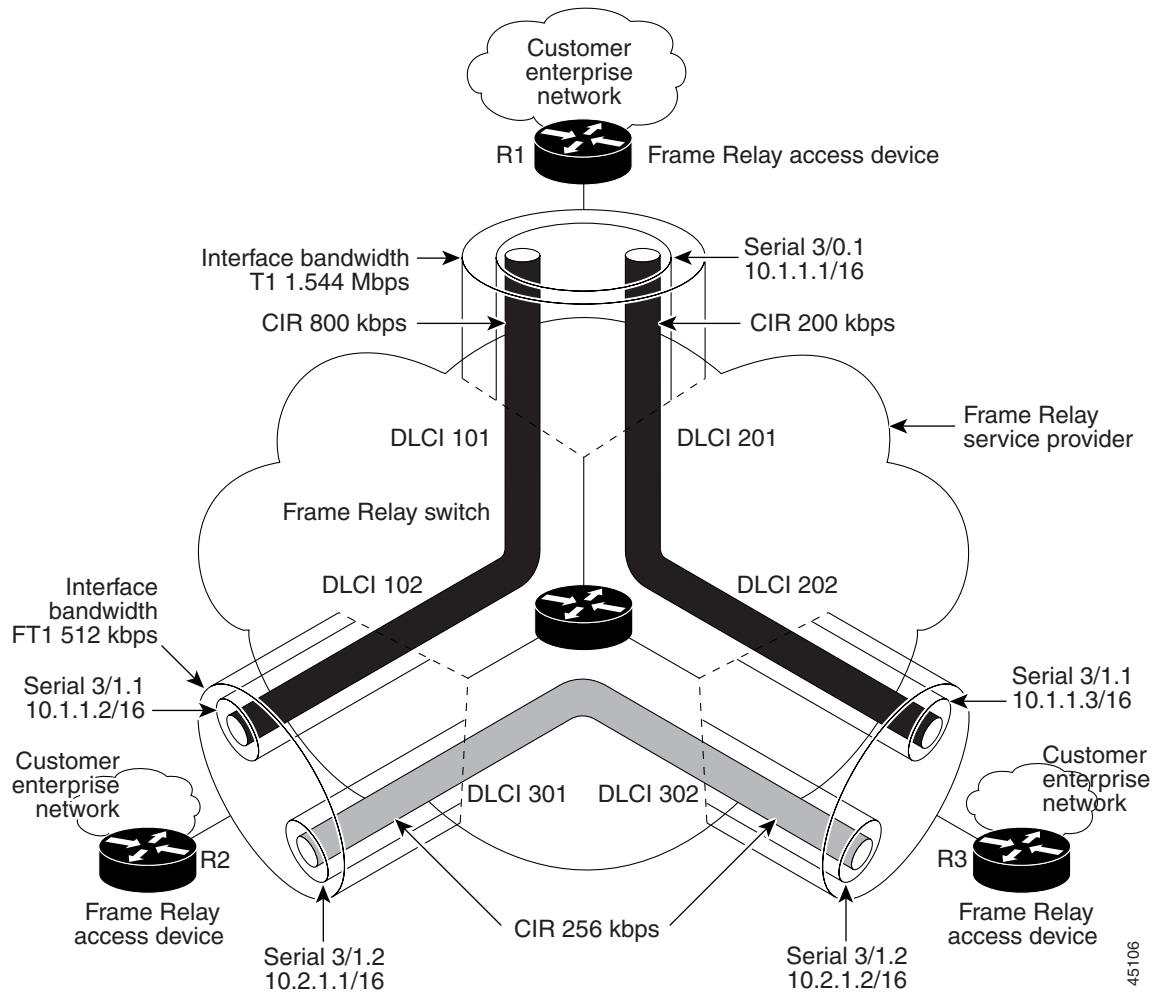
- [Multipoint Configuration Example](#)
- [Point-to-Point Configuration Example](#)

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 34 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 34 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  
!
```

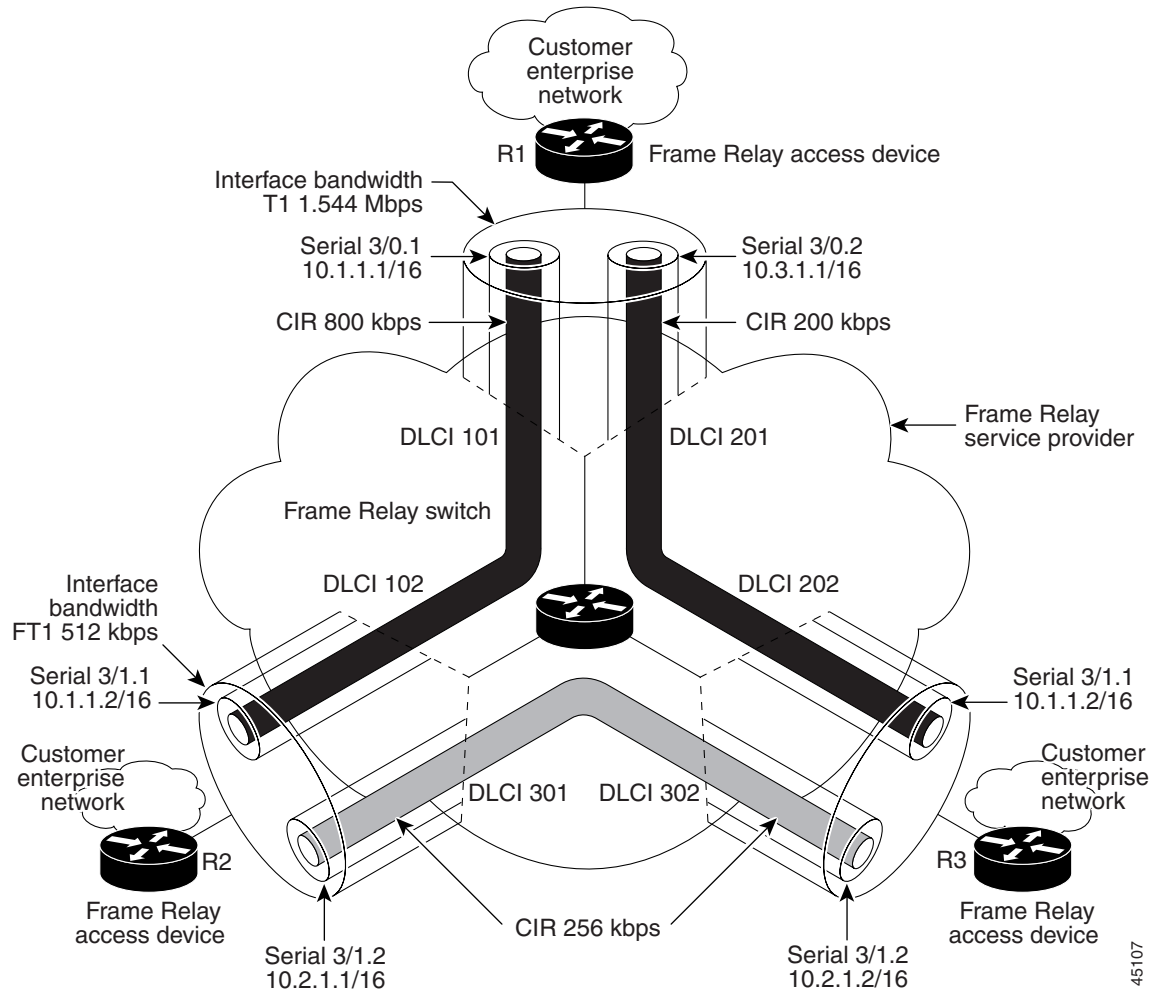
**Note**

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 35 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 35 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.



Configuring RSVP-ATM QoS Interworking

This chapter describes the tasks for configuring the RSVP-ATM QoS Interworking feature, which provides support for Controlled Load Service using RSVP over an ATM core network.

For complete conceptual information, see the [“RSVP-ATM QoS Interworking”](#) section on page 485 in the chapter [Signalling Overview](#) in this book.

For a complete description of the RSVP-ATM QoS Interworking commands in this chapter, refer to 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.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the [“Finding Additional Feature Support Information”](#) section on page lxix in the [Using Cisco IOS Software for Release 12.4](#) chapter in this book.

RSVP-ATM QoS Interworking Configuration Task List

To configure RSVP-ATM QoS Interworking, perform the tasks described in the following sections. Each task is identified as either optional or required.

- [Enabling RSVP and Limiting Reservable Bandwidth](#) (Required)
- [Enabling Creation of SVCs for Reserved Flows](#) (Required)
- [Limiting the Peak Rate Applied to the PCR for SVCs](#) (Optional)
- [Configuring per-VC DWRED](#) (Required)
- [Monitoring RSVP-ATM Configuration for an Interface](#) (Optional)

Before you configure RSVP-ATM QoS Interworking, you must enable and configure the following features:

- Cisco Express Forwarding (CEF) switching (required for RSVP-ATM)
- Distributed CEF (dCEF) (required for per-switched virtual circuit (SVC) DWRED)
- NetFlow services

For information on how to configure these features, refer to the *Cisco IOS Switching Services Configuration Guide* and the *Cisco IOS Switching Services Command Reference*.

The RSVP-ATM QoS Interworking feature does not support Resource Reservation Protocol (RSVP) with multicast.

See the end of this chapter for the section [“RSVP-ATM QoS Interworking Configuration Examples.”](#)

Enabling RSVP and Limiting Reservable Bandwidth

RSVP 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. By default, RSVP is disabled so that it is backward compatible with systems that do not implement RSVP.

To enable RSVP on an interface and restrict the total amount of bandwidth that can be reserved for RSVP and the amount that can be reserved for a single RSVP reservation or flow, use the following command in global configuration mode:

Command	Purpose
Router(config)# ip rsvp bandwidth [interface-kbps] [single-flow-kbps]	Enables RSVP for IP on an interface.

For RSVP over ATM, reservations are needed primarily between routers across the ATM backbone. To limit the number of locations where reservations are made, enable RSVP selectively only at subinterfaces corresponding to router-to-router connections across the backbone network. Preventing reservations being made between the host and the router both limits VC usage and reduces load on the router.

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, the more restrictive of the available bandwidths of the physical interface and the subinterface is applied.

Enabling Creation of SVCs for Reserved Flows

Normally, reservations are serviced when RSVP classifies packets and a queueing mechanism polices the packet. To enable establishment of an SVC to service each new RSVP reservation on the interface, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# ip rsvp svc-required	Enables creation of an SVC for each new reservation made on the interface or subinterface.

To ensure defined QoS, SVCs created in response to RSVP reservation requests are established having QoS profiles consistent with the mapped RSVP flow specifications.

The sustainable cell rate (SCR) of an ATM SVC is equal to the RSVP reservation rate; the maximum burst size (MBS) of an ATM SVC is equal to the RSVP burst size. RSVP attempts to compensate for the cell tax when establishing the reservation so that the requested bandwidth is actually available for IP data traffic.

The sustained cell rate formula is given as follows:

$$r_{atm} = r_{rsvp} * (53/48) * (MPS + DLE + (MPS + DLE) \% 48) / MPS$$

The formula terms used in the equation (and subsequent equations) are described in [Table 4](#), followed by an explanation of how the formula was derived.

Table 4 **SCR Formula Terms**

Term	Definition
r_{atm}	ATM rate (SCR).
r_{rsvp}	RSVP rate.
MPS	Minimum IP packet size, including the IP headers (300 bytes minimum).
DLE	Data-link encapsulation overhead. For RSVP ATM SVCs, ATM adaptation layer 5 (AAL5), Subnetwork Access Protocol (SNAP) encapsulation is used, which imposes a 5-byte encapsulation header on each protocol data unit (PDU).
%	Modulus operator. It yields the integer remainder from an integer division operation. For example, $57 \% 53$ results in 4.
CPS	Cell payload size. The total number of bytes in all the payloads of all the cells required to send a single packet with encapsulation.
UCO	Unused cell overhead (0 to 47).
COMP	Compensation factor. CPS divided by MPS.

There are two reasons for converting from RSVP rate to the ATM cell rate, as follows:

- To account for the ATM encapsulation header overhead and cell header overhead
- To account for the fact that ATM cell sizes are fixed

Because a portion of the last cell is unused, it is possible that a certain IP packet size requires more ATM cell layer bytes.

$MPS + DLE$ is the length of the data packet that needs to be segmented into a number of fixed-length (48-byte payload) pieces that would then be put into a cell and sent.

Because the CPS needs to be greater than or equal to $MPS + DLE$, CPS must be larger than MPS.

CPS can be calculated as follows:

$$CPS = \text{ceil}((MPS + DLE) / 48) * 48$$

where $\text{ceil}(x)$ is the ceiling operator that returns the smallest integer greater than or equal to the real number x . Upon expanding the implementation of the $\text{ceil}(x)$ operator, the expression can be arithmetically transformed into the following equation:

$$CPS = MPS + DLE + (MPS + DLE) \% 48$$

where $(MPS + DLE) \% 48$ yields the integer remainder when $MPS + DLE$ is divided by 48. Because $(MPS + DLE) \% 48$ is equal to the UCO, the equation for CPS can be rewritten as follows:

$$CPS = MPS + DLE + UCO$$

Because the IP bit rate was calculated by considering only the IP data and header (that is, packets of length MPS or larger), the IP bit rate (r_{rsvp}) needs to be multiplied by COMP. According to [Table 4](#), $COMP = CPS/MPS$. Thus:

$$\text{ATM cell payload bit rate} = r_{\text{rsvp}} * COMP = r_{\text{rsvp}} * CPS/MPS$$

When expanded, the ATM cell payload bit rate is as follows:

$$\text{ATM cell payload bit rate} = r_{\text{rsvp}} * (MPS + DLE + UCO) / MPS$$

Each ATM cell has a 5-byte header and a 48-byte payload, resulting in a 53-byte cell. Because the entire cell needs to be accounted for (not just the payload), we need to multiply the equation by a compensation factor of 53/48, which yields the desired equation:

$$r_{\text{atm}} = r_{\text{rsvp}} * (53/48) * (MPS + DLE + UCO) / MPS$$

Thus, the SCR of the SVC created to carry the RSVP flow is calculated by the following formula:

$$r_{\text{atm}} = r_{\text{rsvp}} * (53/48) * (MPS + DLE + (MPS + DLE) \% 48) / MPS$$

The ATM peak cell rate (PCR) is derived using the same formula as the cell rate formula. It is either based on the maximum line rate of the ATM interface or on a configured maximum.

The maximum burst size of the SVC is derived by the following formula:

$$r_{\text{atm}} = r_{\text{rsvp}} * (MPS + DLE + UCO) / (MPS * 48)$$

Note that the actual PCR, SCR, and MBS will be slightly larger than these formulas indicate.

See the task “[Limiting the Peak Rate Applied to the PCR for SVCs](#)” for information on setting the PCR of the ATM SVC.

Each new RSVP reservation causes establishment of a new SVC. If an existing reservation is refreshed, no new signalling is needed. If the reservation is not refreshed and it times out, the SVC is torn down. If the reservation is refreshed but the RSVP flowspec has changed, the existing SVC is torn down and a new one with the correct QoS parameters is established.

Limiting the Peak Rate Applied to the PCR for SVCs

To set a limit on the PCR of reservations for all new RSVP SVCs established on the current interface or any of its subinterfaces, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# ip rsvp atm-peak-rate-limit <i>limit</i>	Configures the peak rate limit for new RSVP SVCs on an interface or subinterface.

For Controlled Load Service, the nominal peak rate is not defined and is taken as infinity. Consequently, the PCR is set to the available line rate. However, you can use the **ip rsvp atm-peak-rate-limit** command to further limit the PCR to a specific value on a per-interface basis.

Configuring per-VC DWRED

To configure Distributed Weighted Random Early Detection (DWRED) with per-VC DWRED enabled as a drop policy at the interface level for a specific DWRED group, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# random-detect [attach <i>group-name</i>]	Configures interface-level per-VC DWRED for a specific DWRED group.

The per SVC-DWRED drop policy ensures that packets that match reservations and conform to the appropriate token bucket have the highest priority. Attaching DWRED group definitions to the interface to support per-VC DWRED drop policy 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 the RSVP. This drop policy meets the loss requirements of controlled load called for by the Controlled Load Service class.

To meet the loss goals of controlled load, it is necessary to ensure that if packets must be dropped, best-effort packets are dropped first. Given that packets matching reservations and conforming to the appropriate token bucket will have the highest precedence, per-SVC DWRED is used as the drop policy.

**Note**

In order to use per-SVC DWRED, dCEF must be configured on the router. For information on how to configure dCEF, refer to the *Cisco IOS Switching Services Configuration Guide* and the *Cisco IOS Switching Services Command Reference*.

Monitoring RSVP-ATM Configuration for an Interface

To display the peak rate limit for the interface, the IP Precedence and ToS bit values configured for packets that conform to and exceed the flowspec, and other RSVP-related information for the interface, such as whether the interface has been configured to establish SVCs to service reservation request messages and whether RSVP is enabled to attach itself to NetFlow, use the following commands in EXEC mode, as needed:

Command	Purpose
Router# show ip rsvp atm-peak-rate-limit [interface]	Displays the current peak rate limit set for an interface, if any.
Router# show ip rsvp interface [interface-type interface-number]	Displays RSVP-related interface information.
Router# show ip rsvp {precedence tos} [interface]	Displays the IP Precedence bit values and type of service (ToS) bit values to be used to mark the ToS byte of the IP headers of all packets in an RSVP reserved path that conform to or exceed the RSVP flowspec for a given interface.

RSVP-ATM QoS Interworking Configuration Examples

This section provides RSVP-ATM QoS Interworking configuration examples.

For information about configuring RSVP-ATM QoS Interworking, see the section “[RSVP-ATM QoS Interworking Configuration Task List](#)” in this chapter.

The following example configures two Cisco 7500 series routers that connect over an ATM core network through a permanent virtual circuit (PVC) and multiple SVCs. As depicted in [Figure 36](#), Router A is connected to the ATM core network downstream; upstream it is connected across an Ethernet connection to the RSVP sender host system. Router B is connected upstream to the ATM core network and downstream across an Ethernet connection to the RSVP receiver host.

The example configuration shows three PVCs, two of which are required by ATM. One of the PVCs is used for RSVP-ATM QoS Interworking. It is used for transmission of best-effort traffic and to control traffic such as routing and RSVP messages. The ATM SVCs are established in response to reservation request messages in order to service those requests.

Figure 36 Example RSVP-ATM QoS Interworking Configuration



Router A Configuration

The following portion of the example configures Router A in global configuration mode. It enables CEF, which must be turned on before the RSVP-ATM QoS Interworking feature can be enabled at the interface configuration level.

```

RouterA# config terminal
RouterA(config)# ip routing
RouterA(config)# ip cef

```

The following segment of the configuration for Router A configures ATM interface 2/1/0. The **ip route-cache flow** command enables NetFlow on the interface. If you do not enter the **ip RSVP bandwidth** command before the **ip RSVP svc-required** command, a warning is issued requesting that you change the order of the commands.

The **ip RSVP bandwidth** command enables RSVP on the interface with default values for bandwidth allocation to RSVP. The **ip RSVP svc-required** command enables establishment of an SVC to service each new RSVP reservation on the interface. The **ip RSVP tos** and **ip RSVP precedence** commands configure conform and exceed values to be used for setting the ToS and IP Precedence bits of packets that either conform to or exceed the RSVP flowspec. (Note that once set, the ToS and IP Precedence bit values remain for the duration of the packet.)

You should configure the **ip route-cache flow** command only on the input interfaces of a router on whose output interfaces you configured the **ip RSVP svc-required** command.

```

RouterA(config)# interface ATM2/1/0
RouterA(config-if)# no shut
RouterA(config-if)# ip address 145.5.5.1 255.255.255.0
RouterA(config-if)# no ip proxy
RouterA(config-if)# no ip redirects
RouterA(config-if)# ip route-cache
RouterA(config-if)# ip mroute-cache
RouterA(config-if)# ip route-cache flow
RouterA(config-if)# no ip mroute-cache
RouterA(config-if)# ip route-cache cef
RouterA(config-if)# atm pvc 1 0 5 qsaal
RouterA(config-if)# atm pvc 2 0 16 ilmi
RouterA(config-if)# atm esi-address 111111111151.00
RouterA(config-if)# pvc pvc12 0/51
RouterA(config-if-atm-vc)# inarp 5
RouterA(config-if-atm-vc)# broadcast
RouterA(config-if-atm-vc)# exit
RouterA(config-if)# ip RSVP bandwidth
RouterA(config-if)# ip RSVP svc-required
RouterA(config-if)# ip RSVP tos conform 4
RouterA(config-if)# ip RSVP precedence conform 3 exceed 2

```

The following portion of the configuration configures Ethernet interface 0/1 on Router A that is used for the connection between the sender host and Router A. RSVP is enabled on the interface with default bandwidth allocations.

```

RouterA(config)# interface Ethernet0/1
RouterA(config-if)# ip address 145.1.1.1 255.255.255.0
RouterA(config-if)# no ip proxy

```

```
RouterA(config-if)# no ip redirects
RouterA(config-if)# no shut
RouterA(config-if)# ip route-cache
RouterA(config-if)# ip mroute-cache
RouterA(config-if)# ip route-cache flow
RouterA(config-if)# no ip mroute-cache
RouterA(config-if)# ip route-cache cef
RouterA(config-if)# fair
RouterA(config-if)# ip rsvp bandwidth
```

The following section displays configuration for Router A after the preceding commands were used to configure it:

```
RouterA# write terminal

Current configuration:
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname RouterA
boot system tftp rsp-jv-mz 171.69.209.28
enable password
!
ip subnet-zero
ip cef
interface Ethernet0/1
 ip address 145.1.1.1 255.255.255.0
 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
 ip rsvp bandwidth 7500 7500
 no ip route-cache cef
 no ip mroute-cache
 fair-queue 64 256 1000
!
interface ATM2/1/0
 ip address 145.5.5.1 255.255.255.0
 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
 ip rsvp bandwidth 112320 112320
 ip rsvp svc-required
 ip route-cache flow
 ip rsvp tos conform 4
 ip rsvp precedence conform 3 exceed 2
 no ip route-cache cef
 no ip route-cache distributed
 no ip mroute-cache
 atm pvc 1 0 5 qsaal
 atm pvc 2 0 16 ilmi
 atm esi-address 111111111151.00
 pvc pvc12 0/51
  inarp 5
  broadcast
!
```

Router B Configuration

Router B is configured similarly to Router A. In the following global configuration portion of the example, Router B is configured so that CEF is enabled before the RSVP-ATM QoS Interworking feature can be enabled.

```
RouterB# config terminal
RouterB(config)# ip routing
RouterB(config)# ip cef
```

The following segment of the configuration for Router B configures ATM interface 3/0/0. The **ip rsvp bandwidth** command enables RSVP and the **ip route-cache flow** command enables NetFlow on the interface. The **ip rsvp svc-required** command enables the RSVP-ATM QoS Interworking feature, allowing for the establishment of an SVC to service each new RSVP reservation on the interface.

```
RouterB(config)# interface ATM3/0/0
RouterB(config-if)# atm pvc 1 0 5 qsaal
RouterB(config-if)# atm pvc 2 0 16 ilmi
RouterB(config-if)# atm esi-address 11111111152.00
RouterB(config-if)# pvc pvc12 0/52
RouterB(config-if-atm-vc)# inarp 5
RouterB(config-if-atm-vc)# broadcast
RouterB(config-if-atm-vc)# exit
RouterB(config-if)# ip rsvp bandwidth
RouterB(config-if)# ip route-cache flow
RouterB(config-if)# ip rsvp svc-required
```

The following portion of the configuration configures the Ethernet interface on Router B. This interface is used for the connection between the receiver host and Router B. RSVP is enabled on the interface.

```
RouterB(config)# interface Ethernet0/2
RouterB(config-if)# no shut
RouterB(config-if)# ip address 145.4.4.2 255.255.255.0
RouterB(config-if)# no ip proxy
RouterB(config-if)# no ip redirects
RouterB(config-if)# ip route-cache
RouterB(config-if)# ip mroute-cache
RouterB(config-if)# ip route-cache flow
RouterB(config-if)# no ip mroute-cache
RouterB(config-if)# ip route-cache cef
RouterB(config-if)# fair
RouterB(config-if)# ip rsvp bandwidth
RouterB(config-if)# end
RouterB(config)# ip routing
RouterB(config)# router eigrp 17
RouterB(config-router)# network 145.5.5.0
RouterB(config-router)# network 145.4.4.0
```

The following section displays configuration for Router B after the preceding commands were used to configure it:

```
RouterB# write terminal

Current configuration:
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname RouterB
!
!
boot system tftp rsp-jv-mz 171.69.209.28
```

```
enable password
!
ip subnet-zero
ip cef distributed
interface Ethernet0/2
 ip address 145.4.4.2 255.255.255.0
 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
 ip rsvp bandwidth 7500 7500
 ip route-cache flow
 no ip mroute-cache
 fair-queue 64 256 1000
!
interface ATM3/0/0
 ip address 145.5.5.2 255.255.255.0
 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
 ip rsvp bandwidth 112320 112320
 ip rsvp svc-required
 ip route-cache flow
 no ip route-cache cef
 no ip route-cache distributed
 no ip mroute-cache
 atm pvc 1 0 5 qsaal
 atm pvc 2 0 16 ilmi
 atm esi-address 11111111152.00
 pvc pvc12 0/52
  inarp 5
  broadcast
!
```




Configuring COPS for RSVP

This chapter describes the tasks for configuring the COPS for RSVP feature. Common Open Policy Service (COPS) is a protocol for communicating network traffic policy information to network devices. Resource Reservation Protocol (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.

For complete conceptual information, see the [“COPS for RSVP” section on page 488](#) in the chapter [Signalling Overview](#) in this book.

For a complete description of the COPS for RSVP commands in this chapter, refer to 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.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the [“Finding Additional Feature Support Information” section on page lxix](#) in the [Using Cisco IOS Software for Release 12.4](#) chapter in this book.

COPS for RSVP Configuration Task List

To configure COPS for RSVP, 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.

- [Specifying COPS Servers and Enabling COPS for RSVP](#) (Required)
- [Restricting RSVP Policy to Specific Access Control Lists](#) (Optional)
- [Rejecting Unmatched RSVP Messages](#) (Optional)
- [Confining Policy to PATH and RESV Messages](#) (Optional)
- [Retaining RSVP Information After Losing Connection with the COPS Server](#) (Optional)
- [Reporting the Results of Outsourcing and Configuration Decisions](#) (Optional)
- [Verifying the Configuration](#) (Optional)

See the end of this chapter for the section [“COPS for RSVP Configuration Examples.”](#)

Specifying COPS Servers and Enabling COPS for RSVP

To specify COPS servers and enable COPS for RSVP, use the following commands beginning in interface configuration mode:

	Command	Purpose
Step 1	Router(config-if)# configure terminal	Enters global configuration mode.
Step 2	Router(config)# ip rsvp policy cops servers 161.44.130.168 161.44.129.6	Tells the router to request RSVP policy decisions from the first server listed, and if that fails to connect, from the next server listed. Also enables a COPS-RSVP client on the router.

Restricting RSVP Policy to Specific Access Control Lists

To restrict RSVP policy to specific access control lists (ACLs), use the following commands beginning in interface configuration mode:

	Command	Purpose
Step 1	Router(config-if)# configure terminal	Enters global configuration mode.
Step 2	Router(config)# ip rsvp policy cops 40 160 servers 161.44.130.164 161.44.129.2	Tells the router to apply RSVP policy to messages that match ACLs 40 and 160, and specifies the servers for those sessions.

Rejecting Unmatched RSVP Messages

To reject unmatched RSVP messages, use the following commands beginning in interface configuration mode:

	Command	Purpose
Step 1	Router(config-if)# configure terminal	Enters global configuration mode.
Step 2	Router(config)# ip rsvp policy default-reject	Tells the router to reject unmatched PATH and RESV messages, instead of just letting them pass through unadjudicated.

Confining Policy to PATH and RESV Messages

To confine policy to PATH and RESV messages, use the following commands beginning in interface configuration mode:

	Command	Purpose
Step 1	Router(config-if)# configure terminal	Enters global configuration mode.
Step 2	Router(config)# ip rsvp policy cops minimal	Tells the router to adjudicate only PATH and RESV messages, and to accept and pass onward PATH ERROR, RESV ERROR, and RESV CONFIRM messages.

Retaining RSVP Information After Losing Connection with the COPS Server

To retain RSVP information after losing connection with the COPS server, use the following commands beginning in interface configuration mode:

	Command	Purpose
Step 1	Router(config-if)# configure terminal	Enters global configuration mode.
Step 2	Router(config)# ip rsvp policy cops timeout 600	Tells the router to hold policy information for 10 minutes (600 seconds) while attempting to reconnect to a COPS server.

Reporting the Results of Outsourcing and Configuration Decisions

To report the results of outsourcing and configuration decisions, use the following commands beginning in interface configuration mode:

	Command	Purpose
Step 1	Router(config-if)# configure terminal	Enters global configuration mode.
Step 2	Router(config)# ip rsvp policy cops report-all	Tells the router to report to the Policy Decision Point (PDP) the success or failure of outsourcing and configuration decisions.

Verifying the Configuration

To verify the COPS for RSVP configuration, use the following commands in EXEC mode, as needed:

Command	Purpose
Router# show cops servers	Displays server addresses, port, state, keepalives, and policy client information.
Router# show ip rsvp policy cops	Displays policy server addresses, ACL IDs, and client/server connection status.
Router# show ip rsvp policy	Displays ACL IDs and their connection status.

COPS for RSVP Configuration Examples

The following sections provide COPS for RSVP configuration examples:

- [COPS Server Specified Example](#)
- [RSVP Behavior Customized Example](#)
- [Verification of the COPS for RSVP Configuration Example](#)

For information about configuring COPS for RSVP, see the section “[COPS for RSVP Configuration Task List](#)” in this chapter.

COPS Server Specified Example

The following example specifies the COPS server and enables COPS for RSVP on the server. Both of these functions are accomplished by using the **ip rsvp policy cops** command. By implication, the default settings for all remaining COPS for RSVP commands are accepted.

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# ip rsvp policy cops servers 161.44.130.168 161.44.129.6
Router(config)# exit
```

RSVP Behavior Customized Example

Once the COPS server has been specified and COPS for RSVP has been enabled, the remaining COPS for RSVP commands can be used to customize the COPS for RSVP behavior of the router. The following example uses the remaining COPS for RSVP commands to customize the RSVP behavior of the router:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# ip rsvp policy cops 40 160 servers 161.44.130.168 161.44.129.6
Router(config)# ip rsvp policy default-reject
Router(config)# ip rsvp policy cops minimal
Router(config)# ip rsvp policy cops timeout 600
Router(config)# ip rsvp policy cops report-all
Router(config)# exit
```

Verification of the COPS for RSVP Configuration Example

The following examples display three views of the COPS for RSVP configuration on the router, which can be used to verify the COPS for RSVP configuration.

This example displays the policy server address, state, keepalives, and policy client information:

```
Router# show cops servers

COPS SERVER: Address: 161.44.135.172. Port: 3288. State: 0. Keepalive: 120 sec
Number of clients: 1. Number of sessions: 1.
COPS CLIENT: Client type: 1. State: 0.
```

This example displays the policy server address, the ACL ID, and the client/server connection status:

```
Router# show ip rsvp policy cops

COPS/RSVP entry. ACLs: 40 60
PDPs: 161.44.135.172
Current state: Connected
Currently connected to PDP 161.44.135.172, port 0
```

This example displays the ACL ID numbers and the status for each ACL ID:

```
Router# show ip rsvp policy

Local policy: Currently unsupported
COPS:
ACLs: 40 60 . State: CONNECTED.
ACLs: 40 160 . State: CONNECTING.
```



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 the [“Subnetwork Bandwidth Manager” section on page 494](#) in the chapter [Signalling Overview](#) in this book.

For a complete description of the SBM commands in this chapter, refer to 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.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the [“Finding Additional Feature Support Information” section on page lxix](#) in the [Using Cisco IOS Software for Release 12.4](#) chapter in this book.

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:

Command	Purpose
Router(config-if)# ip rsvp dsbm candidate [<i>priority</i>]	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 <i>kBps</i>	Configures the average rate, in kbps, for the DSBM candidate.
Router(config-if)# ip rsvp dsbm non-resv-send-limit burst <i>kilobytes</i>	Configures the maximum burst size, in KB, for the DSBM candidate.
Router(config-if)# ip rsvp dsbm non-resv-send-limit peak <i>kBps</i>	Configures the peak rate, in kbps, for the DSBM candidate.
Router(config-if)# ip rsvp dsbm non-resv-send-limit min-unit <i>bytes</i>	Configures the minimum policed unit, in bytes, for the DSBM candidate.
Router(config-if)# ip rsvp dsbm non-resv-send-limit max-unit <i>bytes</i>	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# <code>show ip RSVP sbm [detail] [interface]</code>	Displays information about an SBM configured for a specific RSVP-enabled interface or for all 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
  IP Address:10.2.2.150
  DSBM candidate:yes
  Priority:100
  Non Resv Send Limit
    Rate:500 Kbytes/sec
    Burst:1000 Kbytes
    Peak:500 Kbytes/sec
    Min Unit:unlimited
    Max Unit:unlimited
Current DSBM
  IP Address:10.2.2.150
  I Am DSBM:yes
  Priority:100
  Non Resv Send Limit
    Rate:500 Kbytes/sec
    Burst:1000 Kbytes
    Peak:500 Kbytes/sec
    Min 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
```




Part 6: 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”](#) chapter.

Frame Relay Fragmentation

Cisco has developed the following three methods of performing Frame Relay fragmentation:

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

For more information about Frame Relay fragmentation, see the [Cisco IOS Wide-Area Networking Configuration Guide](#), Release 12.4.

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 “” chapter.



Reduction of Latency and Jitter for Real-Time Traffic Using Multilink PPP

This part consists of the following:

- [Reducing Latency and Jitter Using Multilink PPP Roadmap](#)
- [Reducing Latency and Jitter for Real-Time Traffic Using Multilink PPP](#)
- [Using Multilink PPP over ATM Links](#)
- [Using Multilink PPP over Dialer Interface Links](#)
- [Using Multilink PPP over Frame Relay](#)
- [Using Multilink PPP over Serial Interface Links](#)





Reducing Latency and Jitter Using Multilink PPP Roadmap

This roadmap lists the features documented in the *Cisco IOS Quality of Service Solutions* configuration guide and maps them to the modules in which they appear.

Roadmap History

This roadmap was first published on May 2, 2005, and last updated on May 2, 2005.

Feature and Release Support

[Table 5](#) lists Multilink PPP feature support (as it relates to reducing latency and jitter) for the following Cisco IOS software release trains:

- [Cisco IOS Releases 12.2T, 12.3, and 12.3T](#)

Only features that were introduced or modified in Cisco IOS Release 12.2(1) or later appear in the table. *Not all features may be supported in your Cisco IOS software release.*

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 Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.



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 Supported Multilink PPP-Related Features

Release	Feature Name	Feature Description	Where Documented
Cisco IOS Releases 12.2T, 12.3, and 12.3T			
12.2(8)T	Distributed Link Fragmentation and Interleaving Over Leased Lines	<p>The Distributed Link Fragmentation and Interleaving over Leased Lines feature extends distributed link fragmentation and interleaving functionality to leased lines.</p> <p>This feature was extensively rewritten from the perspective of using Multilink PPP for link fragmentation and interleaving over ATM, Frame Relay, serial, and dialer interface links.</p>	<p>Using Multilink PPP over ATM Links</p> <p>Using Multilink PPP over Frame Relay</p> <p>Using Multilink PPP over Serial Interface Links</p> <p>Using Multilink PPP over Dialer Interface Links</p>
12.2(4)T	Distributed Link Fragmentation and Interleaving for Frame Relay and ATM Interfaces on Cisco 7500 Series Routers	<p>The Distributed Link Fragmentation and Interleaving (dLFI) for Frame Relay and ATM Interfaces on Cisco 7500 Series Routers feature extends link fragmentation and interleaving functionality to versatile interface processor (VIP)-enabled Cisco 7500 series routers.</p> <p>This feature was extensively rewritten from the perspective of using Multilink PPP for link fragmentation and interleaving over ATM, Frame Relay, serial, and dialer interface links.</p>	<p>Using Multilink PPP over ATM Links</p> <p>Using Multilink PPP over Frame Relay</p> <p>Using Multilink PPP over Serial Interface Links</p> <p>Using Multilink PPP over Dialer Interface Links</p>



Reducing Latency and Jitter for Real-Time Traffic Using Multilink PPP

This module contains information about reducing latency and jitter for real-time traffic on your network. One Cisco mechanism for reducing latency and jitter for real-time traffic is Multilink PPP (MLP), also known as Multilink. This module contains conceptual information about Multilink and describes how Multilink PPP can be used with network peers to reduce latency and jitter for real-time traffic on your network.

Module History

This module was first published on May 2, 2005, and last updated on May 2, 2005.

Contents

- [Information About Multilink, page 651](#)
- [Where to Go Next, page 655](#)
- [Additional References, page 656](#)
- [Glossary, page 657](#)

Information About Multilink

Before configuring Multilink, you should understand the following concepts:

- [Restrictions for Multilink, page 652](#)
- [Multilink Functionality, page 652](#)
- [Multiclass Multilink PPP, page 654](#)
- [Distributed Multilink PPP, page 655](#)

Restrictions for Multilink

Multilink uses first-in first-out (FIFO) queuing to queue and interleave packets. Alternative mechanisms such as low latency queuing (LLQ), weighted fair queuing (WFQ), or class-based weighted fair queuing (CBWFQ) may be used. If you want to use one of these alternative mechanisms, enable it before configuring Multilink. For more information about queuing mechanisms, see the [Cisco IOS Quality of Service Solutions Configuration Guide](#), Release 12.3.

Multilink Functionality

At the top level, Multilink provides packet interleaving, packet fragmentation, and packet resequencing across multiple logical data links. The packet interleaving, packet fragmentation, and packet resequencing is 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).

Multilink Interleaving

Multilink interleaving is based upon two other integral Multilink activities:

- The ability to fragment packets (or datagrams)
- The ability to multiplex at least two independent data streams

The term interleaving comes from the latter activity, that is, the interleaving of two (or more) independent data streams which are processed independently by the network peer.

Multilink interleaving is a mechanism that allows short, real-time (that is, time-sensitive) packets to be transmitted to a network peer within a certain amount of time (the “delay budget”). To accomplish this task, Multilink interleaving interrupts the transmission of large non-time-sensitive (sometimes referred to as “bulk”) datagrams or packets in favor of transmitting the time-sensitive packet. Once the real-time packet is sent, the system resumes sending the bulk packet.

An example may help to illustrate the concept of delay budget. The network starts transmitting a large datagram to a network peer. This large datagram takes 500 milliseconds (ms) to transmit. Three milliseconds later (while the large datagram is still being transmitted), a voice packet arrives in the transmit queue. By the time the large datagram is completely transmitted (497 ms later) the voice packet (which is highly time-sensitive) is subject to unacceptable delay (that is, its delay budget is exceeded).

Multilink interleaving is particularly useful for applications where too much latency (that is, delay) is detrimental to the function of the application, such as Voice over IP (VoIP). However, it is also beneficial for other forms of “interactive” data, such as Telnet packets where the Telnet packets echo the keystrokes entered by the user at a keyboard.

Multilink Fragmentation

With Multilink fragmentation, the large datagram is fragmented (“chopped”) into a number of small packet fragments, Multilink headers are added to the packet fragments, and the packet fragments are transmitted individually to a network peer.

When interleaving is enabled, the packet fragments are small enough so that the time it takes to transmit them does not exceed the time budgeted for transmitting the real-time (time-sensitive) data packet. The real-time data packets are interleaved between the fragments of the large datagram.

Each time Multilink prepares to send another data packet fragment or frame to the receiving network peer, Multilink first checks to see if a real-time (time-sensitive) packet has arrived in the transmit queue. If so, the high-priority packet is sent first before sending the next fragment from the large datagram.

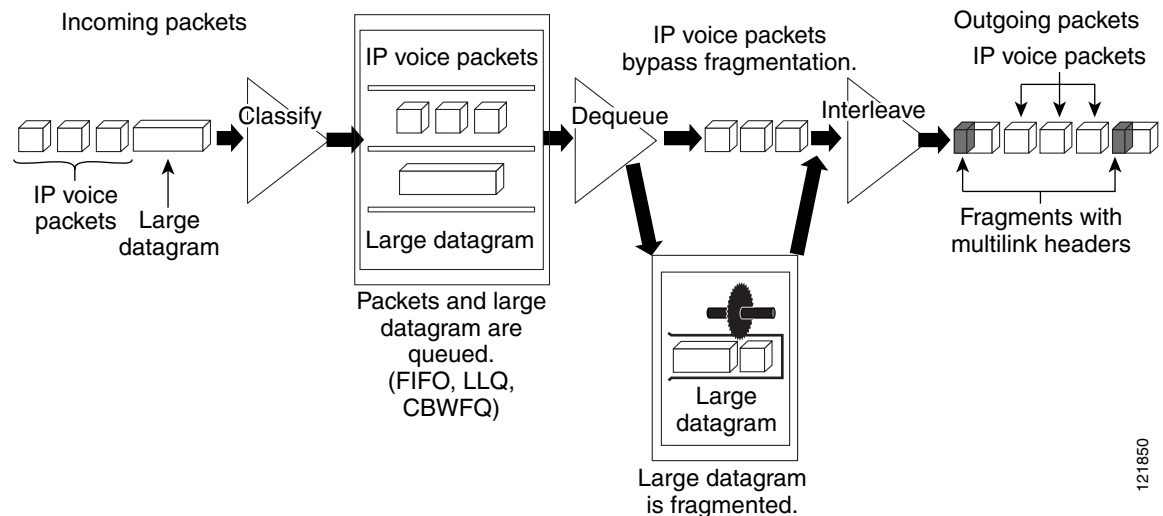
The time delay before the priority packets arrive at the receiving network link is subject to the usual serialization delays at the network link level. That is, any other data already being transmitted has to be finished before the priority packet can be sent. By segmenting long datagrams into small fragments, and checking for newly arrived priority frames between fragments, the priority frame is delayed only by the time it takes to transmit a previously queued fragment rather than a complete large datagram.

Thus, the maximum size of the fragments dictates the responsiveness for insertion of priority packets into the stream. The fragment size can be tuned by adjusting the fragment delay with the **ppp multilink fragment delay** command.

To ensure correct order of transmission and reassembly (which occurs later), multilink headers are added to the large datagram fragments after the packets are dequeued and ready to be sent.

Figure 37 is a simplified illustration of how Multilink fragments and interleaves packets.

Figure 37 *Multilink Fragmentation and Interleaving*



In Figure 37, both IP voice packets and a large datagram arrive at the interface from a single network link. Your network may have multiple links. The IP voice packet and large datagram are queued according to their classification. The large datagram is fragmented (the IP voice packets are not). The IP voice packets are interleaved between the fragments of the large datagram, to which multilink headers are added.

Packets Dequeued and Transmitted

When the large datagram is dequeued, and space becomes available on a member link, Multilink takes a fragment from the original large datagram and transmits the fragments over that link. If an IP voice packet (or other real-time packet) arrives at the transmit queue before Multilink has completely sent the datagram fragment, the next time a link is available to send more packets, Multilink will dequeue and send the high-priority packet. The high-priority packet will be sent instead of another fragment from the large datagram.

Multilink Resequencing

A multilink bundle is a virtual Point-to-Point Protocol (PPP) connection or session over a network link. A multilink bundle at the transmitting end of the network sends the fragments to a multilink bundle on the receiving end of the network link.

The multilink bundle at the receiving end of the network accepts the fragments from the transmitting multilink bundle.

As fragments are received, the multilink bundle reassembles (resequences) the original large datagram from the fragments using the sequence number in the multilink header attached to the fragment by the sender. The reassembled large datagrams are then forwarded in normal fashion.

Multilink Bundles and Their Network Links

As mentioned earlier, a multilink bundle is a virtual PPP connection over a network link. The transmitting multilink bundle transmits the packet over a network link to a receiving multilink bundle, where the multilink bundle reassembles the fragments using the sequence number in the multilink header of the fragment.

The individual member links in a multilink bundle are standard serial PPP connections. Most forms of PPP connections may be used as member links in a bundle, including PPP over ATM, PPP over Frame Relay, and PPP over dial interfaces. However, there may be certain limitations and issues associated with using PPP sessions over certain media types, particularly those for “tunneling” protocols such as PPP over ATM, PPP over Frame Relay, and PPP over Ethernet.

The instructions for using Multilink over PPP on certain links are documented in separate modules (see the [“Where to Go Next”](#) section for more details). Follow the instructions for the type of link you are using.

Multiclass Multilink PPP

Multiclass Multilink PPP (MCMP) is based on RFC 2686: *Multi-Class Extension to Multi-Link PPP*. Multiclass Multilink PPP is an extension to the multilink functionality that adds the ability to divide network traffic over the multilink bundle into several independently sequenced streams of fragments. Multilink, as defined by RFC 1990: *The PPP Multilink Protocol (MP)*, provides for one sequenced stream only. RFC 1990 also implicitly allows one additional unsequenced stream, as large datagrams may be transmitted without multilink headers as long as the large datagrams do not need to be fragmented.

In Multiclass Multilink PPP, outgoing packets may be divided into as many as 16 different streams, for which RFC 2686 uses the term classes. Each stream or class has its own governing sequence number, and the receiving network peer (bundle) sorts and processes each stream independently.

Packets can still be sent without multilink headers. However, part of the purpose behind Multiclass Multilink PPP is to reduce or eliminate the need to send unsequenced data.

Multiclass Multilink PPP was created explicitly to allow the packets to be divided into several preemptable classes, so that any lower priority class could be interrupted in favor of sending a packet from a higher priority class. Each class of data can be fragmented, and all classes are expected to be fragmented (with the possible exception of the highest priority class). Also, frames from the different streams may be mixed if necessary.

Multiclass Multilink PPP was created as a mechanism to allow implementations to do interleaving, yet without giving up the sequencing of the interleaved packets such as occurs with standard interleaving.

In the Cisco IOS software, when Multilink Multiclass PPP is used instead of standard interleaving, the regular non-priority data is fragmented and transmitted in one class, and interleaved frames are sent in a separate class. Specifically, the regular traffic is sent in class 0 and the interleaved frames are sent in class 1. Thus, interleaving works just as it does with standard interleaving, except that the interleaved frames are sent in class 1 rather than as unsequenced frames. Multilink does not transmit data using additional classes, although Multilink is capable of receiving data from peers that do.

Multiclass Multilink PPP must be successfully negotiated with the peer system. If interleaving and Multiclass Multilink PPP are both configured, but the use of Multiclass Multilink PPP cannot be negotiated with the peer system, standard interleaving will be used.

For more information about Multiclass Multilink PPP, see the [Multiclass Multilink PPP](#) feature, Cisco IOS Release 12.2(13)T.

Distributed Multilink PPP

Distributed Multilink PPP (dMLP) is an implementation of Multilink on systems that support distributed processing. With distributed processing, packet processing can be handled by “dedicated hardware”—that is, either by the CPU or by another internal device such as a Versatile Interface Processor (VIP) inside the router or a FlexWAN inside the switch. This dedicated hardware can also be referred to as the “dMLP engine.”

One system that supports distributed processing is the Cisco 7500 series router with a Versatile Interface Processor (VIP2-40 or higher). Distributed processing is supported on a number of additional routers and switches as well. Refer to the documentation for your specific router or switch to see if it supports distributed processing.



Note On a Cisco 7500 series router, a VIP2-50 or higher 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.

With dMLP, packet fragmentation, interleaving, and fragment reassembly are done by the dMLP engine instead of by the Cisco IOS software. However, the Cisco IOS software manages the member links, creates and disassembles the bundles, and handles the control plane processing (including the handling of all PPP control packets).

However, once a bundle is established, the handling of Multilink packets is turned over to the dMLP engine. The dMLP engine handles all the multilink data-path functionality, including fragmentation, interleaving, multilink encapsulation, load balancing among the multiple links, and sorting and reassembly of inbound fragments.

The capabilities of the dMLP engines vary widely, and they may not always behave like the Cisco IOS Multilink feature. The dMLP engine may fragment and load balance using entirely different schemes than those used by the Cisco IOS software, and they may not support the same multilink features. For more information, refer to the documentation for the dMLP engine you are using or see the [Distributed Multilink Point-to-Point Protocol for Cisco 7500 Series Routers](#) feature, Cisco IOS Release 12.0(3)T.

Where to Go Next

To use Multilink PPP over Frame Relay, see the “” module.

To use Multilink PPP over ATM links, see the “” module.

To use Multilink PPP over dialer interface links, see the “” module.

To use Multilink PPP over serial interface links, see the “” module.

Additional References

The following sections provide additional references about Multilink.

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 , Release 12.3 T
LLQ, WFQ, CBWFQ, PQ, CQ, FIFO and other queueing mechanisms	“Congestion Management” section of the Cisco IOS Quality of Service Solutions Configuration Guide , Release 12.3
Frame Relay, ATM interfaces, ATM PVCs	Cisco IOS Wide-Area Networking Configuration Guide , Release 12.3
Multiclass Multilink PPP	Multiclass Multilink PPP feature, Cisco IOS Release 12.2(15)T
dMLP	Distributed Multilink Point-to-Point Protocol for Cisco 7500 Series Routers feature, Cisco IOS Release 12.0(3)T
Multilink PPP configuration information	Cisco IOS Dial Technologies Configuration Guide , Release 12.3
Multilink PPP over ATM links (including ATM interfaces and ATM PVCs)	“” module
Multilink PPP over Frame Relay	“” module
Multilink PPP over dialer interface links	“” module
Multilink PPP over serial interface links	“” 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 1990	<i>The PPP Multilink Protocol (MP)</i>
RFC 2686	<i>Multi-Class Extension to Multi-Link PPP</i>

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

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.

datagram—Logical grouping of information sent as a network layer unit over a transmission medium without prior establishment of a virtual circuit. IP datagrams are the primary information units in the Internet. The terms cell, frame, message, packet, and segment also are used to describe logical information groupings at various layers of the OSI reference model and in various technology circles.

jitter—1. The interpacket delay variance; that is, the difference between interpacket arrival and departure. Jitter is an important QoS metric for voice and video applications. 2. Analog communication line distortion caused by the variation of a signal from its reference timing positions. Jitter can cause data loss, particularly at high speeds.

latency—1. Delay between the time a device requests access to a network and the time it is granted permission to transmit. 2. Delay between the time a device receives a frame and the time that frame is forwarded out the destination port.

packet—Logical grouping of information that includes a header containing control information and (usually) user data. Packets most often are used to refer to network layer units of data. The terms datagram, frame, message, and segment also are used to describe logical information groupings at various layers of the OSI reference model and in various technology circles.

PVC— permanent virtual circuit (or connection). 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.

PPP—Point-to-Point Protocol. Successor to Serial Line Internet Protocol (SLIP) that provides router-to-router and host-to-network connections over synchronous and asynchronous circuits. Whereas SLIP was designed to work with IP, PPP was designed to work with several network layer protocols, such as IP, Internetwork Packet Exchange (IPX), and AppleTalk Remote Access (ARA). PPP also has built-in security mechanisms, such as Challenge Handshake Authentication Protocol (CHAP) and Password Authentication Protocol (PAP). PPP relies on two protocols: link control protocol (LCP) and Network Control Protocol (NCP).

VoIP—Voice over IP. The capability to carry normal telephony-style voice over an IP-based internet with plain old telephone service (POTS)-like functionality, reliability, and voice quality. VoIP enables a router to carry voice traffic (for example, telephone calls and faxes) over an IP network. In VoIP, the digital signal processor (DSP) segments the voice signal into frames, which then are coupled in groups of two and stored in voice packets. These voice packets are transported using IP in compliance with International Telecommunication Union Telecommunication Standardization Sector (ITU-T) specification H.323.

**Note**

Refer to [Internetworking Terms and Acronyms](#) for terms not included in this glossary.



Using Multilink PPP over ATM Links

Multilink PPP is a method used to reduce latency and jitter for real-time traffic. This module contains conceptual information and configuration tasks for using Multilink PPP over ATM links.

Module History

This module was first published on May 2, 2005, and last updated on May 2, 2005.

Your Cisco IOS software release may not support all features. To find information about feature support and configuration, use the [“Feature Information for Using Multilink PPP over ATM Links”](#) section on [page 673](#).

Contents

- [Prerequisites for Using Multilink PPP over ATM Links, page 659](#)
- [Restrictions for Using Multilink PPP over ATM Links, page 660](#)
- [Information About Using Multilink PPP over ATM Links, page 660](#)
- [How to Configure Multilink PPP over ATM Links, page 661](#)
- [Configuration Examples for Using Multilink PPP over ATM Links, page 669](#)
- [Where to Go Next, page 671](#)
- [Additional References, page 671](#)
- [Glossary, page 673](#)
- [Feature Information for Using Multilink PPP over ATM Links, page 673](#)

Prerequisites for Using Multilink PPP over ATM Links

Knowledge

- Be familiar with the concepts in the [“”](#) module.

Enable Queuing Mechanism

- Multilink PPP uses first-in first out (FIFO) queuing for queuing and interleaving packets. Other queuing mechanisms such as low latency queuing (LLQ), weighted fair queuing (WFQ), and class-based weighted fair queuing (CBWFQ) can be used. If you want to use one of these alternative mechanisms, enable it before configuring Multilink.

Restrictions for Using Multilink PPP over ATM Links

VoIP Support

- Only Voice over IP (VoIP) is supported; Voice over ATM is not supported.

ATM Network Modules Supported

- Multilink PPP over ATM must use the following ATM network modules:
 - Multiport T1/E1 ATM Network Module with Inverse Multiplexing over ATM
 - ATM OC-3 Network Module
 - Enhanced ATM Port Adapter

Information About Using Multilink PPP over ATM Links

To use Multilink PPP over ATM links, you should understand the following concepts:

- [MQC and Multilink PPP over ATM Links, page 660](#)
- [Multilink Group Interfaces, page 660](#)

MQC and Multilink PPP over ATM Links

Before using Multilink PPP over ATM links, a policy map must be created. (See the “[Prerequisites](#)” [section on page 663](#).) Policy maps are created using the Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC).

The MQC is a CLI structure that allows users to create traffic polices (policy maps) and attach these policy maps to interfaces. A policy map contains a traffic class and one or more QoS features. A traffic class is used to classify traffic. The QoS features in the traffic policy determine how to treat the classified traffic.

For more information about MQC, see the “Modular Quality of Service Command-Line Interface” section of the [Cisco IOS Quality of Service Solutions Configuration Guide](#), Release 12.3.

Virtual Template Interfaces

A virtual template interface is logical interface configured with generic configuration information for a specific purpose or configuration 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.

For more information virtual template interfaces, see the [Cisco IOS Dial Technologies Configuration Guide](#), Release 12.3.

Multilink Group Interfaces

A multilink group interface is a collection of interfaces bundled together in the multilink PPP configuration. With a multilink group interface, you can bundle interfaces into logical multilink groups.

For more information about multilink group interfaces and multilink groups, see the [Cisco IOS Dial Technologies Configuration Guide](#), Release 12.3.

How to Configure Multilink PPP over ATM Links

This section contains the procedures for configuring Multilink PPP over ATM links.



Note

While the first two procedures are listed as optional, you must choose one or the other according to the Cisco router you are using on your network.

- [Configuring Multilink PPP over ATM Links on a Virtual Template Interface, page 661](#) (optional; applies only if you are using the Cisco 7500 series router or the Cisco 7600 series router) or
- [Configuring Multilink PPP over ATM Links on a Multilink Group Interface, page 663](#) (optional)
- [Associating the Virtual Template Interface with an ATM PVC, page 666](#) (required)
- [Verifying the Multilink PPP over ATM Links Configuration, page 668](#) (optional)

Configuring Multilink PPP over ATM Links on a Virtual Template Interface

To configure Multilink PPP over ATM links on a virtual template interface, complete the following steps.



Note

These steps apply if you are using the Cisco 7500 series router or the Cisco 7600 series router only. If you are using another series of Cisco router, do not complete these steps. Instead, advance to [“Configuring Multilink PPP over ATM Links on a Multilink Group Interface” section on page 663](#).

Prerequisites

Before proceeding with this task, you must create a policy map. The policy map contains the configuration parameters used to apply a specific QoS features such as distributed LLQ (dLLQ) to the network traffic. To create a policy map and configure the appropriate QoS feature, use the MQC. See the [“MQC and Multilink PPP over ATM Links” section on page 660](#).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface virtual-template** *number*
4. **bandwidth** *kbps*
5. **ip address** *ip-address mask* [**secondary**]
6. **service-policy output** *policy-map-name*
7. **service-policy input** *policy-map-name*
8. **ppp multilink**

9. `ppp multilink fragment delay milliseconds [microseconds]`
10. `ppp multilink interleave`
11. `end`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p><code>enable</code></p> <p>Example: Router> enable</p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	<p><code>configure terminal</code></p> <p>Example: Router# configure terminal</p>	<p>Enters global configuration mode.</p>
Step 3	<p><code>interface virtual-template number</code></p> <p>Example: Router(config)# interface virtual-template 4</p>	<p>Creates a virtual template and enters interface configuration mode.</p> <ul style="list-style-type: none"> Enter the virtual template number.
Step 4	<p><code>bandwidth kbps</code></p> <p>Example: Router(config-if)# bandwidth 32</p>	<p>Sets the bandwidth value for an interface.</p> <ul style="list-style-type: none"> Enter the bandwidth value in kilobits per second.
Step 5	<p><code>ip address ip-address mask [secondary]</code></p> <p>Example: Router(config-if)# ip address 10.10.100.1 255.255.255.0</p>	<p>Sets a primary IP address for an interface. This command can also set the optional secondary IP address for an interface.</p> <ul style="list-style-type: none"> Enter the primary IP address (and, optionally, the secondary IP address).
Step 6	<p><code>service-policy output policy-map-name</code></p> <p>Example: Router(config-if)# service-policy output policy1</p>	<p>Attaches the previously created QoS traffic policy (policy map). See the “Prerequisites” section on page 663. The policy map evaluates and applies QoS features for traffic <i>leaving</i> the interface.</p> <ul style="list-style-type: none"> Enter the policy map name.
Step 7	<p><code>service-policy input policy-map-name</code></p> <p>Example: Router(config-if)# service-policy input policy1</p>	<p>Attaches the previously created QoS traffic policy (policy map). See the “Prerequisites” section on page 663. The policy map evaluates and applies QoS features for traffic <i>entering</i> the interface.</p> <ul style="list-style-type: none"> Enter the policy map name.
Step 8	<p><code>ppp multilink</code></p> <p>Example: Router(config-if)# ppp multilink</p>	<p>Enables MLP on the interface.</p>

	Command or Action	Purpose
Step 9	<pre>ppp multilink fragment delay milliseconds [microseconds]</pre> <p>Example: Router(config-if)# ppp multilink fragment delay 20</p>	<p>Specifies a maximum size in units of time for packet fragments on a Multilink PPP (MLP) bundle.</p> <ul style="list-style-type: none"> Enter the maximum amount of time, in milliseconds.
Step 10	<pre>ppp multilink interleave</pre> <p>Example: Router(config-if)# ppp multilink interleave</p>	<p>Enables interleaving of packets among the fragments of larger packets on a multilink bundle.</p>
Step 11	<pre>end</pre> <p>Example: Router(config-if)# end</p>	<p>(Optional) Exits interface configuration mode.</p>

Fragment size at the MLP bundle can be configured using the following formula:

$$\text{fragment size} = \text{bandwidth} \times \text{fragment-delay} / 8$$

The ideal fragment size for should allow the fragments to fit into an exact multiple of ATM cells. The fragment size can be calculated using the following formula:

$$\text{fragment size} = 48 \times \text{number of cells} - 10$$

Configuring Multilink PPP over ATM Links on a Multilink Group Interface

To configure Multilink PPP over ATM links on a multilink group interface, complete the following steps.



Note

If you are using the Cisco 7500 series router or the Cisco 7600 series router, do not complete these steps. Instead, complete the steps in “[Configuring Multilink PPP over ATM Links on a Virtual Template Interface](#)” section on page 661.

Prerequisites

Before proceeding with this task, you must create a policy map. The policy map contains the configuration parameters used to apply a specific QoS features such as distributed LLQ (dLLQ) to the network traffic. To create a policy map and configure the appropriate QoS feature, use the MQC. See the “[MQC and Multilink PPP over ATM Links](#)” section on page 660.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface multilink** *multilink-bundle-number*
4. **ip address** *ip-address mask* [**secondary**]
5. **service-policy output** *policy-map-name*

6. **service-policy input** *policy-map-name*
7. **ppp multilink fragment delay** *milliseconds* [*microseconds*]
8. **ppp multilink interleave**
9. **ppp multilink multiclass**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface multilink <i>multilink-bundle-number</i> Example: Router(config)# interface multilink 1	Creates a multilink bundle and enters interface configuration mode. <ul style="list-style-type: none"> Enter the multilink bundle number.
Step 4	ip address <i>ip-address mask</i> [secondary] Example: Router(config-if)# ip address 10.10.100.1 255.255.255.0	Sets a primary IP address for an interface. This command can also set the optional secondary IP address for an interface. <ul style="list-style-type: none"> Enter the primary IP address (and, optionally, the secondary IP address).
Step 5	service-policy output <i>policy-map-name</i> Example: Router(config-if)# service-policy output policy1	Attaches the previously created QoS traffic policy (policy map). See the “Prerequisites” section on page 663. The policy map evaluates and applies QoS features for traffic <i>leaving</i> the interface. <ul style="list-style-type: none"> Enter the policy map name.
Step 6	service-policy input <i>policy-map-name</i> Example: Router(config-if)# service-policy input policy1	Attaches the previously created QoS traffic policy (policy map). See the “Prerequisites” section on page 663. The policy map evaluates and applies QoS features for traffic <i>entering</i> the interface. <ul style="list-style-type: none"> Enter the policy map name.
Step 7	ppp multilink fragment delay <i>milliseconds</i> [<i>microseconds</i>] Example: Router(config-if)# ppp multilink fragment delay 20	Specifies a maximum size in units of time for packet fragments on a Multilink PPP (MLP) bundle. <ul style="list-style-type: none"> Enter the maximum amount of time, in milliseconds.
Step 8	ppp multilink interleave Example: Router(config-if)# ppp multilink interleave	Enables interleaving of packets among the fragments of larger packets on a multilink bundle.

	Command or Action	Purpose
Step 9	<code>ppp multilink multiclass</code> Example: Router(config-if)# ppp multilink multiclass	(Optional) Enables Multiclass Multilink PPP (MCMP) on an interface. Note Use this command only if there are multiple links in the multilink bundle.
Step 10	<code>end</code> Example: Router(config-if)# end	(Optional) Exits interface configuration mode.

What to Do Next

After configuring Multilink PPP over ATM links on a multilink group interface, the next step is to associate the virtual template interface with the multilink group by completing the steps in the following section.

If you are using a Cisco 7500 series router or a Cisco 7600 series router, advance to [“Associating the Virtual Template Interface with an ATM PVC”](#) section on page 666 to continue.

Associating the Virtual Template Interface with the Multilink Group

To associate the virtual template interface with the multilink group, complete the following steps.

SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `interface virtual-template number`
4. `no ip address`
5. `ppp multilink group group-number`
6. `end`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>enable</code> Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	<code>configure terminal</code> Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	interface virtual-template <i>number</i> Example: Router# interface virtual-template 2	Creates a virtual template interface that can be configured and applied dynamically in creating virtual access interfaces, and enters interface configuration mode. <ul style="list-style-type: none"> Enter the number used to identify the virtual template interface.
Step 4	no ip address Example: Router(config-if)# no ip address	Removes an IP address or disables IP processing.
Step 5	ppp multilink group <i>group-number</i> Example: Router(config-if)# ppp multilink group 1	Restricts a physical link to joining only a designated multilink group interface. <ul style="list-style-type: none"> Enter the multilink group number.
Step 6	end Example: Router(config-if)# end	(Optional) Exits interface configuration mode.

Associating the Virtual Template Interface with an ATM PVC

To associate the virtual template interface with an ATM PVC, complete the following steps.

SUMMARY STEPS

- enable**
- configure terminal**
- interface** *type number* [**name-tag**]
- pvc** [*name*] *vpi/vci* [**ces** | **ilmi** | **qsaal** | **smds** | **I2transport**]
- abr** *output-pcr output-mcr*
- vbr-nrt** *output-pcr output-scr output-mbs* [*input-pcr*] [*input-scr*] [*input-mbs*]
- protocol ppp virtual-template** *number*
- end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> [name-tag] Example: Router(config)# interface atm2/0/0	Configures an interface type and enters interface configuration mode. <ul style="list-style-type: none">Enter the interface type and number.
Step 4	pvc [<i>name</i>] <i>vpi/vci</i> [ces ilmi qsaal smds l2transport] Example: Router(config-if)# pvc cisco 0/16	Creates or assigns a name to an ATM permanent virtual circuit (PVC), and enters ATM virtual circuit configuration mode. <ul style="list-style-type: none">Enter the ATM PVC name, the network virtual path identifier, and the network virtual channel identifier.
Step 5	abr <i>output-pcr output-mcr</i> Example: Router(config-if-atm-vc)# abr 100 80	(Optional) Selects available bit rate (ABR) QoS and configures the output peak cell rate (PCR) and output minimum guaranteed cell rate (MCR) for an ATM PVC. <ul style="list-style-type: none">Enter the output PCR and the output MCR.
Step 6	vbr-nrt <i>output-pcr output-scr output-mbs</i> [<i>input-pcr</i>] [<i>input-scr</i>] [<i>input-mbs</i>] Example: Router(config-if-atm-vc)# abr 1100 1100 1100	(Optional) Configures the variable bit rate-nonreal time (VBR-NRT) quality of service (QoS) and specifies the output peak cell rate (PCR), output sustainable cell rate (SCR), and output maximum burst cell size (MBS) for an ATM PVC, PVC range, switched virtual circuit (SVC), VC class, or VC bundle member. <ul style="list-style-type: none">Enter the PCR, SCR, and MBS.
Step 7	protocol ppp virtual-template <i>number</i> Example: Router(config-if-atm-vc)# protocol ppp virtual-template 2	Specifies that PPP is established over the ATM PVC using the configuration from the specified virtual template. <ul style="list-style-type: none">Enter the virtual-template number.
Step 8	end Example: Router(config-if-atm-vc)# end	(Optional) Exits ATM virtual circuit configuration mode.

Verifying the Multilink PPP over ATM Links Configuration

To verify the multilink PPP over ATM links configuration, complete the following steps.

SUMMARY STEPS

1. **enable**
2. **show atm pvc** [*vpi/vci* | *name* | **interface atm** *interface-number* [*.subinterface-number* **multipoint**]] [**ppp**]
3. **show interfaces** [*type number*] [*first*] [*last*] [**accounting**]
4. **show ppp multilink** [**active** | **inactive** | **interface** *bundle-interface* | [**username** *name*] [**endpoint** *endpoint*]]
5. **show policy-map interface** *interface-name* [**vc** [*vpi/* *vci*] [**dlei** *dlei*] [**input** | **output**]
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	show atm pvc [<i>vpi/vci</i> <i>name</i> interface atm <i>interface-number</i> [<i>.subinterface-number</i> multipoint]] [ppp] Example: Router# show atm pvc	(Optional) Displays all ATM permanent virtual connections (PVCs) and traffic information.
Step 3	show interfaces [<i>type number</i>] [<i>first</i>] [<i>last</i>] [accounting] Example: Router# show interfaces	(Optional) Displays statistics for all interfaces configured on the router or access server.
Step 4	show ppp multilink [active inactive interface <i>bundle-interface</i> [username <i>name</i>] [endpoint <i>endpoint</i>]] Example: Router# show ppp multilink	(Optional) Displays bundle information for multilink bundles.

	Command or Action	Purpose
Step 5	<pre>show policy-map interface interface-name [vc [vpi/] vci] [dlci dlci] [input output]</pre> <p>Example: Router# show policy-map interface serial0/0</p>	(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 permanent virtual circuit (PVC) on the interface.
Step 6	<pre>exit</pre> <p>Example: Router# exit</p>	(Optional) Exits privileged EXEC mode.

Configuration Examples for Using Multilink PPP over ATM Links

This section contains the following examples:

- [Configuring Multilink PPP over ATM Links on a Virtual Template Interface: Example, page 669](#)
- [Configuring Multilink PPP over ATM Links on a Multilink Group Interface: Example, page 670](#)
- [Associating the Virtual Template Interface with the Multilink Group: Example, page 670](#)
- [Associating the Virtual Template Interface with an ATM PVC: Example, page 670](#)
- [Verifying the Multilink PPP over ATM Links Configuration: Example, page 670](#)

Configuring Multilink PPP over ATM Links on a Virtual Template Interface: Example

The following is an example of configuring Multilink PPP over ATM links on a virtual template interface:

```
Router> enable
Router# configure terminal
Router(config)# interface virtual-template 4
Router(config-if)# bandwidth 32
Router(config-if)# ip address 10.10.100.1 255.255.255.0
Router(config-if)# service-policy output policy1
Router(config-if)# service-policy input policy1
Router(config-if)# ppp multilink
Router(config-if)# ppp multilink fragment delay 20
Router(config-if)# ppp multilink interleave
Router(config-if)# end
```

Configuring Multilink PPP over ATM Links on a Multilink Group Interface: Example

The following is an example of configuring Multilink PPP over ATM links on a multilink group interface:

```
Router> enable
Router# configure terminal
Router(config)# interface multilink 1
Router(config-if)# ip address 10.10.100.1 255.255.255.0
Router(config-if)# service-policy output policy1
Router(config-if)# service-policy input policy1
Router(config-if)# ppp multilink fragment delay 20
Router(config-if)# ppp multilink interleave
Router(config-if)# ppp multilink multiclass
Router(config-if)# end
```

Associating the Virtual Template Interface with the Multilink Group: Example

The following is an example of associating the virtual template interface with the multilink group:

```
Router> enable
Router# configure terminal
Router(config)# interface virtual-template 2
Router(config-if)# no ip address
Router(config-if)# ppp multilink group 1
Router(config-if)# end
```

Associating the Virtual Template Interface with an ATM PVC: Example

The following is an example of associating the virtual template interface with an ATM PVC:

```
Router> enable
Router# configure terminal
Router(config)# interface atm2/0/0
Router(config-if)# pvc cisco 0/16
Router(config-if-atm-vc)# abr 100 80
Router(config-if-atm-vc)# protocol ppp virtual-template 2
Router(config-if-atm-vc)# end
```

Verifying the Multilink PPP over ATM Links Configuration: Example

You can verify the Multilink PPP over ATM links configuration by using one or more of the following **show** commands:

- **show atm pvc**
- **show interfaces**
- **show ppp multilink**
- **show policy-map interface**

The following section provides sample output of the **show ppp multilink** command only. For sample output of the other commands, see the appropriate Cisco IOS Release 12.3 T command reference publication.

show ppp multilink Command Output Example

The following is an example of the **show ppp multilink** command output. In this example, one multilink bundle called 7206-2 is on the system. This bundle has two member links: one active link and one inactive link.

```
Router# show ppp multilink

Multilink1, bundle name is 7206-2
Endpoint discriminator is 7206-2
Bundle up for 00:00:24, 1/255 load
Receive buffer limit 12000 bytes, frag timeout 1000 ms
 0/0 fragments/bytes in reassembly list
 0 lost fragments, 0 reordered
 0/0 discarded fragments/bytes, 0 lost received
 0x0 received sequence, 0x0 sent sequence
Member links: 1 active, 1 inactive (max not set, min not set)
Vi3, since 00:00:24
  PPPoATM link, ATM PVC 2/101 on ATM2/0/0
  Packets in ATM PVC Holdq: 0 , Particles in ATM PVC Tx Ring: 1
  Vt1 (inactive)
```

Where to Go Next

To use Multilink PPP over Frame Relay, see the “” module.

To use Multilink PPP over dialer interface links, see the “” module.

To use Multilink PPP over serial interface links, see the “” module.

Additional References

The following sections provide references related to using Multilink PPP over ATM links.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	<i>Cisco IOS Quality of Service Solutions Command Reference</i> , Release 12.3 T
LLQ, dLLQ, WFQ, CBWFQ and other queuing mechanisms	“Congestion Management” section of the <i>Cisco IOS Quality of Service Solutions Configuration Guide</i> , Release 12.3
MQC	“Modular Quality of Service Command-Line Interface” section of the <i>Cisco IOS Quality of Service Solutions Configuration Guide</i> , Release 12.3
Multilink PPP configurations and virtual template interfaces	<i>Cisco IOS Dial Technologies Configuration Guide</i> , Release 12.3

Related Topic	Document Title
Multilink PPP overview module	“” module
Multilink PPP over Frame Relay	“” module
Multilink PPP over dialer interface links	“” module
Multilink PPP over serial interface links	“” 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 1990	<i>The PPP Multilink Protocol (MP)</i>
RFC 2686	<i>Multiclass Extension to Multilink PPP (MCML)</i>

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

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.

PVC—permanent virtual circuit (or connection). 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.

virtual template interface—A logical interface configured with generic configuration information for a specific purpose or configuration 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.



Note

See [Internetworking Terms and Acronyms](#) for terms not included in this glossary.

Feature Information for Using Multilink PPP over ATM Links

[Table 6](#) 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.

For information on a feature in this technology that is not documented here, see the “[Reducing Latency and Jitter Using Multilink PPP Roadmap](#).”

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 Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.



Note

[Table 6](#) 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 6 Feature Information for Using Multilink PPP over ATM Links

Feature Name	Software Releases	Feature Configuration Information
Distributed Link Fragmentation and Interleaving Over Leased Lines	12.2(8)T	<p>The Distributed Link Fragmentation and Interleaving over Leased Lines feature extends distributed link fragmentation and interleaving functionality to leased lines.</p> <p>This feature was extensively rewritten from the perspective of using Multilink PPP for link fragmentation and interleaving over ATM interface links.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Information About Using Multilink PPP over ATM Links, page 660 • How to Configure Multilink PPP over ATM Links, page 661
Distributed Link Fragmentation and Interleaving for Frame Relay and ATM Interfaces on Cisco 7500 Series Routers	12.2(4)T	<p>The Distributed Link Fragmentation and Interleaving (dLFI) for Frame Relay and ATM Interfaces on Cisco 7500 Series Routers feature extends link fragmentation and interleaving functionality to VIP-enabled Cisco 7500 series routers.</p> <p>This feature was extensively rewritten from the perspective of using Multilink PPP for link fragmentation and interleaving over ATM interface links.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Information About Using Multilink PPP over ATM Links, page 660 • How to Configure Multilink PPP over ATM Links, page 661



Using Multilink PPP over Dialer Interface Links

Multilink PPP is a method used to reduce latency and jitter for real-time traffic. This module contains conceptual information and configuration tasks for using Multilink PPP over dialer interface links.

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 features. To find information about feature support and configuration, use the [“Feature Information for Using Multilink PPP over Dialer Interface Links”](#) section on page 686.

Contents

- [Prerequisites for Using Multilink PPP over Dialer Interface Links, page 675](#)
- [Restrictions for Using Multilink PPP over Dialer Interface Links, page 676](#)
- [Information About Using Multilink PPP over Dialer Interface Links, page 676](#)
- [How to Configure Multilink PPP over Dialer Interface Links, page 677](#)
- [Configuration Examples for Using Multilink PPP over Dialer Interface Links, page 682](#)
- [Where to Go Next, page 683](#)
- [Additional References, page 684](#)
- [Glossary, page 685](#)
- [Feature Information for Using Multilink PPP over Dialer Interface Links, page 686](#)

Prerequisites for Using Multilink PPP over Dialer Interface Links

Knowledge

- Be familiar with the concepts in the “” module.

Enable Queuing Mechanism

- Multilink uses first-in first out (FIFO) queuing for queuing and interleaving packets. Other queuing mechanisms such as low latency queuing (LLQ), weighted fair queuing (WFQ), and class-based weighted fair queuing (CBWFQ) can be used. If you want to use one of these alternative mechanisms, enable it before configuring Multilink.

Restrictions for Using Multilink PPP over Dialer Interface Links

RSP Support

- Route/switch processing (RSP) is not recommended when using Multilink PPP over dialer interface links.

Information About Using Multilink PPP over Dialer Interface Links

To use Multilink PPP over dialer interface links, you should understand the following concepts:

- [Dialer Profiles, page 676](#)
- [MQC and Multilink PPP over Dialer Interface Links, page 677](#)

Dialer Profiles

The dialer profiles implementation of dial-on-demand routing (DDR) is based on a separation between logical and physical interface configuration. Dialer profiles also allow the logical and physical configurations to be bound together dynamically on a per-call basis.

Dialer profiles are advantageous in the following situations:

- When you want to share an interface (ISDN, asynchronous, or synchronous serial) to place or receive calls.
- When you want to change any configuration on a per-user basis.
- When you want to maximize ISDN channel usage using the Dynamic Multiple Encapsulations feature to configure various encapsulation types and per-user configurations on the same ISDN B channel at different times according to the type of call.
- When you want to bridge to many destinations, and for avoiding split horizon problems.

Most routed protocols are supported; however, International Organization for Standardization Connectionless Network Service (ISO CLNS) is not supported.

If you decide to configure dialer profiles, you must disable validation of source addresses for the routed protocols you support.

For more information about dialer profiles, see the *Cisco IOS Dial Technologies Configuration Guide*, Release 12.3, or see the [Configuring and Troubleshooting Dialer Profiles](#) document on Cisco.com.

MQC and Multilink PPP over Dialer Interface Links

Before using Multilink PPP over dialer interface links, a traffic policy (also known as a policy map) must be created. (See the [“Prerequisites” section on page 677](#).) Policy maps are created using the Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC).

The MQC is a CLI structure that allows users to create traffic policies (policy maps) and attach these policy maps to interfaces. A policy map contains a traffic class and one or more QoS features. A traffic class is used to classify traffic. The QoS features in the traffic policy determine how to treat the classified traffic.

For more information about MQC, see the “Modular Quality of Service Command-Line Interface” section of the [Cisco IOS Quality of Service Solutions Configuration Guide](#), Release 12.3.

How to Configure Multilink PPP over Dialer Interface Links

This section contains the following procedures:

- [Configuring Multilink PPP over Dialer Interface Links, page 677](#) (required)
- [Associating the Dialer Interface with a BRI, page 680](#) (required)
- [Verifying the Multilink PPP over Dialer Interface Link Configuration, page 681](#) (optional)

Configuring Multilink PPP over Dialer Interface Links

To configure Multilink PPP over dialer interface links, complete the following steps.

Prerequisites

Before proceeding with this task, you must create a policy map. The policy map contains the configuration parameters used to apply the specific quality of service feature to the network traffic. To create a policy map, use the MQC. See the [“MQC and Multilink PPP over Dialer Interface Links” section on page 677](#).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface dialer** *dialer-rotary-group-number*
4. **ip address** *ip-address mask* [**secondary**]
5. **ip unnumbered** *type number*
6. **encapsulation** *encapsulation-type*
7. **dialer pool** *number*
8. **dialer in-band** [**no-parity** | **odd-parity**]
9. **service-policy output** *policy-map-name*
10. **service-policy input** *policy-map-name*

11. **ppp authentication** {*protocol1* [*protocol2...*]} [**if-needed**] [*list-name* | **default**] [**callin**] [**one-time**] [**optional**]
12. **ppp chap hostname** *hostname*
13. **ppp chap password** *secret*
14. **ppp multilink** [**bap**]
15. **ppp multilink fragment delay** *milliseconds* [*microseconds*]
16. **ppp multilink interleave**
17. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface dialer <i>dialer-rotary-group-number</i> Example: Router(config)# interface dialer 1	Defines a dialer rotary group and enters interface configuration mode. <ul style="list-style-type: none">Enter the dialer rotary group number.
Step 4	ip address <i>ip-address mask</i> [secondary] Example: Router(config-if)# ip address 10.10.100.1 255.255.255.0	Sets a primary IP address for an interface. This command can also set the optional secondary IP address for an interface. <ul style="list-style-type: none">Enter the primary IP address (and, optionally, the secondary IP address).
Step 5	ip unnumbered <i>type number</i> Example: Router(config-if)# ip unnumbered ethernet 0	(Optional) Enables IP processing on a serial interface without assigning an explicit IP address to the interface. <ul style="list-style-type: none">Enter the type and number of another interface on which the router has an assigned IP address. It cannot be another unnumbered interface.
Step 6	encapsulation <i>encapsulation-type</i> Example: Router(config-if)# encapsulation ppp	Sets the encapsulation method used by the interface. <ul style="list-style-type: none">Enter the encapsulation method. For this feature, enter PPP.
Step 7	dialer pool <i>number</i> Example: Router(config-if)# dialer pool 3	(Optional) Specifies which dialing pool to use to connect to a specific destination subnetwork. <ul style="list-style-type: none">Enter the dialing pool number.

	Command or Action	Purpose
Step 8	dialer in-band [no-parity odd-parity] Example: Router(config-if)# dialer in-band	(Optional) Specifies that dial-on-demand routing (DDR) is to be supported.
Step 9	service-policy output <i>policy-map-name</i> Example: Router(config-if)# service-policy output policy1	Attaches the previously created QoS traffic policy (policy map). See the “Prerequisites” section on page 677. The policy map evaluates and applies QoS features for traffic <i>leaving</i> the interface. <ul style="list-style-type: none"> Enter the policy map name.
Step 10	service-policy input <i>policy-map-name</i> Example: Router(config-if)# service-policy input policy1	Attaches the previously created QoS traffic policy (policy map). See the “Prerequisites” section on page 677. The policy map evaluates and applies QoS features for traffic <i>entering</i> the interface. <ul style="list-style-type: none"> Enter the policy map name.
Step 11	ppp authentication { <i>protocol1</i> [<i>protocol2...</i>]} [if-needed] [<i>list-name</i> default] [callin] [one-time] [optional] Example: Router(config-if)# ppp authentication chap	Enables at least one Point-to-Point Protocol (PPP) authentication protocol and specifies the order in which the protocols are selected on the interface. <ul style="list-style-type: none"> Enter the PPP authentication protocol to be used.
Step 12	ppp chap hostname <i>hostname</i> Example: Router(config-if)# ppp chap hostname ISPCorp	Creates a pool of dialup routers that all appear to be the same host when authenticating with Challenge Handshake Authentication Protocol (CHAP). <ul style="list-style-type: none"> Enter the name sent in the CHAP challenge.
Step 13	ppp chap password <i>secret</i> Example: Router(config-if)# ppp chap password 7	Enables a router calling a collection of routers that do not support this command (such as routers running older Cisco IOS software images) to configure a CHAP secret password to use in response to challenges from an unknown peer. <ul style="list-style-type: none"> Enter the secret password used to compute the response value for any CHAP challenge from an unknown peer.
Step 14	ppp multilink [bap] Example: Router(config-if)# ppp multilink	Enables multilink on an interface.
Step 15	ppp multilink fragment delay <i>milliseconds</i> [<i>microseconds</i>] Example: Router(config-if)# ppp multilink fragment delay 20	Specifies a maximum size in units of time for packet fragments on a Multilink PPP (MLP) bundle. <ul style="list-style-type: none"> Enter the maximum amount of time, in milliseconds.

	Command or Action	Purpose
Step 16	ppp multilink interleave Example: Router(config-if)# ppp multilink interleave	Enables interleaving of packets among the fragments of larger packets on a multilink bundle.
Step 17	end Example: Router(config-if)# end	(Optional) Exits interface configuration mode.

Associating the Dialer Interface with a BRI

The Basic Rate Interface (BRI) is used as the backup for the dialer interface. The BRI interface has multilink capability, and all the other characteristics of the dialer interface.

To associate the dialer interface with a BRI, complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface bri** *number*
4. **dialer pool-member** *number* [**priority** *priority*] [**min-link** *minimum*] [**max-link** *maximum*]
5. **dialer rotary-group** *interface-number*
6. **ppp multilink** [**bap**]
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface bri <i>number</i> Example: Router(config)# interface bri 1	Configures a BRI interface and enters interface configuration mode. <ul style="list-style-type: none"> • Enter port, connector, or interface card number.

	Command or Action	Purpose
Step 4	dialer pool-member <i>number</i> [priority <i>priority</i>] [min-link <i>minimum</i>] [max-link <i>maximum</i>] Example: Router(config-if)# dialer pool-member 3	(Optional) Configures a physical interface to be a member of a dialer profile dialing pool. <ul style="list-style-type: none"> Enter the dialer profile dialing pool number.
Step 5	dialer rotary-group <i>interface-number</i> Example: Router(config-if)# dialer rotary-group 1	(Optional) Includes a specified interface in a dialer rotary group. <ul style="list-style-type: none"> Enter the number of the dialer interface (defined in Step 4) in whose rotary group this interface is to be included.
Step 6	ppp multilink [ba] Example: Router(config-if)# ppp multilink	Enables Multilink on an interface.
Step 7	end Example: Router(config-if)# end	(Optional) Exits interface configuration mode.

Verifying the Multilink PPP over Dialer Interface Link Configuration

To verify the Multilink PPP over dialer interface link configuration, complete the following steps.

SUMMARY STEPS

- enable**
- show interfaces** [*type number*] [*first*] [*last*] [**accounting**]
- show ppp multilink** [**active** | **inactive** | **interface** *bundle-interface* | [**username** *name*] [**endpoint** *endpoint*]]
- exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	show interfaces [<i>type number</i>] [<i>first</i>] [<i>last</i>] [accounting] Example: Router# show interfaces	(Optional) Displays statistics for all interfaces configured on the router or access server.

	Command or Action	Purpose
Step 3	<pre>show ppp multilink [active inactive interface bundle-interface [username name] [endpoint endpoint]]</pre> <p>Example: Router# show ppp multilink</p>	(Optional) Displays bundle information for the multilink bundles.
Step 4	<pre>exit</pre> <p>Example: Router# exit</p>	(Optional) Exits privileged EXEC mode.

Configuration Examples for Using Multilink PPP over Dialer Interface Links

This section contains the following examples:

- [Configuring Multilink PPP over Dialer Interface Links: Example, page 682](#)
- [Associating the Dialer Interface with a BRI: Example, page 683](#)
- [Verifying the Multilink PPP over Dialer Interface Link Configuration: Example, page 683](#)

Configuring Multilink PPP over Dialer Interface Links: Example

The following is an example of configuring Multilink PPP over a dialer interface link:

```
Router> enable
Router# configure terminal
Router(config)# interface dialer 1
Router(config-if)# ip address 10.10.100.1 255.255.255.0
Router(config-if)# encapsulation ppp
Router(config-if)# dialer pool 3
Router(config-if)# service-policy output policy1
Router(config-if)# service-policy input policy1
Router(config-if)# ppp authentication chap
Router(config-if)# ppp chap hostname ISPCorp
Router(config-if)# ppp chap password 7
Router(config-if)# ppp multilink
Router(config-if)# ppp multilink fragment delay 20
Router(config-if)# ppp multilink interleave
Router(config-if)# end
```


Associating the Dialer Interface with a BRI: Example

The following is an example of associating the dialer interface with a BRI:

```
Router> enable
Router# configure terminal
Router(config)# interface bri 1
Router(config-if)# dialer pool-member 3
Router(config-if)# ppp multilink
Router(config-if)# end
```

Verifying the Multilink PPP over Dialer Interface Link Configuration: Example

You can verify the Multilink PPP over dialer interface link configuration by using one or more of the following **show** commands:

- **show interfaces**
- **show ppp multilink**

The following section provides sample output of the **show ppp multilink** command only. For sample output of the other commands, see the appropriate Cisco IOS Release 12.3 T command reference publication.

show ppp multilink Command Output Example

The following is an example of the **show ppp multilink** command output. In this example, one multilink bundle called 7206-2 is on the system. This bundle has one member link.

```
Router# show ppp multilink

Dialer2, bundle name is 7206-2
  Username is 7206-2
  Endpoint discriminator is 7206-2
  Bundle up for 00:00:10, 1/255 load
  Receive buffer limit 12000 bytes, frag timeout 1500 ms
    0/0 fragments/bytes in reassembly list
    0 lost fragments, 0 reordered
    0/0 discarded fragments/bytes, 0 lost received
    0x0 received sequence, 0x0 sent sequence
  Member links:1 (max not set, min not set)
    BR2/0:1, since 00:00:09
```

Where to Go Next

To use Multilink PPP over ATM links, see the “” module.

To use Multilink PPP over Frame Relay, see the “” module.

To use Multilink PPP over serial interface links, see the “” module.

Additional References

The following sections provide references related to Multilink PPP over dialer interface links.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	<i>Cisco IOS Quality of Service Solutions Command Reference</i> , Release 12.3 T
LLQ, dLLQ, WFQ, CBWFQ and other queuing mechanisms	“Congestion Management” section of the <i>Cisco IOS Quality of Service Solutions Configuration Guide</i> , Release 12.3
MQC	“Modular Quality of Service Command-Line Interface” section of the <i>Cisco IOS Quality of Service Solutions Configuration Guide</i> , Release 12.3
Dialer profiles and DDR	<i>Cisco IOS Dial Technologies Configuration Guide</i> , Release 12.3, <i>Configuring and Troubleshooting Dialer Profiles</i> .
Multilink PPP configuration information	<i>Cisco IOS Dial Technologies Configuration Guide</i> , Release 12.3
Multilink PPP overview module	“” module
Multilink PPP over Frame Relay	“” module
Multilink PPP over ATM links (including ATM interfaces and ATM PVCs)	“” module
Multilink PPP over serial interface links	“” 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 1990	<i>The PPP Multilink Protocol (MP)</i>
RFC 2686	<i>Multiclass Extension to Multilink PPP (MCML)</i>

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

BRI—Basic Rate Interface. ISDN interface composed of two B channels and one D channel for circuit-switched communication of voice, video, and data.

CHAP—Challenge Handshake Authentication Protocol. Security feature supported on lines using Point-to-Point Protocol (PPP) encapsulation that prevents unauthorized access. CHAP does not itself prevent unauthorized access, but merely identifies the remote end. The router or access server then determines whether that user is allowed access.

DDR—dial-on-demand routing. Technique whereby a router can automatically initiate and close a circuit-switched session as transmitting stations demand. The router spoofs keepalives so that end stations treat the session as active. DDR permits routing over ISDN or telephone lines using an external ISDN terminal adaptor or modem.

ISDN—Integrated Services Digital Network. Communication protocol offered by telephone companies that permits telephone networks to carry data, voice, and other source traffic.

PRI—Primary Rate Interface. ISDN interface to primary rate access. Primary rate access consists of a single 64-kbps D channel plus 23 (T1) or 30 (E1) B channels for voice or data.

RSP—Route/Switch Processor. Processor module in the Cisco 7500 series routers that integrates the functions of the route processor (RP) and the switch processor (SP).



Note

See *Internetworking Terms and Acronyms* for terms not included in this glossary.

Feature Information for Using Multilink PPP over Dialer Interface Links

[Table 7](#) 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.

For information on a feature in this technology that is not documented here, see the “[Reducing Latency and Jitter Using Multilink PPP Roadmap](#).”

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 Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

**Note**

[Table 7](#) 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 7 **Feature Information for Using Multilink PPP over Dialer Interface Links**

Feature Name	Software Releases	Feature Configuration Information
Distributed Link Fragmentation and Interleaving Over Leased Lines	12.2(8)T	<p>The Distributed Link Fragmentation and Interleaving over Leased Lines feature extends distributed link fragmentation and interleaving functionality to leased lines.</p> <p>This feature was extensively rewritten from the perspective of using Multilink PPP for link fragmentation and interleaving over dialer interface links.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Information About Using Multilink PPP over Dialer Interface Links, page 676 • How to Configure Multilink PPP over Dialer Interface Links, page 677
Distributed Link Fragmentation and Interleaving for Frame Relay and ATM Interfaces on Cisco 7500 Series Routers	12.2(4)T	<p>The Distributed Link Fragmentation and Interleaving (dLFI) for Frame Relay and ATM Interfaces on Cisco 7500 Series Routers feature extends link fragmentation and interleaving functionality to VIP-enabled Cisco 7500 series routers.</p> <p>This feature was extensively rewritten from the perspective of using Multilink PPP for link fragmentation and interleaving over dialer interface links.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Information About Using Multilink PPP over Dialer Interface Links, page 676 • How to Configure Multilink PPP over Dialer Interface Links, page 677



Using Multilink PPP over Frame Relay

Multilink PPP is a method used to reduce latency and jitter for real-time traffic. This module contains conceptual information and configuration tasks for using Multilink PPP over Frame Relay.

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 features. To find information about feature support and configuration, use the [“Feature Information for Using Multilink PPP over Frame Relay”](#) section on [page 704](#).

Contents

- [Prerequisites for Using Multilink PPP over Frame Relay, page 690](#)
- [Restrictions for Using Multilink PPP over Frame Relay, page 690](#)
- [Information About Using Multilink PPP over Frame Relay, page 690](#)
- [How to Configure Multilink PPP over Frame Relay, page 691](#)
- [Configuration Examples for Multilink PPP over Frame Relay, page 699](#)
- [Where to Go Next, page 701](#)
- [Additional References, page 702](#)
- [Glossary, page 703](#)
- [Feature Information for Using Multilink PPP over Frame Relay, page 704](#)

Prerequisites for Using Multilink PPP over Frame Relay

Knowledge

- Be familiar with the concepts in the “” module.

Enable Queuing Mechanism

- Multilink uses first-in first out (FIFO) queuing for queuing and interleaving packets. Other queuing mechanisms such as low latency queuing (LLQ), weighted fair queuing (WFQ), and class-based weighted fair queuing (CBWFQ) can be used. If you want to use one of these alternative mechanisms, enable it before configuring Multilink.

Enable FRTS

- Frame Relay Traffic Shaping (FRTS) must be enabled on the Frame Relay interface.

Restrictions for Using Multilink PPP over Frame Relay

Number of Links per Multilink Bundle

Only one link per multilink bundle is supported.

VoIP Support

Only Voice over IP (VoIP) is supported; Voice over Frame Relay (VoFR) is not supported.

Information About Using Multilink PPP over Frame Relay

To use Multilink PPP over Frame Relay, you should understand the following concepts:

- [Frame Relay Traffic Shaping and Multilink PPP over Frame Relay, page 690](#)
- [MQC and Multilink PPP over Frame Relay, page 691](#)
- [Multilink Group Interfaces, page 691](#)

Frame Relay Traffic Shaping and Multilink PPP over Frame Relay

Before using Multilink PPP over Frame Relay, FRTS must be enabled.



Note

On the Cisco 7200 and lower series of routers, the **frame-relay traffic-shaping** command is used to enable FRTS. On the Cisco 7500 and higher series of routers, the Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC) is used to enable FRTS. For more information about MQC, see the “[MQC and Multilink PPP over Frame Relay](#)” section on page 691.

FRTS is a Cisco traffic shaping mechanism. A traffic shaping mechanism allows you to regulate (that is, “shape”) the packet flow on a network. When you shape traffic, you control the speed of traffic leaving an interface. This way, you can match the flow of the traffic to the speed of the interface and avoid bottlenecks on the network.

Cisco has long provided support for forward explicit congestion notification (FECN) for DECnet and OSI, and backward explicit congestion notification (BECN) for Systems Network Architecture (SNA) traffic using Logical Link Control, type 2 (LLC2) encapsulation via RFC 1490 and discard eligible (DE) bit support. FRTS builds upon this existing Frame Relay support with additional capabilities that improve the scalability and performance of a Frame Relay network, increasing the density of virtual circuits (VCs) and improving response time.

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

MQC and Multilink PPP over Frame Relay

Before using Multilink PPP over Frame Relay, a policy map must be created. (See the [“Prerequisites” section on page 694](#).) Policy maps are created using the Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC).

The MQC is a CLI structure that allows users to create traffic policies (policy maps) and attach these policy maps to interfaces. A policy map contains a traffic class and one or more QoS features. A traffic class is used to classify traffic. The QoS features in the traffic policy determine how to treat the classified traffic.

For more information about MQC, see the “Modular Quality of Service Command-Line Interface” section of the [Cisco IOS Quality of Service Solutions Configuration Guide](#), Release 12.3.

Virtual Template Interfaces

A virtual template interface is logical interface configured with generic configuration information for a specific purpose or configuration 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.

For more information virtual template interfaces, see the [Cisco IOS Dial Technologies Configuration Guide](#), Release 12.3.

Multilink Group Interfaces

A multilink group interface is a collection of interfaces bundled together in the multilink PPP configuration. With a multilink group interface, you can bundle interfaces into logical multilink groups.

For more information about multilink group interfaces and multilink groups, refer to the [Cisco IOS Dial Technologies Configuration Guide](#), Release 12.3.

How to Configure Multilink PPP over Frame Relay

This section contains the procedures for configuring Multilink PPP over Frame Relay.

**Note**

While the first two procedures are listed as optional, you must choose one or the other according to the Cisco router you are using on your network.

- [Configuring Multilink PPP over Frame Relay on a Virtual Template Interface, page 692](#) (optional; applies only if you are using the Cisco 7500 series router or the Cisco 7600 series router)
or
- [Configuring Multilink PPP over Frame Relay on a Multilink Group Interface, page 694](#) (optional)
- [Associating the Virtual Template Interface with a Frame Relay PVC, page 697](#) (required)
- [Verifying the Multilink PPP over Frame Relay Configuration, page 698](#) (optional)

Configuring Multilink PPP over Frame Relay on a Virtual Template Interface

To configure Multilink PPP over Frame Relay on a virtual template interface, complete the following steps.



Note

These steps apply if you are using the Cisco 7500 series router or the Cisco 7600 series router only. If you are using another series of Cisco router, do not complete these steps. Instead, advance to [“Configuring Multilink PPP over Frame Relay on a Multilink Group Interface” section on page 694](#).

Prerequisites

Before proceeding with this task, you must create a policy map. The policy map contains the configuration parameters used to apply a specific QoS features such as distributed LLQ (dLLQ) to the network traffic. To create a policy map and configure the appropriate QoS feature, use the MQC. See the [“MQC and Multilink PPP over Frame Relay” section on page 691](#).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface virtual-template** *number*
4. **bandwidth** *kbps*
5. **ip address** *ip-address mask* [**secondary**]
6. **service-policy output** *policy-map-name*
7. **service-policy input** *policy-map-name*
8. **ppp multilink**
9. **ppp multilink fragment delay** *milliseconds* [*microseconds*]
10. **ppp multilink interleave**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>enable</p> <p>Example: Router> enable</p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	<p>configure terminal</p> <p>Example: Router# configure terminal</p>	<p>Enters global configuration mode.</p>
Step 3	<p>interface virtual-template <i>number</i></p> <p>Example: Router(config)# interface virtual-template 4</p>	<p>Creates a virtual template and enters interface configuration mode.</p> <ul style="list-style-type: none"> Enter the virtual template number.
Step 4	<p>bandwidth <i>kbps</i></p> <p>Example: Router(config-if)# bandwidth 32</p>	<p>Sets the bandwidth value for an interface.</p> <ul style="list-style-type: none"> Enter the bandwidth value in kilobits per second. <p>Note The bandwidth value for the interface should match the traffic speed of the PVC; for instance, if the VBR peak cell rate is 128 kbps, the <i>kbps</i> option in the bandwidth command should be entered as 128. Similarly, if the PVC is being shaped to 64 kbps, the <i>kbps</i> option should be entered as 64.</p>
Step 5	<p>ip address <i>ip-address mask [secondary]</i></p> <p>Example: Router(config-if)# ip address 10.10.100.1 255.255.255.0</p>	<p>Sets a primary IP address for an interface. This command can also set the optional secondary IP address for an interface.</p> <ul style="list-style-type: none"> Enter the primary IP address (and, optionally, the secondary IP address).
Step 6	<p>service-policy output <i>policy-map-name</i></p> <p>Example: Router(config-if)# service-policy output policy1</p>	<p>Attaches the previously created QoS traffic policy (policy map). See the “Prerequisites” section on page 692. The policy map evaluates and applies QoS features for traffic <i>leaving</i> the interface.</p> <ul style="list-style-type: none"> Enter the policy map name.
Step 7	<p>service-policy input <i>policy-map-name</i></p> <p>Example: Router(config-if)# service-policy input policy1</p>	<p>Attaches the previously created QoS traffic policy (policy map). See the “Prerequisites” section on page 692. The policy map evaluates and applies QoS features for traffic <i>entering</i> the interface.</p> <ul style="list-style-type: none"> Enter the policy map name.
Step 8	<p>ppp multilink</p> <p>Example: Router(config-if)# ppp multilink</p>	<p>Enables MLP on the interface.</p>

	Command or Action	Purpose
Step 9	<code>ppp multilink fragment delay milliseconds</code> [<i>microseconds</i>] Example: Router(config-if)# ppp multilink fragment delay 20	Specifies a maximum size in units of time for packet fragments on a Multilink PPP (MLP) bundle. <ul style="list-style-type: none"> Enter the maximum amount of time, in milliseconds.
Step 10	<code>ppp multilink interleave</code> Example: Router(config-if)# ppp multilink interleave	Enables interleaving of packets among the fragments of larger packets on a multilink bundle.
Step 11	<code>end</code> Example: Router(config-if)# end	(Optional) Exits interface configuration mode.

The fragment size can be configured using the following formula:

$$\text{fragment size} = \text{bandwidth} \times \text{fragment-delay} / 8$$

Configuring Multilink PPP over Frame Relay on a Multilink Group Interface

To configure Multilink PPP over Frame Relay on a multilink group interface, complete the following steps.



Note

If you are using the Cisco 7500 series router or the Cisco 7600 series router, do not complete these steps. Instead, complete the steps in [“Configuring Multilink PPP over Frame Relay on a Virtual Template Interface”](#) section on page 692.

Prerequisites

Before proceeding with this task, you must create a policy map. The policy map contains the configuration parameters used to apply a specific QoS features such as distributed LLQ (dLLQ) to the network traffic. To create a policy map and configure the appropriate QoS feature, use the MQC. See the [“MQC and Multilink PPP over Frame Relay”](#) section on page 691.

SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `interface multilink multilink-bundle-number`
4. `ip address ip-address mask [secondary]`
5. `service-policy output policy-map-name`
6. `service-policy input policy-map-name`
7. `ppp multilink fragment delay milliseconds [microseconds]`

8. `ppp multilink interleave`
9. `end`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p><code>enable</code></p> <p>Example: Router> enable</p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	<p><code>configure terminal</code></p> <p>Example: Router# configure terminal</p>	<p>Enters global configuration mode.</p>
Step 3	<p><code>interface multilink multilink-bundle-number</code></p> <p>Example: Router(config)# interface multilink 1</p>	<p>Creates a multilink bundle and enters interface configuration mode.</p> <ul style="list-style-type: none"> Enter the multilink bundle number.
Step 4	<p><code>ip address ip-address mask [secondary]</code></p> <p>Example: Router(config-if)# ip address 10.10.100.1 255.255.255.0</p>	<p>Sets a primary IP address for an interface. This command can also set the optional secondary IP address for an interface.</p> <ul style="list-style-type: none"> Enter the primary IP address (and, optionally, the secondary IP address).
Step 5	<p><code>service-policy output policy-map-name</code></p> <p>Example: Router(config-if)# service-policy output policy1</p>	<p>Attaches the previously created QoS traffic policy (policy map). See the “Prerequisites” section on page 694. The policy map evaluates and applies QoS features for traffic <i>leaving</i> the interface.</p> <ul style="list-style-type: none"> Enter the policy map name.
Step 6	<p><code>service-policy input policy-map-name</code></p> <p>Example: Router(config-if)# service-policy input policy1</p>	<p>Attaches the previously created QoS traffic policy (policy map). See the “Prerequisites” section on page 694. The policy map evaluates and applies QoS features for traffic <i>entering</i> the interface.</p> <ul style="list-style-type: none"> Enter the policy map name.
Step 7	<p><code>ppp multilink fragment delay milliseconds [microseconds]</code></p> <p>Example: Router(config-if)# ppp multilink fragment delay 20</p>	<p>Specifies a maximum size in units of time for packet fragments on a multilink bundle.</p> <ul style="list-style-type: none"> Enter the maximum amount of time, in milliseconds, required to transmit a fragment.

	Command or Action	Purpose
Step 8	<code>ppp multilink interleave</code> Example: Router(config-if)# <code>ppp multilink interleave</code>	Enables interleaving of packets among the fragments of larger packets on a multilink bundle.
Step 9	<code>end</code> Example: Router(config-if)# <code>end</code>	(Optional) Exits interface configuration mode.

What to Do Next

After configuring Multilink PPP over Frame Relay on a multilink group interface, the next step is to associate the virtual template interface with the multilink group by completing the steps in the following section.

If you are using a Cisco 7500 series router or a Cisco 7600 series router, advance to [“Associating the Virtual Template Interface with a Frame Relay PVC” section on page 697](#) to continue.

Associating the Virtual Template Interface with the Multilink Group

To associate the virtual template interface with the multilink group, complete the following steps.

SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `interface virtual-template number`
4. `no ip address`
5. `ppp multilink group group-number`
6. `end`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>enable</code> Example: Router> <code>enable</code>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<code>configure terminal</code> Example: Router# <code>configure terminal</code>	Enters global configuration mode.

	Command or Action	Purpose
Step 3	interface virtual-template <i>number</i> Example: Router# interface virtual-template 1	Creates a virtual template interface that can be configured and applied dynamically in creating virtual access interfaces, and enters interface configuration mode. <ul style="list-style-type: none"> Enter the number used to identify the virtual template interface.
Step 4	no ip address Example: Router(config-if)# no ip address	Removes an IP address or disables IP processing.
Step 5	ppp multilink group <i>group-number</i> Example: Router(config-if)# ppp multilink group 1	Restricts a physical link to joining only a designated multilink group interface. <ul style="list-style-type: none"> Enter the multilink group number.
Step 6	end Example: Router(config-if)# end	(Optional) Exits interface configuration mode.

Associating the Virtual Template Interface with a Frame Relay PVC

To associate the virtual template interface with the Frame Relay PVC, complete the following steps.

SUMMARY STEPS

- enable
- configure terminal
- interface *type number* [**name-tag**]
- frame-relay traffic-shaping
- frame-relay interface-dlci *dlci* [**ietf** | **cisco**] [**voice-cir** *cir*] [**ppp** *virtual-template-name*]
- class *name*
- end

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	interface <i>type number</i> [name-tag] Example: Router(config)# interface serial1/0/0/1:0	Configures an interface type and enters interface configuration mode. • Enter the interface type and number.
Step 4	frame-relay traffic-shaping Example: Router(config-if)# frame-relay traffic-shaping	Enables both traffic shaping and per-virtual-circuit queuing for all permanent virtual circuits (PVCs) and switched virtual circuits (SVCs) on a Frame Relay interface. Note Use this command on Cisco 7200 and lower series routers <i>only</i> . Do not use this command on Cisco 7500 or higher series routers. For Cisco 7500 and higher series routers, use the MQC instead of this command.
Step 5	frame-relay interface-dlci <i>dlci</i> [ietf cisco] [voice-cir <i>cir</i>] [ppp <i>virtual-template-name</i>] Example: Router(config-if)# frame-relay interface-dlci 100 ppp virtual-templatel	Assigns a data-link connection identifier (DLCI) to a specified Frame Relay subinterface on the router or access server, assigns a specific PVC to a DLCI, or applies a virtual template configuration for a PPP session. Enters Frame Relay DLCI configuration mode. • Enter the DLCI number and any optional keywords and arguments, as appropriate.
Step 6	class <i>name</i> Example: Router(config-fr-dlci)# class frdlci	Associates a map class with a specified DLCI. • Enter the name of the map class to associate with the specified DLCI. Note Use this command on Cisco 7200 and lower series routers <i>only</i> . For Cisco 7500 and higher series routers, this command is not needed.
Step 7	end Example: Router(config-fr-dlci)# end	(Optional) Exits Frame Relay DLCI configuration mode.

Verifying the Multilink PPP over Frame Relay Configuration

To verify the Multilink PPP over Frame Relay configuration, complete the following steps.

SUMMARY STEPS

1. **enable**
2. **show frame-relay pvc** [**interface** *interface*] [**dlci**] [**64-bit**]
3. **show interfaces** [*type number*] [*first*] [*last*] [**accounting**]
4. **show ppp multilink** [**active** | **inactive** | **interface** *bundle-interface* | [**username** *name*] [**endpoint** *endpoint*]]
5. **show policy-map interface** *interface-name* [**vc** [*vpi/*] *vci*] [**dlci** *dlci*] [**input** | **output**]
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 frame-relay pvc [interface <i>interface</i>] [dlci] [64-bit] Example: Router# show frame-relay pvc	(Optional) Displays statistics about permanent virtual circuits (PVCs) for Frame Relay interfaces.
Step 3	show interfaces [<i>type number</i>] [<i>first</i>] [<i>last</i>] [accounting] Example: Router# show interfaces	(Optional) Displays statistics for all interfaces configured on the router or access server.
Step 4	show ppp multilink [active inactive interface <i>bundle-interface</i> [username <i>name</i>] [endpoint <i>endpoint</i>]] Example: Router# show ppp multilink	(Optional) Displays bundle information for multilink bundles.
Step 5	show policy-map interface <i>interface-name</i> [vc [<i>vpi/</i>] <i>vci</i>] [dlci <i>dlci</i>] [input output] Example: Router# show policy-map interface serial0/0	(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.
Step 6	exit Example: Router# exit	(Optional) Exits privileged EXEC mode.

Configuration Examples for Multilink PPP over Frame Relay

This section contains the following examples:

- [Configuring Multilink PPP over Frame Relay on a Virtual Template Interface: Example, page 700](#)
- [Configuring Multilink PPP over Frame Relay on a Multilink Group Interface: Example, page 700](#)
- [Associating the Virtual Template Interface with the Multilink Group: Example, page 700](#)
- [Associating the Virtual Template Interface with a Frame Relay PVC: Example, page 701](#)
- [Verifying the Multilink PPP over Frame Relay Configuration: Example, page 701](#)

Configuring Multilink PPP over Frame Relay on a Virtual Template Interface: Example

The following is an example of configuring Multilink PPP over Frame Relay on a virtual template interface:

```
Router> enable
Router# configure terminal
Router(config)# interface virtual-template 4
Router(config-if)# bandwidth 32
Router(config-if)# ip address 10.10.100.1 255.255.255.0
Router(config-if)# service-policy output policy1
Router(config-if)# service-policy input policy1
Router(config-if)# ppp multilink
Router(config-if)# ppp multilink fragment delay 20
Router(config-if)# ppp multilink interleave
Router(config-if)# end
```

Configuring Multilink PPP over Frame Relay on a Multilink Group Interface: Example

The following is an example of configuring Multilink PPP over Frame Relay on a multilink group interface:

```
Router> enable
Router# configure terminal
Router(config)# interface multilink 1
Router(config-if)# ip address 10.10.100.1 255.255.255.0
Router(config-if)# service-policy output policy1
Router(config-if)# service-policy input policy1
Router(config-if)# ppp multilink fragment delay 20
Router(config-if)# ppp multilink interleave
Router(config-if)# end
```

Associating the Virtual Template Interface with the Multilink Group: Example

The following is an example of associating the virtual template interface with the multilink group:

```
Router> enable
Router# configure terminal
Router(config)# interface virtual-template 1
Router(config-if)# no ip address
Router(config-if)# ppp multilink group 1
Router(config-if)# end
```

Associating the Virtual Template Interface with a Frame Relay PVC: Example

The following is an example of associating the virtual template interface with a Frame Relay PVC:

```
Router> enable
Router# configure terminal
Router(config)# interface serial1/0/0/1:0
Router(config-if)# frame-relay interface-dlci 100 ppp virtual-template1
Router(config-fr-dlci)# class frdlci
Router(config-fr-dlci)# end
```

Verifying the Multilink PPP over Frame Relay Configuration: Example

You can verify the Multilink with PPP over Frame Relay configuration by using one or more of the following **show** commands:

- **show frame relay pvc**
- **show interfaces**
- **show ppp multilink**
- **show policy-map interface**

The following section provides sample output of the **show ppp multilink** command only. For sample output of the other commands, see the appropriate Cisco IOS Release 12.3 T command reference publication.

show ppp multilink Command Output Example

The following is an example of the **show ppp multilink** command output. In this example, one Multilink bundle called 7206-2 is on the system. This bundle has two member links: one active link and one inactive link.

```
Router# show ppp multilink

Multilink1, bundle name is 7206-2
  Endpoint discriminator is 7206-2
  Bundle up for 00:00:15, 1/255 load
  Receive buffer limit 12000 bytes, frag timeout 3428 ms
    0/0 fragments/bytes in reassembly list
    1 lost fragments, 1 reordered
    0/0 discarded fragments/bytes, 0 lost received
    0x3 received sequence, 0x3 sent sequence
  Member links:1 active, 1 inactive (max not set, min not set)
  Vi2, since 00:00:15, 105 weight, 93 frag size
  Vt1 (inactive)
```

Where to Go Next

To use Multilink PPP over ATM links, see the “” module.

To use Multilink PPP over dialer interface links, see the “” module.

To use Multilink PPP over serial interface links, see the “” module.

Additional References

The following sections provide references related to using Multilink PPP over Frame Relay.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	<i>Cisco IOS Quality of Service Solutions Command Reference</i> , Release 12.3 T
LLQ, dLLQ, WFQ, CBWFQ and other queuing mechanisms	“Congestion Management” section of the <i>Cisco IOS Quality of Service Solutions Configuration Guide</i> , Release 12.3
MQC	“Modular Quality of Service Command-Line Interface” section of the <i>Cisco IOS Quality of Service Solutions Configuration Guide</i> , Release 12.3
FRTS	<i>Cisco IOS Wide-Area Networking Configuration Guide</i> , Release 12.3
Multilink PPP configurations and virtual template interfaces	<i>Cisco IOS Dial Technologies Configuration Guide</i> , Release 12.3
Multilink PPP overview module	“” module
Multilink PPP over ATM links (including ATM interfaces and ATM PVCs)	“” module
Multilink PPP dialer interface links	“” module
Multilink PPP serial interface links	“” 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 1990	<i>The PPP Multilink Protocol (MP)</i>
RFC 2686	<i>Multiclass Extension to Multilink PPP (MCML)</i>

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

BECN—backward explicit congestion notification. Bit set by a Frame Relay network in frames traveling in the opposite direction of frames encountering a congested path. DTE receiving frames with the BECN bit set can request that higher-level protocols take flow control action as appropriate. Compare with FECN.

DE—discard eligible. If the network is congested, DE traffic can be dropped to ensure the delivery of higher priority traffic.

FECN—forward explicit congestion notification. Bit set by a Frame Relay network to inform DTE receiving the frame that congestion was experienced in the path from source to destination. DTE receiving frames with the FECN bit set can request that higher-level protocols take flow-control action as appropriate. Compare with BECN.

LLC2—Logical Link Control, type 2. Connection-oriented Open System Interconnection (OSI) LLC-sublayer protocol.

PVC—permanent virtual circuit (or connection). 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.

virtual template interface—A logical interface configured with generic configuration information for a specific purpose or configuration 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.



Note

See [Internetworking Terms and Acronyms](#) for terms not included in this glossary.

Feature Information for Using Multilink PPP over Frame Relay

Table 8 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.

For information on a feature in this technology that is not documented here, see the “[Reducing Latency and Jitter Using Multilink PPP Roadmap](#).”

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 Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

**Note**

Table 8 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 8 **Feature Information for Using Multilink PPP over Frame Relay**

Feature Name	Software Releases	Feature Configuration Information
Distributed Link Fragmentation and Interleaving Over Leased Lines	12.2(8)T	<p>The Distributed Link Fragmentation and Interleaving over Leased Lines feature extends distributed link fragmentation and interleaving functionality to leased lines.</p> <p>This feature was extensively rewritten from the perspective of using Multilink PPP for link fragmentation and interleaving over Frame Relay.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Information About Using Multilink PPP over Frame Relay, page 690 • How to Configure Multilink PPP over Frame Relay, page 691
Distributed Link Fragmentation and Interleaving for Frame Relay and ATM Interfaces on Cisco 7500 Series Routers	12.2(4)T	<p>The Distributed Link Fragmentation and Interleaving (dLFI) for Frame Relay and ATM Interfaces on Cisco 7500 Series Routers feature extends link fragmentation and interleaving functionality to VIP-enabled Cisco 7500 series routers.</p> <p>This feature was extensively rewritten from the perspective of using Multilink PPP for link fragmentation and interleaving over Frame Relay.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Information About Using Multilink PPP over Frame Relay, page 690 • How to Configure Multilink PPP over Frame Relay, page 691



Using Multilink PPP over Serial Interface Links

Multilink PPP is a method used to reduce latency and jitter for real-time traffic. This module contains conceptual information and configuration tasks for using Multilink PPP over serial interface links.

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 features. To find information about feature support and configuration, use the [“Feature Information for Using Multilink PPP over Serial Interface Links”](#) section on page 717.

Contents

- [Prerequisites for Using Multilink PPP over Serial Interface Links, page 707](#)
- [Restrictions for Using Multilink PPP over Serial Interface Links, page 708](#)
- [Information About Using Multilink PPP over Serial Interface Links, page 708](#)
- [How to Configure Multilink PPP over Serial Interface Links, page 709](#)
- [Configuration Examples for Using Multilink PPP over Serial Interface Links, page 713](#)
- [Where to Go Next, page 714](#)
- [Additional References, page 715](#)
- [Glossary, page 716](#)
- [Feature Information for Using Multilink PPP over Serial Interface Links, page 717](#)

Prerequisites for Using Multilink PPP over Serial Interface Links

Knowledge

- Be familiar with the concepts in the “” module.

Enable Queuing Mechanism

- Multilink uses first-in first out (FIFO) queuing for queuing and interleaving packets. Other queuing mechanisms such as low latency queuing (LLQ), weighted fair queuing (WFQ), and class-based weighted fair queuing (CBWFQ) can be used. If you want to use one of these alternative mechanisms, enable it before configuring Multilink.

Restrictions for Using Multilink PPP over Serial Interface Links

Number of Links per Multilink Bundle

Only one link per multilink bundle is supported.

VoIP Support

Only Voice over IP (VoIP) is supported.

Queuing Mechanisms Not Supported

Many of the legacy queuing mechanisms are not supported by multilink. These mechanisms include:

- Fair queuing on a virtual template interface
- Weighted random early detection (WRED) on a virtual template interface
- Custom queuing
- Priority queuing



Note Fair queuing, WRED, and priority queuing can be configured in a traffic policy using the Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC).

Information About Using Multilink PPP over Serial Interface Links

To use Multilink PPP over serial interface links, you should understand the following concept:

- [MQC and Multilink PPP over Serial Interface Links, page 708](#)
- [Multilink Group Interfaces, page 709](#)

MQC and Multilink PPP over Serial Interface Links

Before using Multilink PPP over serial interface links, a traffic policy (also known as a policy map) must be created. (See the “Prerequisites” section on page 709.) Policy maps are created using the Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC).

The MQC is a CLI structure that allows users to create traffic polices (policy maps) and attach these policy maps to interfaces. A policy map contains a traffic class and one or more QoS features. A traffic class is used to classify traffic. The QoS features in the traffic policy determine how to treat the classified traffic.

For more information about MQC, see the “Modular Quality of Service Command-Line Interface” section of the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.3.

Multilink Group Interfaces

A multilink group interface is a collection of interfaces bundled together in the multilink PPP configuration. With a multilink group interface, you can bundle interfaces into logical multilink groups.

For more information about multilink group interfaces and multilink groups, see the *Cisco IOS Dial Technologies Configuration Guide*, Release 12.3.

How to Configure Multilink PPP over Serial Interface Links

This section contains the following procedures:

- [Configuring Multilink PPP over Serial Interface Links on a Multilink Group Interface](#), page 709 (required)
- [Associating the Serial Interface with the Multilink Group](#), page 711 (required)
- [Verifying the Multilink PPP over Serial Interface Link Configuration](#), page 712 (optional)

Configuring Multilink PPP over Serial Interface Links on a Multilink Group Interface

To configure Multilink PPP over serial interface links on a multilink group interface, complete the following steps.

Prerequisites

Before proceeding with this task, you must create a policy map. The policy map contains the configuration parameters used to apply the specific quality of service feature to the network traffic. To create a policy map, use the MQC. See the “MQC and Multilink PPP over Serial Interface Links” section on page 708.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface multilink** *multilink-bundle-number*
4. **ip address** *ip-address mask* [**secondary**]
5. **service-policy output** *policy-map-name*
6. **service-policy input** *policy-map-name*
7. **ppp multilink fragment delay** *milliseconds* [*microseconds*]
8. **ppp multilink interleave**
9. **ppp multilink multiclass**

10. end

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface multilink <i>multilink-bundle-number</i> Example: Router(config)# interface multilink 1	Creates a multilink bundle and enters interface configuration mode. <ul style="list-style-type: none"> Enter the multilink bundle number.
Step 4	ip address <i>ip-address mask [secondary]</i> Example: Router(config-if)# ip address 10.10.100.1 255.255.255.0	Sets a primary IP address for an interface. This command can also set the optional secondary IP address for an interface. <ul style="list-style-type: none"> Enter the primary IP address (and, optionally, the secondary IP address).
Step 5	service-policy output <i>policy-map-name</i> Example: Router(config-if)# service-policy output policy1	Attaches the previously created QoS traffic policy (policy map). See the “Prerequisites” section on page 709. The policy map evaluates and applies QoS features for traffic <i>leaving</i> the interface. <ul style="list-style-type: none"> Enter the policy map name.
Step 6	service-policy input <i>policy-map-name</i> Example: Router(config-if)# service-policy input policy1	Attaches the previously created QoS traffic policy (policy map). See the “Prerequisites” section on page 709. The policy map evaluates and applies QoS features for traffic <i>entering</i> the interface. <ul style="list-style-type: none"> Enter the policy map name.
Step 7	ppp multilink fragment delay <i>milliseconds [microseconds]</i> Example: Router(config-if)# ppp multilink fragment delay 20	Specifies a maximum size in units of time for packet fragments on a Multilink PPP (MLP) bundle. <ul style="list-style-type: none"> Enter the maximum amount of time, in milliseconds.
Step 8	ppp multilink interleave Example: Router(config-if)# ppp multilink interleave	Enables interleaving of packets among the fragments of larger packets on a multilink bundle.

	Command or Action	Purpose
Step 9	ppp multilink multiclass Example: Router(config-if)# ppp multilink multiclass	(Optional) Enables Multiclass Multilink PPP (MCMP) on an interface. Note Use this command only if there are multiple links in the multilink bundle.
Step 10	end Example: Router(config-if)# end	(Optional) Exits interface configuration mode.

Associating the Serial Interface with the Multilink Group

To associate the serial interface with the multilink group, complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface serial** *slot/port:timeslot*
4. **no fair-queue**
5. **ppp multilink**
6. **ppp multilink group** *group-number*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface serial <i>slot/port:timeslot</i> Example: Router# interface serial 4/1:23	Specifies a serial interface created on a channelized E1 or channelized T1 controller (for ISDN PRI, channel-associated signalling, or robbed-bit signalling), and enters interface configuration mode. <ul style="list-style-type: none"> • Enter the slot number and port number where the channelized E1 or T1 controller is located.

	Command or Action	Purpose
Step 4	no fair-queue Example: Router(config-if)# no fair-queue	Disables WFQ (or DWFQ for VIP-enabled routers).
Step 5	ppp multilink Example: Router(config-if)# ppp multilink	Enables Multilink on an interface.
Step 6	ppp multilink group group-number Example: Router(config-if)# ppp multilink group 1	Restricts a physical link to joining only a designated multilink group interface. <ul style="list-style-type: none"> Enter the multilink group number.
Step 7	end Example: Router(config-if)# end	(Optional) Exits interface configuration mode.

Verifying the Multilink PPP over Serial Interface Link Configuration

To verify the Multilink PPP over serial interface link configuration, complete the following steps.

SUMMARY STEPS

- enable**
- show interfaces** [*type number*] [*first*] [*last*] [**accounting**]
- show ppp multilink** [**active** | **inactive** | **interface bundle-interface** | [**username name**] [**endpoint endpoint**]]
- show policy-map interface interface-name** [**vc** [*vpi*] *vci*] [**dlsi dlci**] [**input** | **output**]
- exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	show interfaces [<i>type number</i>] [<i>first</i>] [<i>last</i>] [accounting] Example: Router# show interfaces	(Optional) Displays statistics for all interfaces configured on the router or access server.

	Command or Action	Purpose
Step 3	<pre>show ppp multilink [active inactive interface bundle-interface [username name] [endpoint endpoint]]</pre> <p>Example: Router# show ppp multilink</p>	(Optional) Displays bundle information for multilink bundles.
Step 4	<pre>show policy-map interface interface-name [vc [vpi/] vci] [dlci dlci] [input output]</pre> <p>Example: Router# show policy-map interface serial0/0</p>	(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 permanent virtual circuit (PVC) on the interface.
Step 5	<pre>exit</pre> <p>Example: Router# exit</p>	(Optional) Exits privileged EXEC mode.

Configuration Examples for Using Multilink PPP over Serial Interface Links

This section contains the following examples:

- [Configuring Multilink PPP over Serial Interface Links on a Multilink Group Interface: Example, page 713](#)
- [Associating the Serial Interface with the Multilink Group: Example, page 714](#)
- [Verifying the Multilink PPP over Serial Interface Link Configuration: Example, page 714](#)

Configuring Multilink PPP over Serial Interface Links on a Multilink Group Interface: Example

The following is an example of configuring Multilink PPP over serial interface links on a multilink group interface:

```
Router> enable
Router# configure terminal
Router(config)# interface multilink 1
Router(config-if)# ip address 10.10.100.1 255.255.255.0
Router(config-if)# service-policy output policy1
Router(config-if)# service-policy input policy1
Router(config-if)# ppp multilink fragment delay 20
Router(config-if)# ppp multilink interleave
Router(config-if)# ppp multilink multiclass
Router(config-if)# end
```

Associating the Serial Interface with the Multilink Group: Example

The following is an example of associating the serial interface serial4/1 with the multilink group:

```
Router> enable
Router# configure terminal
Router(config)# interface serial 4/1:23
Router(config-if)# no fair-queue
Router(config-if)# ppp multilink
Router(config-if)# ppp multilink group 1
Router(config-if)# end
```

Verifying the Multilink PPP over Serial Interface Link Configuration: Example

You can verify the Multilink PPP over serial interface links configuration by using one or more of the following **show** commands:

- **show interfaces**
- **show ppp multilink**
- **show policy-map interface**

The following section provides sample output of the **show ppp multilink** command only. For sample output of the other commands, see the appropriate Cisco IOS Release 12.3 T command reference publication.

show ppp multilink Command Output Example

The following is an example of the **show ppp multilink** command output. In this example, one multilink bundle called 7206-2 is on the system. This bundle has two member links: one active link and one inactive link.

```
Router# show ppp multilink

Multilink2, bundle name is 7206-2
Endpoint discriminator is 7206-2
Bundle up for 00:00:09, 1/255 load
Receive buffer limit 12000 bytes, frag timeout 1500 ms
 0/0 fragments/bytes in reassembly list
 0 lost fragments, 0 reordered
 0/0 discarded fragments/bytes, 0 lost received
 0x0 received sequence, 0x3 sent sequence
Member links:1 active, 1 inactive (max not set, min not set)
  Se3/2, since 00:00:10, 240 weight, 232 frag size
  Se3/3 (inactive)
```

Where to Go Next

To use Multilink PPP over ATM links, see the “” module.

To use Multilink PPP over Frame Relay, see the “” module.

To use Multilink PPP over dialer interface links, see the “” module.

Additional References

The following sections provide references related to Multilink PPP over serial interface links:

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	<i>Cisco IOS Quality of Service Solutions Command Reference</i> , Release 12.3 T
LLQ, dLLQ, WFQ, CBWFQ and other queuing mechanisms	“Congestion Management” section of the <i>Cisco IOS Quality of Service Solutions Configuration Guide</i> , Release 12.3
MQC	“Modular Quality of Service Command-Line Interface” section of the <i>Cisco IOS Quality of Service Solutions Configuration Guide</i> , Release 12.3
Multilink PPP configuration information	<i>Cisco IOS Dial Technologies Configuration Guide</i> , Release 12.3
Multilink PPP overview module	“” module
Multilink PPP over ATM links (including ATM interfaces and ATM PVCs)	“Using Multilink PPP over ATM Links” module
Multilink PPP over Frame Relay	“” module
Multilink PPP over dialer interface links	“” 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 1990	<i>The PPP Multilink Protocol (MP)</i>
RFC 2686	<i>Multiclass Extension to Multilink PPP (MCML)</i>

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

PVC—permanent virtual circuit (or connection). 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.

virtual template interface—A logical interface configured with generic configuration information for a specific purpose or configuration 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.



Note

See *Internetworking Terms and Acronyms* for terms not included in this glossary.

Feature Information for Using Multilink PPP over Serial Interface Links

Table 9 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.

For information on a feature in this technology that is not documented here, see the “[Reducing Latency and Jitter Using Multilink PPP Roadmap](#).”

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 Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

**Note**

Table 9 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 9 Feature Information for Using Multilink PPP over Serial Interface Links

Feature Name	Software Releases	Feature Configuration Information
Distributed Link Fragmentation and Interleaving Over Leased Lines	12.2(8)T	<p>The Distributed Link Fragmentation and Interleaving over Leased Lines feature extends distributed link fragmentation and interleaving functionality to leased lines.</p> <p>This feature was extensively rewritten from the perspective of using Multilink PPP for link fragmentation and interleaving over serial interface links.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Information About Using Multilink PPP over Serial Interface Links, page 708 • How to Configure Multilink PPP over Serial Interface Links, page 709
Distributed Link Fragmentation and Interleaving for Frame Relay and ATM Interfaces on Cisco 7500 Series Routers	12.2(4)T	<p>The Distributed Link Fragmentation and Interleaving (dLFI) for Frame Relay and ATM Interfaces on Cisco 7500 Series Routers feature extends link fragmentation and interleaving functionality to VIP-enabled Cisco 7500 series routers.</p> <p>This feature was extensively rewritten from the perspective of using Multilink PPP for link fragmentation and interleaving over serial interface links.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Information About Using Multilink PPP over Serial Interface Links, page 708 • How to Configure Multilink PPP over Serial Interface Links, page 709



Header Compression

This part consists of the following:

- [Header-Compression Features Roadmap](#)
- [Header Compression](#)
- [Configuring RTP Header Compression](#)
- [Configuring TCP Header Compression](#)
- [Configuring Class-Based RTP and TCP Header Compression](#)





Header-Compression Features Roadmap

First Published: January 30, 2006
Last Updated: January 30, 2006

This roadmap lists the header-compression features and maps them to the modules in which they appear.

Feature and Release Support

Table 10 lists header-compression feature support for the following Cisco IOS software release trains:

- [Cisco IOS Releases 12.2T, 12.3, and 12.3T](#)

Only features that were introduced or modified in Cisco IOS Release 12.2(1) or a later release appear in the table. *Not all features may be supported in your Cisco IOS software release.*

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 Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.



Note

Table 10 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 10 **Supported Header-Compression Features**

Release	Feature Name	Feature Description	Where Documented
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”

Table 10 **Supported Header-Compression Features (continued)**

Release	Feature Name	Feature Description	Where Documented
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: January 30, 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, page 723](#)
- [Where to Go Next, page 728](#)
- [Additional References, page 728](#)
- [Glossary, page 730](#)

Information About Header Compression

Before configuring header compression, you should understand the following concepts:

- [Header Compression Defined, page 724](#)
- [Types of Header Compression, page 724](#)
- [RTP Functionality and Header Compression, page 724](#)
- [TCP Functionality and Header Compression, page 726](#)
- [Class-Based Header Compression Functionality, page 727](#)

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.

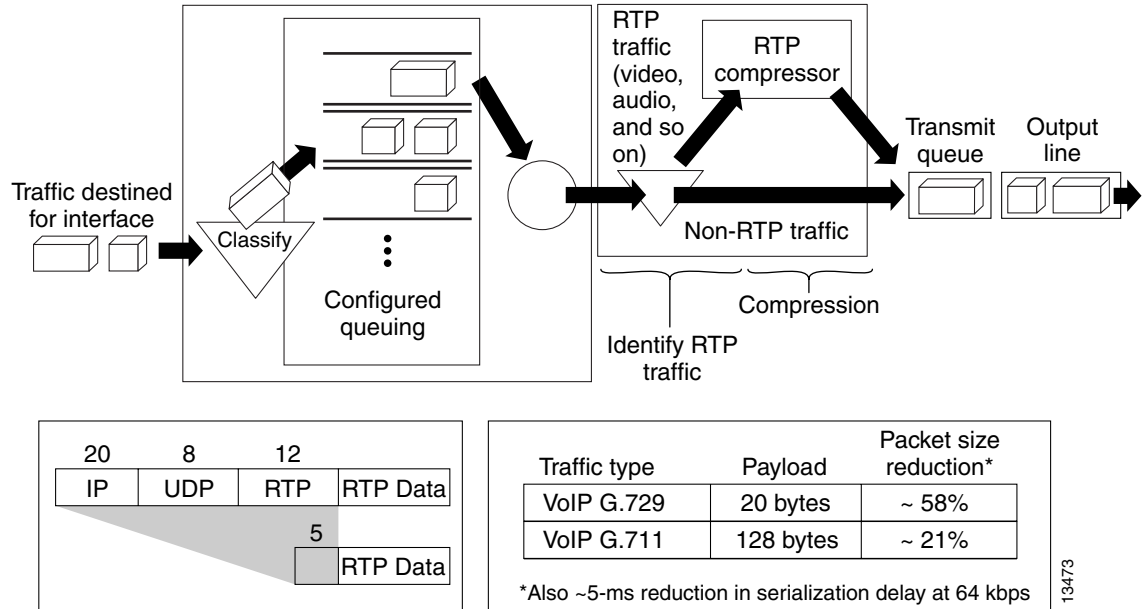
To avoid the unnecessary consumption of available bandwidth, RTP header compression is used on a link-by-link basis.

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 38](#) 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 38 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 38](#), 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 38](#), 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.

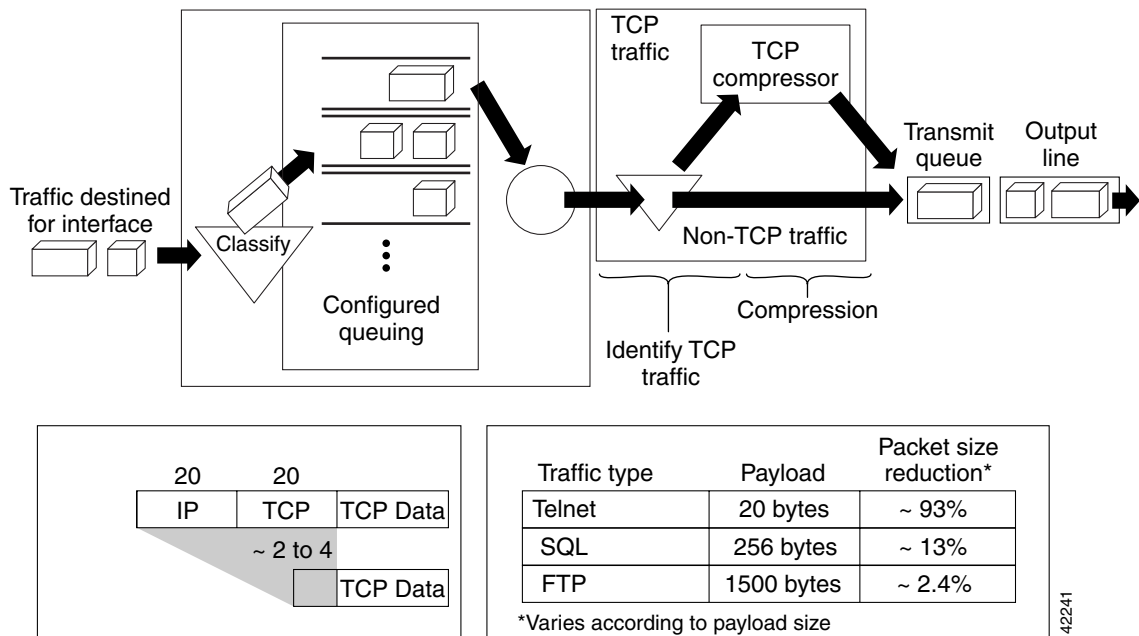
To avoid the unnecessary consumption of available bandwidth, TCP header compression is used on a link-by-link basis.

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 39](#) 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 39 TCP Header Compression



142241

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 39](#), 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 39](#), 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 “Modular Quality of Service Command-Line Interface” section of the [Cisco IOS Quality of Service Solutions Configuration Guide](#), Release 12.4.

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”).

Where to Go Next

Where you go next depends on the type of header compression that you want to configure.

- 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.

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, Release 12.4
MQC	Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.4
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

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	<i>Compressing TCP/IP Headers for Low-Speed Serial Links</i>
RFC 2507	<i>IP Header Compression</i>
RFC 2508	<i>Compressing IP/UDP/RTP Headers for Low-Speed Serial Links</i>
RFC 3544	<i>IP Header Compression over PPP</i>
RFC 3545	<i>Enhanced Compressed RTP (CRTP) for Links with High Delay, Packet Loss and Reordering</i>
RFC 3550	<i>A Transport Protocol for Real-Time Applications</i>

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.

**Note**

See [Internetworking Terms and Acronyms](#) for terms not included in this glossary.



Configuring RTP Header Compression

First Published: January 30, 2006

Last Updated: January 30, 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. 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 751.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- [Prerequisites for Configuring RTP Header Compression, page 732](#)
- [Information About Configuring RTP Header Compression, page 732](#)
- [How to Configure RTP Header Compression, page 735](#)
- [Configuration Examples for RTP Header Compression, page 745](#)
- [Additional References, page 748](#)
- [Glossary, page 750](#)
- [Feature Information for Configuring RTP Header Compression, page 751](#)

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, page 732](#)
- [RTP Header-Compression Keywords, page 732](#)
- [Enhanced RTP Header Compression, page 734](#)
- [RTP Header Compression over Satellite Links, page 734](#)

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](#), Release 12.4.

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*, Release 12.4.

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](#), Release 12.4.

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](#), Release 12.4.

How to Configure RTP Header Compression

This section contains the following tasks:

- [Enabling RTP Header Compression on an Interface, page 735](#) (required)
- [Enabling RTP Header Compression on an Interface That Uses Frame Relay Encapsulation, page 736](#) (optional)
- [Enabling Enhanced RTP Header Compression, page 738](#) (optional)
- [Enabling RTP Header Compression over a Satellite Link, page 740](#) (optional)
- [Specifying the Header-Compression Settings, page 741](#) (optional)
- [Changing the Number of Header-Compression Connections, page 742](#) (optional)
- [Displaying Header-Compression Statistics, page 744](#) (optional)

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 736 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. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> [<i>name-tag</i>] Example: Router(config)# interface serial0	Configures an interface type and enters interface configuration mode. <ul style="list-style-type: none"> Enter the interface type and the interface number.
Step 4	encapsulation <i>encapsulation-type</i> Example: Router(config-if)# encapsulation ppp	Sets the encapsulation method used by the interface. <ul style="list-style-type: none"> Enter the encapsulation method.
Step 5	ip address <i>ip-address mask</i> [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. <ul style="list-style-type: none"> 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	exit Example: Router(config-if)# exit	(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. **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 <i>type number</i> [<i>name-tag</i>] 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.

	Command or Action	Purpose
Step 5	<pre>ip address ip-address mask [secondary]</pre> <p>Example: Router(config-if)# ip address 209.165.200.225 255.255.255.224 </p>	Sets a primary or secondary IP address for an interface. <ul style="list-style-type: none"> Enter the IP address and mask for the associated IP subnet.
Step 6	<pre>frame-relay interface-dlci dlci [cisco]</pre> <p>Example: Router(config-if)# frame-relay interface-dlci 20 </p>	Assigns a data-link connection identifier (DLCI) to a specified Frame Relay interface on the router.
Step 7	<pre>frame-relay ip rtp header-compression [active passive] [periodic-refresh]</pre> <p>Example: Router(config-if)# frame-relay ip rtp header-compression</p> <p>or</p> <pre>frame-relay map ip ip-address dlci [broadcast] rtp header-compression [active passive] [periodic-refresh] [connections number]</pre> <p>Example: Router(config-if)# frame-relay map ip 10.108.175.220 180 rtp header-compression periodic-refresh </p>	Enables RTP header compression for all Frame Relay maps on a physical interface.
Step 8	<pre>exit</pre> <p>Example: Router(config-if)# exit </p>	(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 734](#).

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 735](#).

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. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> [<i>name-tag</i>] Example: Router(config)# interface serial0	Configures an interface type and enters interface configuration mode. <ul style="list-style-type: none"> • Enter the interface type and the interface number.
Step 4	encapsulation <i>encapsulation-type</i> Example: Router(config-if)# encapsulation ppp	Sets the encapsulation method used on the interface. <ul style="list-style-type: none"> • Enter the encapsulation method.
Step 5	ip address <i>ip-address mask</i> [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. <ul style="list-style-type: none"> • 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.

	Command or Action	Purpose
Step 7	<pre>ip header-compression recoverable-loss {dynamic packet-drops}</pre> <p>Example: Router(config-if)# ip header-compression recoverable-loss dynamic </p>	<p>Enables ECRTTP on an interface.</p> <p>Note Enter the dynamic keyword to enable dynamic packet loss recovery, or enter the <i>packet-drops</i> argument to specify the maximum number of consecutive packet drops that are acceptable.</p>
Step 8	<pre>exit</pre> <p>Example: Router(config-if)# exit </p>	(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. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	<pre>enable</pre> <p>Example: Router> enable </p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<pre>configure terminal</pre> <p>Example: Router# configure terminal </p>	Enters global configuration mode.
Step 3	<pre>interface type number [name-tag]</pre> <p>Example: Router(config)# interface serial0 </p>	<p>Configures an interface type and enters interface configuration mode.</p> <ul style="list-style-type: none"> • Enter the interface type and the interface number.

	Command or Action	Purpose
Step 4	<p>ip address <i>ip-address mask</i> [secondary]</p> <p>Example: Router(config-if)# ip address 209.165.200.225 255.255.255.224</p>	<p>Sets a primary or secondary IP address for an interface.</p> <ul style="list-style-type: none"> Enter the IP address and mask for the associated IP subnet.
Step 5	<p>ip rtp header-compression [passive iphc-format ietf-format] [periodic-refresh]</p> <p>Example: Router(config-if)# ip rtp header-compression ietf-format periodic-refresh</p>	<p>Enables RTP header compression.</p> <p>Note For RTP header compression over a satellite link, use the periodic-refresh keyword.</p>
Step 6	<p>ip header-compression disable-feedback</p> <p>Example: Router(config-if)# ip header-compression disable-feedback</p>	<p>(Optional) Disables the context status feedback messages from the interface or link.</p>
Step 7	<p>exit</p> <p>Example: Router(config-if)# exit</p>	<p>(Optional) Exits interface configuration mode.</p>

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

- enable**
- configure terminal**
- interface** *type number* [*name-tag*]
- 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*
- exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> [<i>name-tag</i>] Example: Router(config)# interface serial0	Configures an interface type and enters interface configuration mode. <ul style="list-style-type: none"> Enter the interface type and the interface number.
Step 4	ip header-compression max-header <i>max-header-size</i> Example: Router(config-if)# ip header-compression max-header 100	Specifies the maximum size of the compressed IP header. <ul style="list-style-type: none"> Enter the maximum size of the compressed IP header, in bytes.
	or	
	ip header-compression max-time <i>length-of-time</i> 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. <ul style="list-style-type: none"> Enter the amount of time, in seconds.
	or	
	ip header-compression max-period <i>number-of-packets</i> Example: Router(config-if)# ip header-compression max-period 160	Specifies the maximum number of compressed packets between full headers. <ul style="list-style-type: none"> Enter the maximum number of compressed packets between full headers.
Step 5	exit Example: Router(config-if)# exit	(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. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> [<i>name-tag</i>] Example: Router(config)# interface serial0	Configures an interface type and enters interface configuration mode. <ul style="list-style-type: none"> Enter the interface type and the interface number.
Step 4	ip rtp compression-connections <i>number</i> Example: Router(config-if)# ip rtp compression-connections 150	Specifies the total number of RTP header-compression connections that can exist on an interface. <ul style="list-style-type: none"> 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 <i>number</i> 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). <ul style="list-style-type: none"> Enter the number of compression connections. Note This command can be used for interfaces that use Frame Relay encapsulation <i>only</i> .
Step 5	exit Example: Router(config-if)# exit	(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

- enable**
- show ip rtp header-compression** [*interface-type interface-number*] [**detail**]

or

```
show frame-relay ip rtp header-compression [interface type number]
```

3. exit

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	show ip rtp header-compression [<i>interface-type</i> <i>interface-number</i>] [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 [<i>interface type number</i>] Example: Router# show frame-relay ip rtp header-compression	Displays Frame Relay RTP header-compression statistics for one or all interfaces.
Step 3	exit Example: Router# exit	(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, page 746](#)
- [Enabling RTP Header Compression on an Interface That Uses Frame Relay Encapsulation: Example, page 746](#)
- [Enabling Enhanced RTP Header Compression: Example, page 746](#)
- [Enabling RTP Header Compression over a Satellite Link: Example, page 747](#)
- [Specifying the Header-Compression Settings: Example, page 747](#)
- [Changing the Number of Header-Compression Connections: Example, page 747](#)
- [Displaying Header-Compression Statistics: Example, page 747](#)

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)# exit
```

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)# exit
```

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)# exit
```


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)# exit
```

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)# exit
```

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)# exit
```

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
```

Additional References

```
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, Release 12.4
Frame Relay	Cisco IOS Wide-Area Networking Configuration Guide, Release 12.4
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

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	<i>IP Header Compression</i>
RFC 2508	<i>Compressing IP/UDP/RTP Headers for Low-Speed Serial Links</i>

RFC	Title
RFC 3544	<i>IP Header Compression over PPP</i>
RFC 3545	<i>Enhanced Compressed RTP (CRTP) for Links with High Delay, Packet Loss and Reordering</i>

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.

**Note**

See *Internetworking Terms and Acronyms* for terms not included in this glossary.

Feature Information for Configuring RTP Header Compression

[Table 11](#) 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.

For information on a feature in this technology that is not documented here, see the “[Header-Compression Features Roadmap](#).”

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 Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

**Note**

[Table 11](#) 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 11 **Feature Information for Configuring RTP Header Compression**

Feature Name	Releases	Feature Information
RTP Header Compression over Satellite Links	12.3(2)T	<p>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.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • RTP Header Compression over Satellite Links, page 734 • Enabling RTP Header Compression over a Satellite Link, page 740
Enhanced CRTP for Links with High Delay, Packet Loss and Reordering	12.3(11)T	<p>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.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Enhanced RTP Header Compression, page 734 • Enabling Enhanced RTP Header Compression, page 738



Configuring TCP Header Compression

First Published: January 30, 2006

Last Updated: January 30, 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 TCP header compression.



Note

TCP header compression is configured on a per-interface (or subinterface) basis. If you want to configure TCP 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. 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 TCP Header Compression](#)” section on page 767.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- [Prerequisites for Configuring TCP Header Compression, page 754](#)
- [Information About Configuring TCP Header Compression, page 754](#)
- [How to Configure TCP Header Compression, page 755](#)
- [Configuration Examples for TCP Header Compression, page 763](#)
- [Additional References, page 764](#)
- [Glossary, page 766](#)
- [Feature Information for Configuring TCP Header Compression, page 767](#)

Prerequisites for Configuring TCP Header Compression

- Before configuring TCP header compression, read the information in the “Header Compression” module.
- You must configure TCP header compression on both ends of the network.

Information About Configuring TCP Header Compression

Before configuring TCP header compression, you should understand the following concepts:

- [TCP Header-Compression Keywords, page 754](#)
- [Maximum Compressed IP Header Size and TCP Header Compression, page 755](#)

TCP Header-Compression Keywords

When you configure TCP header compression, you can specify the circumstances under which the TCP 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 TCP header compression, such as the **ip tcp header-compression** command. For more information about the **ip tcp header-compression** command, these keywords, and the other QoS commands, see the [Cisco IOS Quality of Service Solutions Command Reference](#), Release 12.4.

The **passive** Keyword

By default, the **ip tcp header-compression** command compresses outgoing TCP traffic. If you specify the **passive** keyword, outgoing TCP traffic is compressed only if *incoming* TCP traffic on the *same* interface is compressed. If you do not specify the **passive** keyword, *all* outgoing TCP traffic is compressed.

The **passive** keyword is ignored for 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, RTP header compression is also enabled. Since both TCP and RTP header compression are enabled, both TCP and UDP packets are compressed.

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 TCP packets. For PPP interfaces, when the **ietf-format** keyword is specified, RTP header compression is also enabled. Since both TCP header compression and RTP header compression are enabled, both TCP packets and UDP packets are compressed.

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.

Maximum Compressed IP Header Size and TCP Header Compression

With TCP header compression, you can configure the maximum size of the compressed IP header by using the **ip header-compression max-header** command.

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. For more information about the **ip header-compression max-header** command, see the *Cisco IOS Quality of Service Solutions Command Reference*, Release 12.4.

How to Configure TCP Header Compression

This section contains the following tasks:

- [Enabling TCP Header Compression on an Interface, page 755](#) (required)
- [Enabling TCP Header Compression on an Interface That Uses Frame Relay Encapsulation, page 757](#) (optional)
- [Changing the Maximum Size of the Compressed IP Header, page 758](#) (optional)
- [Changing the Number of Header-Compression Connections, page 760](#) (optional)
- [Displaying Header-Compression Statistics, page 762](#) (optional)

Enabling TCP Header Compression on an Interface

To enable TCP header compression on an interface, perform the following steps.

**Note**

To enable TCP header compression on an interface that uses Frame Relay encapsulation, skip these steps and complete the steps in the “[Enabling TCP Header Compression on an Interface That Uses Frame Relay Encapsulation](#)” section on page 757 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 tcp header-compression** [**passive** | **iphc-format** | **ietf-format**]
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> [<i>name-tag</i>] Example: Router(config)# interface serial0	Configures an interface type and enters interface configuration mode. <ul style="list-style-type: none">• Enter the interface type and the interface number.
Step 4	encapsulation <i>encapsulation-type</i> Example: Router(config-if)# encapsulation ppp	Sets the encapsulation method used by the interface. <ul style="list-style-type: none">• Enter the encapsulation method.
Step 5	ip address <i>ip-address mask</i> [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. <ul style="list-style-type: none">• Enter the IP address and mask for the associated IP subnet.
Step 6	ip tcp header-compression [passive iphc-format ietf-format] Example: Router(config-if)# ip tcp header-compression ietf-format	Enables TCP header compression.
Step 7	exit Example: Router(config-if)# exit	(Optional) Exits interface configuration mode.

Enabling TCP Header Compression on an Interface That Uses Frame Relay Encapsulation

To enable TCP 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 TCP 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 tcp header-compression** [**passive**]
or
frame-relay map ip *ip-address dlci* [**broadcast**] **tcp header-compression** [**active** | **passive**] [**connections number**]
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> [<i>name-tag</i>] Example: Router(config)# interface serial0	Configures an interface type and enters interface configuration mode. <ul style="list-style-type: none"> • Enter the interface type and the interface number.

	Command or Action	Purpose
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. <ul style="list-style-type: none"> 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 or access server. <ul style="list-style-type: none"> Enter the DLCI number.
Step 7	frame-relay ip tcp header-compression [passive] Example: Router(config-if)# frame-relay ip tcp header-compression or frame-relay map ip ip-address dlci [broadcast] tcp header-compression [active passive] [connections number] Example: Router(config-if)# frame-relay map ip 10.108.175.200 190 tcp header-compression active	Enables TCP header compression for all Frame Relay maps on a physical interface. Assigns to an IP map header-compression characteristics that differ from the compression characteristics of the interface with which the IP map is associated. <ul style="list-style-type: none"> Enter the IP address, DLCI number, and any optional keywords and arguments.
Step 8	exit Example: Router(config-if)# exit	(Optional) Exits interface configuration mode.

Changing the Maximum Size of the Compressed IP Header

By default, the maximum size of the compressed IP header is 168 bytes. When you configure TCP header compression, you can change this size to suit the needs of your network.

To change the maximum size of the compressed IP header, perform the following steps.

SUMMARY STEPS

- enable**
- configure terminal**
- interface type number [name-tag]**
- ip header-compression max-header max-header-size**

5. exit

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>enable</p> <p>Example: Router> enable</p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	<p>configure terminal</p> <p>Example: Router# configure terminal</p>	<p>Enters global configuration mode.</p>
Step 3	<p>interface <i>type number [name-tag]</i></p> <p>Example: Router(config)# interface serial0</p>	<p>Configures an interface type and enters interface configuration mode.</p> <ul style="list-style-type: none"> Enter the interface type and the interface number.
Step 4	<p>ip header-compression max-header <i>max-header-size</i></p> <p>Example: Router(config-if)# ip header-compression max-header 100</p>	<p>Specifies the maximum size of the compressed IP header.</p> <ul style="list-style-type: none"> Enter the maximum size of the compressed IP header, in bytes.
Step 5	<p>exit</p> <p>Example: Router(config-if)# exit</p>	<p>(Optional) Exits interface configuration mode.</p>

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 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 tcp compression-connections** *number*
or
frame-relay ip tcp compression-connections *number*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>enable</p> <p>Example: Router> enable</p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	<p>configure terminal</p> <p>Example: Router# configure terminal</p>	<p>Enters global configuration mode.</p>
Step 3	<p>interface <i>type number [name-tag]</i></p> <p>Example: Router(config)# interface serial0</p>	<p>Configures an interface type and enters interface configuration mode.</p> <ul style="list-style-type: none"> Enter the interface type and the interface number.
Step 4	<p>ip tcp compression-connections <i>number</i></p> <p>Example: Router(config-if)# ip tcp compression-connections 150</p>	<p>Specifies the total number of TCP header compression connections that can exist on an interface.</p> <ul style="list-style-type: none"> Enter the number of compression connections. <p>Note This command can be used for PPP interfaces, HDLC interfaces, or interfaces that use Frame Relay encapsulation.</p>
	<p>or</p> <p>frame-relay ip tcp compression-connections <i>number</i></p> <p>Example: Router(config-if)# frame-relay ip tcp compression-connections 150</p>	<p>Specifies the maximum number of TCP header compression connections that can exist on an interface that use Frame Relay encapsulation.</p> <ul style="list-style-type: none"> Enter the number of compression connections. <p>Note This command can be used for interfaces that use Frame Relay encapsulation <i>only</i>.</p>
Step 5	<p>exit</p> <p>Example: Router(config-if)# exit</p>	<p>(Optional) Exits interface configuration mode.</p>

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 tcp header-compression** command or the **show frame-relay ip tcp header-compression** command.

To display header-compression statistics, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show ip tcp header-compression** [*interface-type interface-number*] [**detail**]
or
show frame-relay ip tcp header-compression [**interface** *type number*]
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	show ip tcp header-compression [<i>interface-type interface-number</i>] [detail] Example: Router# show ip tcp header-compression	Displays TCP/IP header compression statistics.
	or show frame-relay ip tcp header-compression [interface <i>type number</i>] Example: Router# show frame-relay ip tcp header-compression	Displays Frame Relay TCP/IP header compression statistics for one or all interfaces.
Step 3	exit Example: Router# exit	(Optional) Exits privileged EXEC mode.

Configuration Examples for TCP Header Compression

This section provides the following configuration examples:

- [Enabling TCP Header Compression on an Interface: Example, page 763](#)
- [Enabling TCP Header Compression on an Interface That Uses Frame Relay Encapsulation: Example, page 763](#)
- [Changing the Maximum Size of the Compressed IP Header: Example, page 763](#)
- [Changing the Number of Header-Compression Connections: Example, page 764](#)
- [Displaying Header-Compression Statistics: Example, page 764](#)

Enabling TCP Header Compression on an Interface: Example

In the following example, TCP 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 tcp header-compression ietf-format
Router(config-if)# exit
```

Enabling TCP Header Compression on an Interface That Uses Frame Relay Encapsulation: Example

In the following example, TCP 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 tcp header-compression
Router(config-if)# exit
```

Changing the Maximum Size of the Compressed IP Header: 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)# exit
```

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 tcp compression-connections** command.

```
Router> enable
Router# configure terminal
Router(config)# interface serial0
Router(config-if)# ip tcp compression-connections 150
Router(config-if)# exit
```

Displaying Header-Compression Statistics: Example

You can use the **show ip tcp 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 tcp header-compression** command:

```
Router# show ip tcp header-compression serial0

TCP/IP header compression statistics:
Interface Serial0 (compression on, IETF)
  Rcvd:   53797 total, 53796 compressed, 0 errors, 0 status msgs
         0 dropped, 0 buffer copies, 0 buffer failures
  Sent:   53797 total, 53796 compressed, 0 status msgs, 0 not predicted
         1721848 bytes saved, 430032 bytes sent
         5.00 efficiency improvement factor
  Connect: 16 rx slots, 16 tx slots,
          1 misses, 0 collisions, 0 negative cache hits, 15 free contexts
          99% hit ratio, five minute miss rate 0 misses/sec, 0 max
```

Additional References

The following sections provide references related to configuring TCP 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 , Release 12.4
Frame Relay	Cisco IOS Wide-Area Networking Configuration Guide , Release 12.4
Header compression overview	“Header Compression” module
RTP header compression	“Configuring RTP Header Compression” module
Class-based RTP and TCP header compression	“Configuring Class-Based RTP and TCP Header Compression” 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	<i>Compressing TCP/IP Headers for Low-Speed Serial Links</i>
RFC 2507	<i>IP Header Compression</i>
RFC 3544	<i>IP Header Compression over PPP</i>

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.

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 uniquely specify individual end devices).

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 context identifier (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.

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.

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.

**Note**

See [Internetworking Terms and Acronyms](#) for terms not included in this glossary.

Feature Information for Configuring TCP Header Compression

Table 12 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.

For information on a feature in this technology that is not documented here, see the “[Header-Compression Features Roadmap](#).”

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 Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.


Note

Table 12 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 12 *Feature Information for Configuring TCP Header Compression*

Feature Name	Releases	Feature Information
This table is intentionally left blank because no features were introduced or modified in Cisco IOS Release 12.2(1) or a later release. This table will be updated when feature information is added to this module.	—	—



Configuring Class-Based RTP and TCP Header Compression

First Published: January 30, 2006
Last Updated: January 30, 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) packets or Transmission Control Protocol (TCP) packets.

Cisco provides two types of header compression: RTP header compression and TCP header compression.

RTP and TCP header compression are typically configured on a per-interface (or subinterface) basis. Class-based RTP and TCP header compression allows you to configure either type of header compression on a per-class basis. This module describes the concepts and tasks related to configuring class-based RTP and TCP header compression.



Note

If you want to configure RTP or TCP header compression on a per-interface (or subinterface) basis, see the [“Configuring RTP Header Compression”](#) module or the [“Configuring TCP Header Compression”](#) module, respectively.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. 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 RTP and TCP Header Compression”](#) section on page 781.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- [Prerequisites for Class-Based RTP and TCP Header Compression, page 770](#)
- [Restrictions for Class-Based RTP and TCP Header Compression, page 770](#)

- [Information About Class-Based RTP and TCP Header Compression, page 770](#)
- [How to Configure Class-Based RTP and TCP Header Compression, page 772](#)
- [Configuration Examples for Class-Based RTP and TCP Header Compression, page 776](#)
- [Additional References, page 779](#)
- [Glossary, page 781](#)
- [Feature Information for Class-Based RTP and TCP Header Compression, page 781](#)

Prerequisites for Class-Based RTP and TCP Header Compression

Before configuring class-based RTP and TCP header compression, read the information in the “[Header Compression](#)” module.

Restrictions for Class-Based RTP and TCP Header Compression

Class-based RTP and TCP header compression can be enabled on PPP interfaces, High-Level Data Link Control (HDLC) interfaces, and interfaces that use Frame Relay encapsulation. However, note the following points about the header-compression formats supported on these interfaces:

- For PPP and HDLC interfaces, the only supported format for header compression is the IPHC (IP Header Compression) format.
- For interfaces that use Frame Relay encapsulation, the IPHC format is not available. The only supported format for header compression is the Cisco proprietary format.

Information About Class-Based RTP and TCP Header Compression

Before configuring class-based RTP and TCP header compression, you should understand the following concepts:

- [Class-Based Header Compression and the MQC, page 770](#)
- [Benefits of Class-Based Header Compression, page 771](#)
- [Header Compression on Local and Remote Routers, page 771](#)
- [About Header-Compression Connections, page 771](#)

Class-Based Header Compression and the MQC

Class-based RTP and TCP header compression allows you to configure *either* RTP *or* TCP header compression for a specific class within a policy map (sometimes referred to as a traffic policy). You configure the class and the policy map by using the Modular Quality of Service (QoS) Command-Line Interface (CLI) (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 (or subinterfaces). The policy maps are used to

configure and apply specific QoS features (such as RTP or TCP header compression) to your network. For more information about the MQC, see the [Cisco IOS Quality of Service Solutions Configuration Guide](#), Release 12.4.

Benefits of 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. For example, instead of compressing all RTP (or TCP) packets that traverse 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”).

Header Compression on Local and Remote Routers

In a typical network topology, header compression is configured at both a local router and a remote router. If you configure class-based RTP header compression (or class-based TCP header compression) on the local router, you must also configure RTP header compression (or TCP header compression) on the remote router.

However, when you configure either RTP or TCP header compression on the remote router, you can choose one of the following:

- You can configure *class-based* RTP or TCP header compression on the remote router (by using the instructions in this module)
- or
- You can configure RTP or TCP header compression *directly on the interface* of the remote router (by using the instructions in the “[Configuring RTP Header Compression](#)” module or the “[Configuring TCP Header Compression](#)” module, respectively).



Note If you configure RTP or TCP header compression directly on the interface of the remote router, you must specify the **iphc-format** keyword for PPP and HDLC interfaces. For Frame Relay interfaces, the **iphc-format** keyword is not supported; only the Cisco proprietary format (that is, the **cisco** keyword) is supported.

For more information about the **iphc-format** keyword, see either the “[Configuring RTP Header Compression](#)” module or the “[Configuring TCP Header Compression](#)” module.

About Header-Compression Connections

Number of Connections Calculated on the Basis of Bandwidth

In class-based RTP and TCP header compression, the number of header-compression connections is calculated on the basis of the amount of available bandwidth.

Note the following points about how bandwidth is used:

- The setting of the **bandwidth** command determines the amount of bandwidth available on the interface.
- The number of header-compression connections is calculated by dividing the available bandwidth by 4 (that is, 4 kilobits per connection).

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 calculated (from the bandwidth setting) for use on the local router must match the number configured (or calculated from the bandwidth setting) 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.

How to Configure Class-Based RTP and TCP Header Compression

This section contains the following tasks:

- [Enabling RTP or TCP Header Compression for a Class in a Policy Map, page 772](#) (required)
- [Attaching the Policy Map to an Interface, page 774](#) (required)
- [Verifying the Class-Based RTP and TCP Header Compression Configuration, page 775](#) (optional)

Enabling RTP or TCP Header Compression for a Class in a Policy Map

With class-based header compression, you can configure either RTP or TCP header compression for a specific class inside a policy map. To specify the class, to create a policy map, and to configure either RTP or TCP header compression for the class inside the policy map, perform the following steps.



Note In the following task, the **match protocol** command is shown in step 4. The **match protocol** command matches traffic on the basis on the protocol type and is only an example of a **match** command you can use. You may want to use a different **match** command to specify another criterion. 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.

SUMMARY STEPS

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

5. **exit**
6. **policy-map** *policy-map-name*
7. **class** {*class-name* | **class-default**}
8. **compression header ip** {**rtp** | **tcp**}
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> 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] <i>class-map-name</i> Example: Router(config)# class-map class1	Creates a class map to be used for matching packets to a specified class and enters class-map configuration mode. <ul style="list-style-type: none"> Enter the class map name.
Step 4	match protocol <i>protocol-name</i> Example: Router(config-cmap)# match protocol ip	(Optional) Matches traffic on the basis of the specified protocol. <ul style="list-style-type: none"> Enter the protocol name. <p>Note The match protocol command matches traffic on the basis of the protocol type. The match protocol 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.</p>
Step 5	exit Example: Router(config-cmap)# exit	(Optional) Exits class-map configuration mode.
Step 6	policy-map <i>policy-map-name</i> Example: Router(config)# policy-map policy1	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy and enters policy-map configuration mode. <ul style="list-style-type: none"> Enter the policy map name.
Step 7	class { <i>class-name</i> class-default }	Specifies the name of the class whose policy you want to create or change and enters policy-map class configuration mode. <ul style="list-style-type: none"> Enter the class name or the class-default keyword.

	Command or Action	Purpose
Step 8	<code>compression header ip {rtp tcp}</code> Example: Router(config-pmap-c)# compression header ip rtp	Configures either RTP or TCP header compression for a specific class. • Enter either the rtp keyword (for RTP header compression) or the tcp keyword (for TCP header compression).
Step 9	<code>exit</code> Example: Router(config-pmap-c)# exit	(Optional) Exits policy-map class configuration mode.

Attaching the Policy Map to an Interface

After a policy map is created, the next step is to attach the policy map to an interface (or subinterface). To attach the policy map to an interface or subinterface, perform the following steps.

Restrictions

You configure class-based RTP and TCP header compression in policy maps. Then you attach those policy maps to an interface by using the **service-policy** command. The **service-policy** command gives you the option of specifying either an input service policy (for input interfaces) or an output service policy (for output interfaces). For class-based RTP and TCP header compression, you can specify output service policies *only*.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **service-policy output** *policy-map-name*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>enable</code> Example: Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	<code>configure terminal</code> Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	interface <i>type number</i> [<i>name-tag</i>] Example: Router(config)# interface serial0	Configures an interface type and enters interface configuration mode. <ul style="list-style-type: none"> Enter the interface type and the interface number.
Step 4	service-policy output <i>policy-map-name</i> Example: Router(config-if)# service-policy output policy1	Specifies the name of the policy map to be attached to the interface in the output direction. <ul style="list-style-type: none"> Enter the policy map name. Note Policy maps can be attached in the input or output direction of an interface. For class-based RTP and TCP header compression, always use the output keyword.
Step 5	exit Example: Router(config-if)# exit	(Optional) Exits interface configuration mode.

Verifying the Class-Based RTP and TCP Header Compression Configuration

This task allows you to verify that you created the intended configuration and that the feature is functioning correctly. To verify the configuration, perform the following steps.

SUMMARY STEPS

- enable**
- show policy-map interface** *type number*
or
show policy-map *policy-map class class-name*
- exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	show policy-map interface <i>type number</i> output Example: Router# show policy-map interface serial0 output	Displays the packet statistics of all classes that are configured for all service policies on the specified interface. <ul style="list-style-type: none">Enter the interface type and the interface number.
	or	
	show policy-map <i>policy-map</i> class <i>class-name</i> Example: Router# show policy-map policy1 class class1	Displays the configuration for the specified class of the specified policy map. <ul style="list-style-type: none">Enter the policy map name and the class name.
Step 3	exit Example: Router# exit	(Optional) Exits privileged EXEC mode.

Configuration Examples for Class-Based RTP and TCP Header Compression

This section provides the following configuration examples:

- [Enabling RTP or TCP Header Compression for a Class in a Policy Map: Example, page 776](#)
- [Attaching the Policy Map to an Interface: Example, page 777](#)
- [Verifying the Class-Based RTP and TCP Header Compression Configuration: Example, page 777](#)

Enabling RTP or TCP Header Compression for a Class in a Policy Map: Example

In the following example, a class map called class1 and a policy map called policy1 have been configured. Policy1 contains the class called class1, within which RTP header compression has been enabled by using the **compression header ip rtp** command.

```
Router> enable
Router# configure terminal
Router(config)# class-map class1
Router(config-cmap)# match protocol ip
Router(config-cmap)# exit
Router(config)# policy-map policy1
Router(config-pmap)# class class1
Router(config-pmap-c)# compression header ip rtp
Router(config-pmap-c)# exit
```

Attaching the Policy Map to an Interface: Example

In the following example, the policy map called policy1 has been attached to serial interface 0.

```
Router> enable
Router# configure terminal
Router(config)# interface serial0
Router(config-if)# service-policy output policy1
Router(config-if)# exit
```

Verifying the Class-Based RTP and TCP Header Compression Configuration: Example

This section provides sample output from a typical **show policy-map interface** command.



Note

Depending upon the interface in use and the QoS feature enabled (such as Class-Based Weighted Fair Queuing [CBWFQ]), the output you see may vary from that shown below.

The following sample displays the statistics for serial interface 0. In this sample configuration, three classes, called gold, silver, and voice, have been configured. Traffic is classified and grouped into classes on the basis of the IP precedence value and RTP port protocol number.

```
class-map match-all gold
  match ip precedence 2
class-map match-all silver
  match ip precedence 1
class-map match-all voice
  match ip precedence 5
  match ip rtp 16384 1000
```

This sample configuration also contains a policy map called mypolicy, configured as shown below. QoS features such as RTP header compression and CBWFQ are enabled for specific classes within the policy map.

```
policy-map mypolicy
  class voice
    priority 128                ! A priority queue and bandwidth amount are specified.
    compress header ip rtp      ! RTP header compression is enabled for class voice.
  class gold
    bandwidth 100              ! CBWFQ is enabled for class gold.
  class silver
    bandwidth 80               ! CBWFQ is enabled for class silver.
    random-detect              ! WRED is enabled for class silver.
```

Given the classes and policy map configured as shown above, the following content is displayed for serial interface 0:

```
Router# show policy-map interface serial0 output

Serial0

Service-policy output: mypolicy

Class-map: voice (match-all)
  880 packets, 58080 bytes
  30 second offered rate 1000 bps, drop rate 0 bps
  Match: ip precedence 5
```

```

Match: ip rtp 16384 1000
Queueing
  Strict Priority
  Output Queue: Conversation 136
  Bandwidth 128 (kbps) Burst 3200 (Bytes)
  (pkts matched/bytes matched) 880/26510
  (total drops/bytes drops) 0/0
compress:
  header ip rtp
  UDP/RTP (compression on, IPHC, RTP)
  Sent:      880 total, 877 compressed,
            31570 bytes saved, 24750 bytes sent
            2.27 efficiency improvement factor
            99% hit ratio, five minute miss rate 0 misses/sec, 0 max
            rate 0 bps

```

```

Class-map: gold (match-all)
  100 packets, 53000 bytes
  30 second offered rate 0 bps, drop rate 0 bps
Match: ip precedence 2
Queueing
  Output Queue: Conversation 137
  Bandwidth 100 (kbps) Max Threshold 64 (packets)
  (pkts matched/bytes matched) 100/53000
  (depth/total drops/no-buffer drops) 0/0/0

```

```

Class-map: silver (match-all)
  878 packets, 1255540 bytes
  30 second offered rate 56000 bps, drop rate 0 bps
Match: ip precedence 1
Queueing
  Output Queue: Conversation 138
  Bandwidth 64 (kbps)
  (pkts matched/bytes matched) 878/1255540
  (depth/total drops/no-buffer drops) 0/0/0
  exponential weight: 9
  mean queue depth: 0

```

class	Transmitted pkts/bytes	Random drop pkts/bytes	Tail drop pkts/bytes	Minimum thresh	Maximum thresh	Mark prob
0	0/0	0/0	0/0	20	40	1/10
1	878/1255540	0/0	0/0	22	40	1/10
2	0/0	0/0	0/0	24	40	1/10
3	0/0	0/0	0/0	26	40	1/10
4	0/0	0/0	0/0	28	40	1/10
5	0/0	0/0	0/0	30	40	1/10
6	0/0	0/0	0/0	32	40	1/10
7	0/0	0/0	0/0	34	40	1/10
rsvp	0/0	0/0	0/0	36	40	1/10

```

Class-map: class-default (match-any)
  3 packets, 84 bytes
  30 second offered rate 0 bps, drop rate 0 bps
Match: any

```


Additional References

The following sections provide references related to configuring class-based RTP and TCP header compression.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	<i>Cisco IOS Quality of Service Solutions Command Reference</i> , Release 12.4
QoS features	<i>Cisco IOS Quality of Service Solutions Configuration Guide</i> , Release 12.4
MQC	<i>Cisco IOS Quality of Service Solutions Configuration Guide</i> , Release 12.4
Header compression overview	“Header Compression” module
RTP header compression	“Configuring RTP Header Compression” module
TCP header compression	“Configuring TCP Header Compression” 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	<i>Compressing TCP/IP Headers for Low-Speed Serial Links</i>
RFC 2507	<i>IP Header Compression</i>
RFC 2508	<i>Compressing IP/UDP/RTP Headers for Low-Speed Serial Links</i>
RFC 3544	<i>IP Header Compression over PPP</i>

■ Additional References

RFC	Title
RFC 3545	<i>Enhanced Compressed RTP (CRTP) for Links with High Delay, Packet Loss and Reordering</i>
RFC 3550	<i>A Transport Protocol for Real-Time Applications</i>

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

bandwidth—The rated throughput capacity of a given network medium.

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.

full header (header refresh)—An uncompressed header that updates or refreshes the context for a packet stream. It carries a context identifier (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.

MQC—Modular Quality of Service Command-Line Interface. The MQC is a CLI that 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.

**Note**

See [Internetworking Terms and Acronyms](#) for terms not included in this glossary.

Feature Information for Class-Based RTP and TCP Header Compression

[Table 13](#) 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 “[Header-Compression Features Roadmap](#).”

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

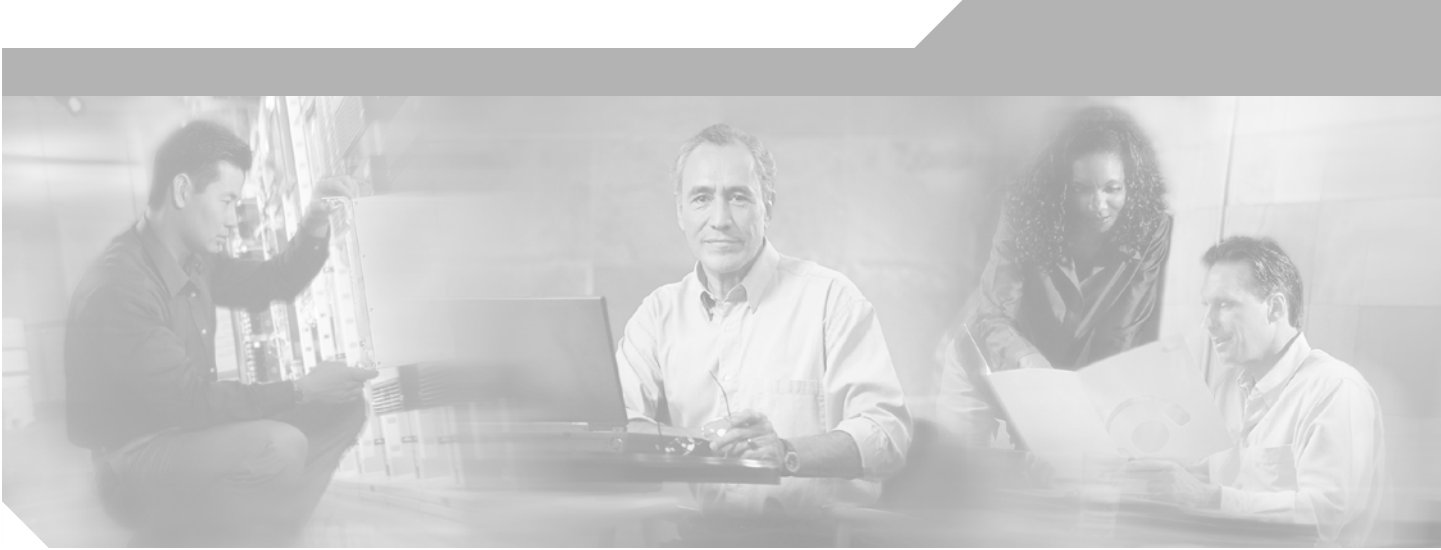
Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

**Note**

Table 13 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 13 Feature Information for Class-Based RTP and TCP Header Compression

Feature Name	Releases	Feature Information
Class-Based RTP and TCP Header Compression	12.2(13)T	<p>This feature allows you to configure Real-Time Transport Protocol (RTP) or Transmission Control Protocol (TCP) IP 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).</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Information About Class-Based RTP and TCP Header Compression, page 770 • How to Configure Class-Based RTP and TCP Header Compression, page 772



Part 7: Quality of Service Solutions





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 chapter in this book.

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

IP to ATM CoS supports configuration of the following features:

- Single ATM VCs
- VC bundles
- Per-VC Low Latency Queueing (LLQ), WFQ, and CBWFQ

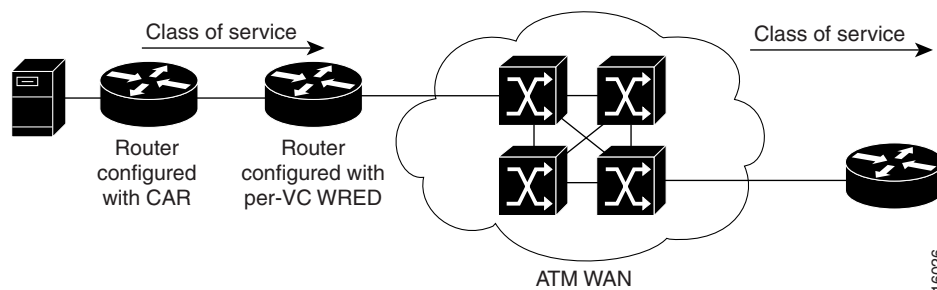
Single ATM VC Support

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 40 illustrates the IP to ATM CoS support for a single ATM VC.

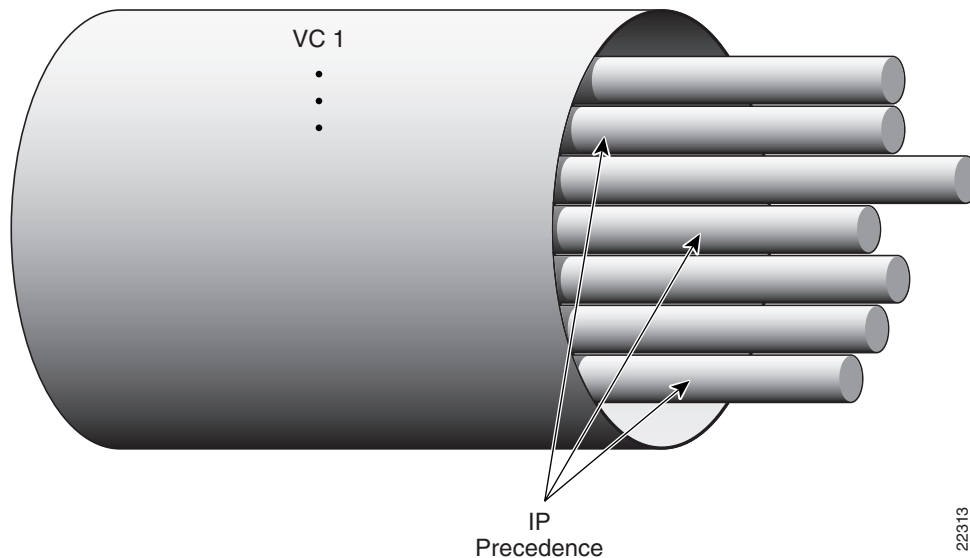
Figure 40 *Single ATM Circuit Class*



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 41, these VCs are grouped in a bundle and are referred to as bundle members.

Figure 41 ATM VC Bundle

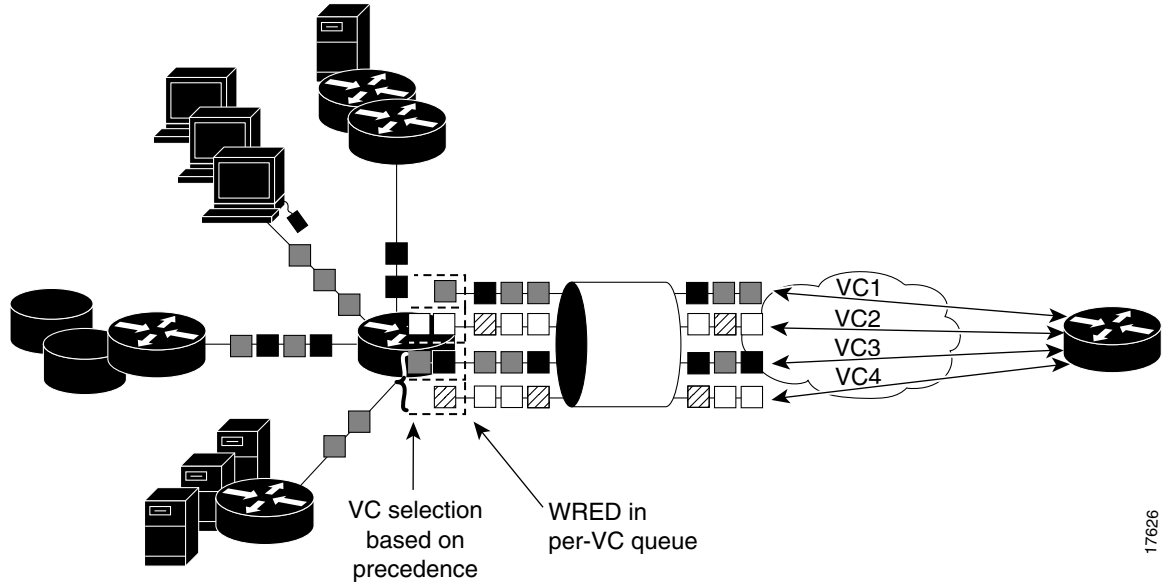


ATM VC bundle management allows you to define an ATM VC bundle and add VCs to it. Each VC of a bundle has its own ATM traffic class and ATM traffic parameters. You can apply attributes and characteristics to discrete VC bundle members or you can apply them collectively at the bundle level.

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 42](#)). 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 42](#) 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 42 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 chapter [Congestion Management Overview](#) in this book.

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 a VC to Use Flow-Based WFQ” section on page 803](#) in the “” chapter in this book.

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.

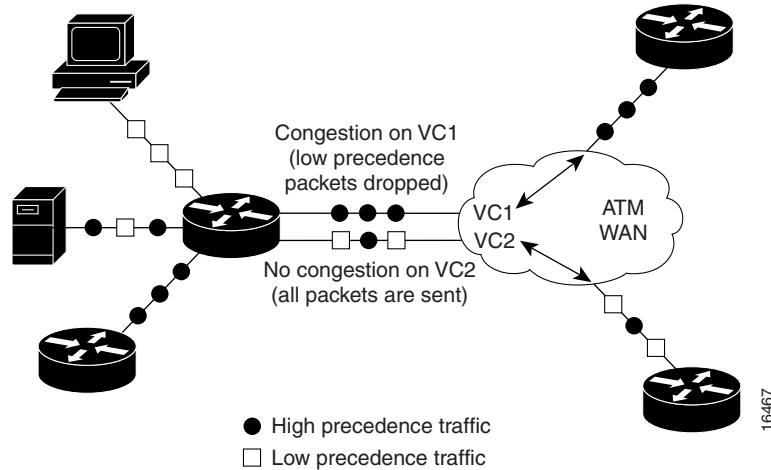
The use of per-VC queues ensures that a direct relationship exists between the outgoing ATM VC and the IP packets to be forwarded on that queue. This mechanism establishes a packet queue for each outgoing ATM VC. In this manner, should an ATM VC become congested, only the packet queue associated with 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 43](#) 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 43 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.

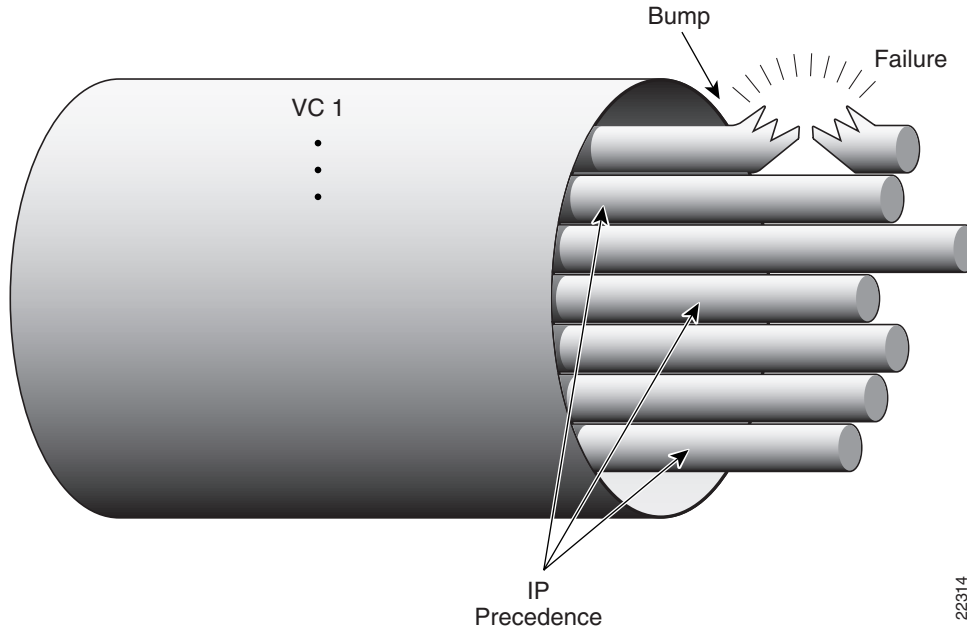
Bumping and ATM VC Bundles

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 44](#) conceptualizes a failed VC bundle member whose failure calls into effect the configured bumping rules.

Figure 44 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).



IP to ATM Class of Service

This part consists of the following:

- [Configuring IP to ATM Class of Service](#)
- [IP to ATM Class of Service Mapping for SVC Bundles](#)
- [ATM PVC Bundle Enhancement — MPLS EXP-Based PVC Selection](#)





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 chapter [IP to ATM Class of Service Overview](#) in this book.

For a complete description of the IP to ATM CoS commands in this chapter, refer to 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.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the [“Finding Additional Feature Support Information”](#) section on page lxix in the [Using Cisco IOS Software for Release 12.4](#) chapter in this book.

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 Queuing 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:

Command	Purpose
Router(config)# random-detect-group <i>group-name</i>	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 <i>group-name</i>	Specifies the WRED or DWRED parameter group.
Step 2	Router(config)# exponential-weighting-constant <i>exponent</i>	Configures the exponential weight factor for the average queue size calculation for the specified WRED or DWRED parameter group.
	or	
	Router(config)# precedence <i>precedence min-threshold max-threshold mark-probability-denominator</i>	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 [<i>interface atm_subinterface</i> [<i>vc</i> [[<i>vpi</i> /] <i>vci</i>]]]	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 <i>interface-number</i> [<i>vc</i> [[<i>vpi</i> /] <i>vci</i>]]	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 <i>bundle-name</i>	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 <i>protocol</i> [<i>protocol-address</i> 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 <i>aal-encap</i>	Configures the ATM adaptation layer (AAL) and encapsulation type for the bundle.
Router(config-atm-bundle)# inarp <i>minutes</i>	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 <i>up-count down-count retry frequency</i>	Configures the VC bundle parameters related to operation, administration, and maintenance (OAM) management.
Router(config-atm-bundle)# oam-bundle [manage] [<i>frequency</i>]	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] [<i>frequency</i>]	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 on these commands, including configuration tasks and command syntax, refer to the *Cisco IOS Wide-Area Networking Configuration Guide* and 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 <i>vc-class-name</i>	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 <i>pvc-name</i> [<i>vpi</i>] [<i>vci</i>]	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
<code>Router(config-if-atm-member)# ubr output-pcr [input-pcr]</code>	Configures the VC for unspecified bit rate (UBR) QoS and specifies the output peak cell rate (PCR) for it.
<code>Router(config-if-atm-member)# ubr+ output-pcr output-mcr [input-pcr] [input-mcr]</code>	Configures the VC for UBR QoS and specifies the output PCR and output minimum guaranteed cell rate for it.
<code>Router(config-if-atm-member)# vbr-nrt output-pcr output-scr output-mbs [input-pcr] [input-scr] [input-mbs]</code>	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.
<code>Router(config-if-atm-member)# precedence [other range]</code>	Configures the precedence levels for the VC.
<code>Router(config-if-atm-member)# bump {implicit explicit precedence-level traffic}</code>	Configures the bumping rules for the VC.
<code>Router(config-if-atm-member)# protect {group vc}</code>	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
<code>Router(config-vc-class)# bump {implicit explicit precedence-level traffic}</code>	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.
<code>Router(config-vc-class)# precedence precedence min-threshold max-threshold mark-probability-denominator</code>	Defines precedence levels for the VC member to which the class is applied.
<code>Router(config-vc-class)# protect {group vc}</code>	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 “[Bumping and ATM VC Bundles](#)” section on page 791 in the [IP to ATM Class of Service Overview](#) chapter 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
<code>Router(config-if-atm-member)# class-vc vc-class -name</code>	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
<code>Router(config-if-atm-member)# no bump traffic</code>	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 <i>bundle-name</i>	Displays the bundle attributes assigned to each bundle VC member and the current working status of the VC members.
Router# show atm bundle <i>bundle-name</i> 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 [“Bandwidth Management” section on page 224](#) in the [Congestion Management Overview](#) chapter in this book.

For information on how to configure CBWFQ and perform the tasks mentioned, see the chapter 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 queuing 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
<code>Router(config-if-atm-vc)# service-policy output <i>policy-map</i></code>	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
<code>Router(config-if-atm-member)# service-policy output <i>policy-map</i></code>	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 <i>policy-map</i>	Specifies the name of the policy map to be created or modified.
Step 2	Router(config-pmap)# class class-default <i>default-class-name</i>	Specifies the default class so that you can configure or modify its policy.
Step 3	Router(config-pmap-c)# fair-queue <i>number-of-dynamic-queues</i>	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 <i>number-of-packets</i>	Specifies the maximum number of packets that can be queued for the class.
	or	
	Router(config-pmap-c)# random-detect	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 chapter 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 <i>policy-map</i>	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 <i>policy-map</i>	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 <i>interface-name interface-number</i> [vc [<i>vpi/</i>] <i>vci</i>]	Displays the contents of packets inside a queue for a particular interface or VC.
Router# show queueing interface <i>interface-number</i> [vc [<i>vpi/</i>] <i>vci</i>]	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 <i>level</i>	Limits messages logged to the console based on severity.

For information on the **logging console** command, including configuration tasks and command syntax, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide* and 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 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 1000 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 isubr+.

```
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 isubr.

```
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 al/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 Mbps 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 Mbps, 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 Mbps) 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
    encaps 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
```




IP to ATM Class of Service Mapping for SVC Bundles

Feature History

Release	Modification
12.2(4) T	This feature was introduced.

This feature module describes the IP to ATM Class of Service Mapping for SVC Bundles feature for Cisco IOS Release 12.2(4)T and includes the following sections:

- [Feature Overview, page 811](#)
- [Supported Platforms, page 812](#)
- [Supported Standards, MIBs, and RFCs, page 813](#)
- [Prerequisites, page 813](#)
- [Configuration Tasks, page 813](#)
- [Monitoring IP to ATM Class of Service Mapping for SVC Bundles, page 815](#)
- [Configuration Examples, page 816](#)
- [Command Reference, page 818](#)

Feature Overview

The IP to ATM Class of Service Mapping for SVC Bundles feature supports multiple switched virtual circuits (SVCs) to the same NSAP destination for different types of service (ToS). This feature is an extension of the feature described in the chapter “Configuring IP to ATM Class of Service” in the *Cisco IOS Quality of Service Solutions Configuration Guide*. The original feature was limited to permanent virtual circuits (PVCs) only. This feature is an extension because it applies to SVCs.

The PVC bundle feature requires that the user configure PVCs for different IP ToS. The PVCs have to be set up throughout the ATM network between endpoints. The IP to ATM Class of Service Mapping for SVC Bundles feature needs configuration only at the endpoints. The user does not configure SVCs; the software sets up SVCs in a bundle between endpoints. When the router receives the first IP packet for the destination that is configured in the SVC bundle, that event triggers the creation of the SVC.

A default SVC is used for non-IP traffic, IP traffic with no precedence, and IP traffic with the precedence bit set but for which no SVC exists. SVC setup for the specific IP precedence traffic is triggered when the first IP packet with that precedence bit set is received.

Benefits

Multiple SVCs with Different QoS Parameters

Users can have multiple SVCs, each with different QoS parameters, between SVC endpoints. This allows the customer to easily offer differentiated services between SVC nodes.

Reduced Configuration

SVC bundle configuration requires less configuration than a PVC configuration. The PVC bundle feature needs the configuration of PVCs in bundles throughout the ATM network. However, an SVC bundle needs configuration only at the endpoints and uses the User-Network Interface (UNI) to set up SVCs in the bundle between endpoints.

Restrictions

- Both router platforms require enhanced ATM port adapters.

Related Features and Technologies

The SVC bundle feature is similar to the IP to ATM Class of Service feature, which is documented in the “Configuring IP to ATM Class of Service” chapter of the *Cisco IOS Quality of Service Solutions Configuration Guide*.

Related Documents

For related information on this feature, refer to the following documents:

- *Cisco IOS Quality of Service Solutions Configuration Guide*
- *Cisco IOS Quality of Service Solutions Command Reference*
- *Cisco IOS Wide-Area Networking Configuration Guide*
- *Cisco IOS Wide-Area Networking Command Reference*

Supported Platforms

- Cisco 7200 series with enhanced ATM port adapters

Platform Support Through Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Feature Navigator. Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

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.

To access 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 at <http://www.cisco.com/register>.

Feature Navigator is updated when major Cisco IOS software releases and technology releases occur. As of May 2001, Feature Navigator supports M, T, E, S, and ST releases. You can access Feature Navigator at the following URL:

<http://www.cisco.com/go/fn>

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

No new or modified MIBs are supported by this feature.

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

<http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml>

RFCs

None

Prerequisites

Before configuring the IP to ATM Class of Service for SVC Bundles feature, you should read and understand the following:

- *Cisco IOS Wide-Area Networking Configuration Guide*, “Configuring ATM” chapter
- *Cisco IOS Quality of Service Solutions Configuration Guide*, “IP to ATM Class of Service Overview” and “Configuring IP to ATM Class of Service” chapter

Configuration Tasks

The following sections describe configuration tasks for the IP to ATM Class of Service Mapping for SVC Bundles feature. Each task in the list is identified as either optional or required.

Note that the bundle members must be configured either directly, by bundle-level parameters, or by class. The bundle configuration should be the same on both ends (the end where the SVC is initiated and the end where it is terminated).

- [Creating an SVC Bundle](#) (required)
- [Configuring Bundle-Level Parameters](#) (optional)

- [Attaching a Class to a Bundle](#) (optional)
- [Configuring an SVC Bundle Member Directly](#) (optional)

Creating an SVC Bundle

To create an SVC bundle and enter SVC-bundle configuration mode, in which you can assign bundle-level parameters to the bundle and all of its member SVCs, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# bundle svc <i>bundle-name</i> nsap <i>destination-nsap-address</i>	Creates or modifies an SVC bundle. The name must be the same on both sides of the VC.

Configuring Bundle-Level Parameters

Configuring bundle-level parameters is optional if a VC class is attached to the bundle to configure it. To attach a class to a bundle, see the “[Attaching a Class to a Bundle](#)” section.

To configure parameters that apply to the bundle and all of its members, use the following commands in SVC-bundle configuration mode, as needed:

Command	Purpose
Router(config-if-atm-svc-bundle)# protocol ip <i>protocol-address</i> [broadcast]	Configures the destination network address of an SVC bundle.
Router(config-if-atm-svc-bundle)# encapsulation aal5 [snap mux ip]	Sets the encapsulation method used by the interface. By default, encapsulation aal5 snap is enabled.
Router(config-if-atm-svc-bundle)# class-bundle <i>vc-class-name</i>	(Optional) Configures a bundle with the bundle-level commands contained in the specified VC class.
Router(config-if-atm-svc-bundle)# oam-bundle [manage] [<i>frequency</i>]	(Optional) Enables end-to-end F5 operation, administration, and maintenance (OAM) loopback cell generation and OAM management for all VC members of a bundle or a VC class that can be applied to a VC bundle.

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 SVC-bundle configuration mode:

Command	Purpose
Router(config-if-atm-svc-bundle)# class-bundle <i>vc-class-name</i>	(Optional) Configures a bundle with the bundle-level commands contained in the specified VC class.

Configuring an SVC Bundle Member Directly

Configuring SVC bundle members directly is optional if a VC class is attached to the bundle member. Each SVC bundle can have a maximum of eight members. The number of members and the precedence values attached to them should be the same on both ends of the SVC (that is, where the SVC is initiated and where it is terminated).

To configure an individual SVC bundle member directly, use the following commands, as appropriate, starting in SVC-bundle configuration mode:

	Command	Purpose
Step 1	Router(config-if-atm-svc-bundle)# svc-bundle <i>svc-handle</i>	Creates or modifies a member of an SVC bundle.
	Router(config-if-atm-svc-member)# ubr <i>output-pcr</i> [<i>input-pcr</i>]	Configures the VC for unspecified bit rate (UBR) QoS and specifies the output peak cell rate (PCR) for it.
	Router(config-if-atm-svc-member)# ubr+ <i>output-pcr output-mcr</i> [<i>input-pcr</i>] [<i>input-mcr</i>]	Configures the VC for UBR QoS and specifies the output PCR and output minimum guaranteed cell rate for it.
	Router(config-if-atm-svc-member)# vbr-rt <i>peak-rate average-rate burst</i>	Configures the real-time variable bit rate (VBR).
	Router(config-if-atm-svc-member)# precedence [other <i>range</i>]	Configures the precedence levels for the VC.
	Router(config-if-atm-svc-member)# bump { implicit explicit <i>precedence-level</i> traffic }	Configures the bumping rules for the VC.
	Router(config-if-atm-svc-member)# idle-timeout <i>seconds</i> [<i>minimum-rate</i>]	Configure the idle timeout parameter for tearing down an ATM SVC.
	Router(config-if-atm-svc-member)# class-vc <i>vc-class-name</i>	Assigns a VC class to a VC bundle member.

Monitoring IP to ATM Class of Service Mapping for SVC Bundles

Use the following commands to monitor SVC bundles:

Command	Purpose
Router# debug atm bundle error	Displays debug messages for SVC bundle errors.
Router# debug atm bundle events	Displays SVC bundle events.
Router# show atm bundle svc <i>bundle-name</i>	Displays the bundle attributes assigned to each bundle VC member and the current working status of the VC members.
Router# show atm bundle svc <i>bundle-name</i> statistics	Displays the statistics of an SVC bundle.

Configuration Examples

This section provides the following configuration examples:

- [IP to ATM Class of Service Mapping with Bundle Parameters Configured in Bundle Mode Example](#)
- [IP to ATM Class of Service Mapping with Bundle Parameters Configured with the class-bundle Command Example](#)

IP to ATM Class of Service Mapping with Bundle Parameters Configured in Bundle Mode Example

In this example, the bundle parameters are configured in bundle mode. Initially, the end station ID (ESI) address and an Integrated Local Management Interface (ILMI) PVC are configured.

The PVC helps in getting the prefix from the switch (for example, an LS 1010). The combined address is the NSAP address.

You also need to know the other NSAP address to configure the SVC bundle. The eight VC classes are configured with precedences and traffic parameters. The classes must be configured before you attach them to the specific members. The **vc-class** commands could also be configured in the bundle-member configuration. The configuration of the members must be the same at both ends (that is, where the bundle is initiated and where it is terminated).

```
vc-class atm seven
  vbr-nrt 10000 5000 32
  precedence 7
!
vc-class atm six
 ubr 6000
  precedence 6
!
vc-class atm five
ubr 5000
precedence 5
bump explicit 7
!
vc-class atm four
ubr 4000
precedence 4
!
vc-class atm three
ubr 3000
precedence 3
!
vc-class atm two
ubr 2000
precedence 2
!
vc-class atm one
ubr 1000
precedence 1
!
vc-class atm zero
ubr 500
precedence other
!
no ip address
no ip mroute-cache
```

```

no atm ilmi-keepalive
atm voice aal2 aggregate-svc upspeed-number 0
pvc qsaal 0/5 qsaal
!
pvc ilmi 0/16 ilmi
!
bundle-enable
!
interface ATM1/0.1 multipoint
ip address 170.100.9.2 255.255.255.0
atm esi-address 11111111111.11
bundle svc test nsap 47.0091810000000003E3924F01.999999999999.99
protocol ip 170.100.9.1
broadcast
oam retry 4 3 10
encapsulation aal5snap
oam-bundle manage
svc-bundle seven
class-vc seven
svc-bundle six
class-vc six
svc-bundle five
class-vc five
svc-bundle four
class-vc four
svc-bundle three
class-vc three
svc-bundle two
class-vc two
svc-bundle one
class-vc one
svc-bundle zero
class-vc zero
!

```

IP to ATM Class of Service Mapping with Bundle Parameters Configured with the class-bundle Command Example

In this example, the bundle parameters are added to the bundle by using the **class-bundle** command. The class attached is named “sanjose”.

```

vc-class atm sanjose          !Here we are attaching this vc-class to the whole bundle
broadcast
oam retry 4 3 10
encapsulation aal5snap
oam-bundle manage 3
!
vc-class atm med
ubr 10000
precedence 4-5
!
vc-class atm high
vbr-nrt 10000 5000 32
precedence 6-7
!
vc-class atm low
ubr+ 100000 5000
precedence 0-3

interface ATM1/0

```

```

ip address 3.3.3.1 255.255.255.0
atm idle-timeout 5
atm esi-address 665544332211.22
no atm ilmi-keepalive
atm voice aal2 aggregate-svc upspeed-number 0
pvc 0/5 qsaal
!
pvc 0/16 ilmi
!
pvc 0/100
!
bundle svc svc-test nsap 47.00918100000000003E3924F01.998877665533.88
class-bundle bundle-test
protocol ip 3.3.3.2
svc-bundle high
  class-vc high
svc-bundle med
  class-vc med
svc-bundle low
  class-vc low
!

```

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at <http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm>.

- **bump**
- **bundle svc**
- **class-bundle**
- **class-vc**
- **debug atm bundle error**
- **debug atm bundle events**
- **encapsulation aal5**
- **idle-timeout**
- **oam-bundle**
- **precedence (VC bundle)**
- **protocol (ATM)**
- **show atm bundle svc**
- **show atm bundle svc statistics**
- **svc-bundle**
- **ubr**
- **ubr+**
- **vbr-rt**



ATM PVC Bundle Enhancement — MPLS EXP-Based PVC Selection

Feature History

Release	Modification
12.2(8)T	This feature was introduced.

This document describes the ATM PVC Bundle Enhancement — MPLS EXP-Based PVC Selection feature in Cisco IOS Release 12.2(8)T. It includes the following sections:

- [Feature Overview, page 819](#)
- [Supported Platforms, page 822](#)
- [Supported Standards, MIBs, and RFCs, page 823](#)
- [Configuration Tasks, page 824](#)
- [Configuration Examples, page 827](#)
- [Command Reference, page 831](#)

Feature Overview

The ATM PVC Bundle Enhancement — MPLS EXP-Based PVC Selection feature is an extension to the IP to ATM Class of Service feature suite. The IP to ATM Class of Service feature suite, using virtual circuit (VC) support and bundle management, maps quality of service (QoS) characteristics between IP and ATM. It provides customers who have multiple VCs (with varying qualities of service to the same destination) the ability to build a QoS differentiated network.

The IP to ATM Class of Service feature suite allowed customers to use IP precedence level as the selection criteria for packet forwarding. This new feature now gives customers the option of using the Multiprotocol Label Switching (MPLS) experimental (EXP) level as an additional selection criteria for packet forwarding.



Note

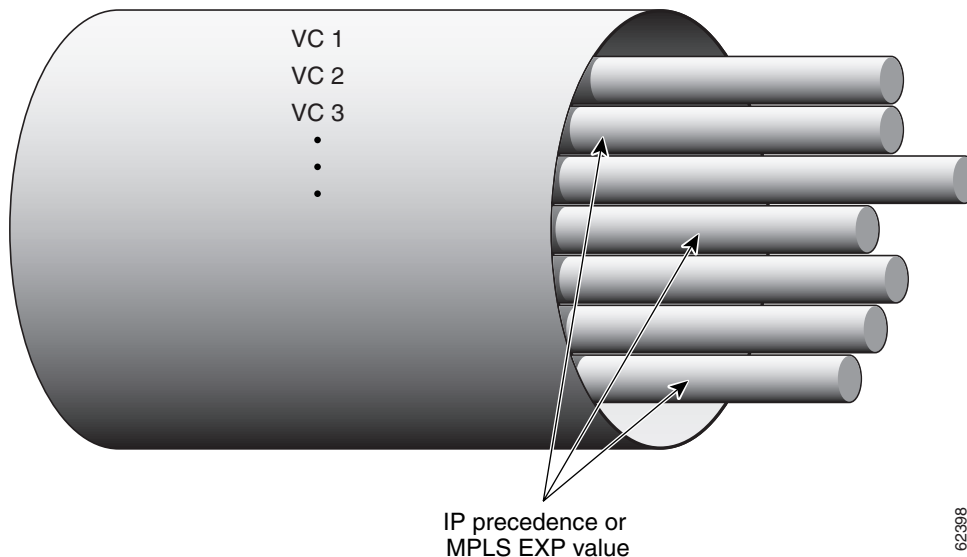
If a selection criteria for packet forwarding is not selected (that is, if the packet is unlabeled), this new feature uses the IP precedence level as the default selection criteria.

For more information about the IP to ATM Class of Service feature suite, refer to the “Configuring IP to ATM Class of Service” chapter of the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2.

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 1](#), these VCs are grouped in a bundle and are referred to as bundle members.

Figure 45 ATM VC Bundle

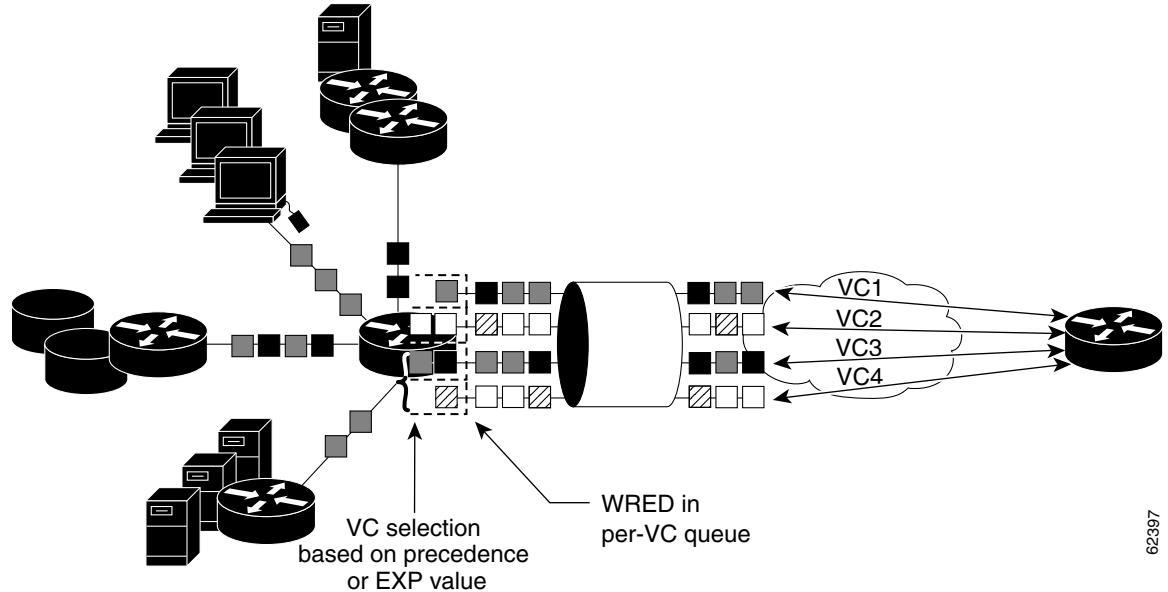


ATM VC bundle management allows you to define an ATM VC bundle and add VCs to it. Each VC of a bundle has its own ATM traffic class and ATM traffic parameters. You can apply attributes and characteristics to discrete VC bundle members, or you can apply them collectively at the bundle level.

Using VC bundles, you can create differentiated service by flexibly distributing MPLS EXP levels over the different VC bundle members. You can map a single MPLS EXP level, or a range of these levels, to each discrete VC in the bundle, thereby enabling individual VCs in the bundle to carry packets marked with different MPLS EXP levels. You can use Weighted Random Early Detection (WRED) or distributed WRED (dWRED) to further differentiate service across traffic that has different MPLS EXP levels.

To determine which VC in the bundle to use to forward a packet to its destination, the ATM VC bundle management software matches MPLS EXP levels between packets and VCs (see [Figure 2](#)). 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 MPLS EXP level of the type of service (ToS) byte of its header. The ATM VC bundle management software matches the MPLS EXP level of the packet to the MPLS EXP levels assigned to a VC, sending the packet out on the appropriate VC. Moreover, the ATM VC bundle management software allows you to configure how traffic will be redirected when the VC to which the packet was initially directed goes down. [Figure 2](#) 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 46 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 configure the network to provide IP traffic belonging to real-time class of service (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 also allowing the network to transport nonreal-time traffic over a more elastic ATM unspecified bit rate (UBR) PVC. UBR is effectively the ATM version of best-effort service. Using a configuration such as this would allow you to make full use of your network capacity.

Benefits

Improved System Performance

This feature 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, ensuring greater QoS for more important traffic and user types.

Additional Selection Criteria

This new feature now gives customers the option of using the MPLS EXP level, in addition to IP precedence, as a selection criteria for packet forwarding.

Restrictions

- This feature requires ATM PVC management, as well as Forwarding Information Base (FIB) and Tag Forwarding Information Base (TFIB) switching functionality.
- This feature is not supported on either the ATM interface processor (AIP) or the ATM Lite port adapter (PA-A1).
- The router at the remote end of the network must be using a version of Cisco IOS that supports MPLS and ATM PVC management.

Related Features and Technologies

This feature is similar to the IP to ATM Class of Service feature suite, which is documented in the “Configuring IP to ATM Class of Service” chapter of the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2.

Related Documents

- *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2
- *Cisco IOS Quality of Service Solutions Command Reference*, Release 12.2
- *Cisco IOS Switching Services Configuration Guide*, Release 12.2
- *Cisco IOS Switching Services Command Reference*, Release 12.2
- *Cisco IOS Wide-Area Networking Configuration Guide*, Release 12.2
- *Cisco IOS Wide-Area Networking Command Reference*, Release 12.2
- *IP to ATM SVC Bundles for Class of Service (CoS) Mapping*, Cisco IOS Release 12.2(4)T feature module
- *MPLS Label Distribution Protocol*, Cisco IOS Release 12.2(4)T feature module

Supported Platforms

- Cisco 3600 series
The ATM Adapter PA-A3 is not supported on either the Cisco 3620 router or the Cisco 3640 router. Because certain QoS features (for example, WRED) require the ATM Adapter PA-A3, specific limitations may apply. For more information about platform and feature support, refer to Cisco Feature Navigator (described below).
- Cisco 3725
- Cisco 3745
- Cisco 7200 series
- Cisco 7500 series

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that support 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 at <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>

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

No new or modified MIBs are supported by this feature.

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

<http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml>

RFCs

No new or modified RFCs are supported by this feature.

Configuration Tasks

See the following sections for configuration tasks for the ATM PVC Bundle Enhancement — MPLS EXP-Based PVC Selection feature. Each task in the list is identified as either required or optional.

- [Enabling MPLS](#) (required)
- [Creating a VC Bundle](#) (required)
- [Applying Parameters to Bundles](#) (required)
 - [Configuring Bundle-Level Parameters](#) (required)
 - [Configuring a VC Bundle Member Directly](#) (optional)
 - [Configuring VC Class Parameters to Apply to a Bundle](#) (optional)
 - [Attaching a Class to a Bundle](#) (optional)
- [Verifying the Configuration](#) (optional)

Enabling MPLS

To enable MPLS, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# ip cef	Enables Cisco Express Forwarding (CEF) on the Route Processor (RP) card. An optional keyword distributed can be used with this command to enable distributed CEF (dCEF) for the Versatile Interface Processor (VIP)-based platforms.
Step 2	Router(config)# mpls label protocol ldp	Specifies the default label distribution protocol for a platform.
Step 3	Router(config)# interface <i>type number</i> [<i>name-tag</i>]	Configures an interface type and enters interface configuration mode.
Step 4	Router(config-if)# mpls ip	Enables MPLS forwarding of IPv4 packets along normally routed paths for the platform.

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 to all of its member VCs, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# bundle <i>bundle-name</i>	Creates the specified bundle and enters bundle configuration mode.

Applying Parameters to Bundles

Parameters (or attributes) can be applied to bundles either by applying the parameters directly to the bundle or by applying the parameters to a VC class assigned to the bundle.

Applying parameters by using VC classes assigned to the bundle allows you to apply multiple parameters at once because you apply the VC class to the bundle and to all of its VC members. This method allows you to apply a parameter across all VCs for the bundle, after which (for some parameters) you can later modify that parameter for individual VCs. After configuring the parameters for the VC class, you need to attach the VC class to the bundle.

Parameters applied directly to a bundle take priority over those applied to VC classes assigned to the bundle. Parameters applied to VC classes assigned to the bundle take priority over those applied to individual VCs.



Note

Note that some 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.

To begin applying parameters to bundles, complete the procedure in the section [“Configuring Bundle-Level Parameters.”](#)

Configuring Bundle-Level Parameters

To begin configuring parameters that apply to the bundle and to all of its members, use the following commands in bundle configuration mode, as needed:

Command	Purpose
Router(config-if-atm-bundle)# protocol <i>protocol</i> { <i>protocol-address</i> inarp } [[no] broadcast]	Configures a static map or enables Inverse Address Resolution Protocol (Inverse ARP) or Inverse ARP broadcasts for the bundle.
Router(config-if-atm-bundle)# encapsulation <i>aal-encap</i>	Configures the ATM adaptation layer (AAL) and encapsulation type for the bundle.

What's Next?

Next, decide if you want to configure the VC bundle member directly or use a VC class attached to a bundle. To configure the VC bundle member directly, complete just the procedure in the [“Configuring a VC Bundle Member Directly”](#) section. To use a VC class attached to a bundle, instead complete the procedures in both the [“Configuring VC Class Parameters to Apply to a Bundle”](#) section and the [“Attaching a Class to a Bundle”](#) section.

Configuring a VC Bundle Member Directly

To configure an individual VC bundle member directly, use the following commands in `bundle-vc` configuration mode, as needed. To enter `bundle-vc` configuration mode, use the **pvc-bundle** command.

Command	Purpose
Router(config-if-atm-member)# ubr <i>output-pcr</i> [<i>input-pcr</i>]	Configures the VC for UBR QoS and specifies the output peak cell rate (PCR) for it.
Router(config-if-atm-member)# vbr-nrt <i>output-pcr</i> <i>output-scr</i> <i>output-mbs</i> [<i>input-pcr</i>] [<i>input-scr</i>] [<i>input-mbs</i>]	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)# mpls experimental [other <i>range</i>]	Configures the MPLS EXP levels for the VC.
Router(config-if-atm-member)# bump { implicit explicit <i>precedence-level</i> 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.

Configuring VC Class Parameters to Apply to a Bundle

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] [<i>frequency</i>]	Enables end-to-end F5 Operation, Administration, and Maintenance (OAM) loopback cell generation and OAM management for all VCs in the bundle.

In addition to the **oam-bundle** 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 on these commands, including configuration tasks and command syntax, refer to the *Cisco IOS Wide-Area Networking Configuration Guide*, Release 12.2 and the *Cisco IOS Wide-Area Networking Command Reference*, Release 12.2.



Note

If you are using a VC class to configure the bundle, you must attach the VC class to the bundle. To do this, complete the procedure in the section “[Attaching a Class to a Bundle](#).”

Attaching a Class to a Bundle

To attach a VC class containing bundle-level configuration commands to a bundle, use the following command in bundle configuration mode. To enter bundle configuration mode, use the **bundle** command.

Command	Purpose
Router(config-if-atm-bundle)# class-bundle <i>vc-class-name</i>	Configures a bundle with the bundle-level commands contained in the specified VC class.

Verifying the Configuration

To verify the configuration of the feature, use the following commands in EXEC mode, as needed:

Command	Purpose
Router# debug atm bundle error	Displays debug messages for PVC bundle errors.
Router# debug atm bundle events	Displays PVC bundle events.
Router# show atm map	Displays the list of all configured ATM static maps to remote hosts on an ATM network.
Router# show atm bundle <i>bundle-name</i>	Displays the bundle attributes assigned to each VC member and the current working status of the VC members.
Router# show mpls forwarding-table	Displays the contents of the MPLS FIB.

Configuration Examples

This section provides the following configuration example:

- [VC Bundle Configuration Using a VC Class Example](#)

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,” which 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 the following parameters: `aal5snap` encapsulation, broadcast on, use of Inverse ARP to resolve IP addresses, and 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 4 3 10
```

The following four sections of the configuration define specific VC classes. Each of these classes contains commands used to specify parameters that can then 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 MPLS EXP 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
 mpls experimental 7
 protect vc
 vbr-nrt 1000 5000 32
```

Premium-Class Class

When the class called “premium-class” is applied to a VC, the VC carries traffic whose MPLS EXP 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 MPLS EXP 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
 mpls experimental 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 an MPLS EXP level 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
 mpls experimental 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 **mpls experimental other** command to carry traffic with MPLS EXP 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 isubr.

```
vc-class atm basic-class
mpls experimental 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 a1/0.1 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 MPLS EXP VC in the bundle. The OSI data packets, if any, will use the lowest MPLS EXP VC in the bundle. If configured, other protocols, such as Internet Packet Exchange (IPX) or AppleTalk, will always use the lowest MPLS EXP 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, the MPLS EXP level 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
  mpls experimental 7-5
  protect vc
  class-vc premium-class
pvc-bundle la-mid 404
  mpls experimental 4-2
```

```
protect group
class-vc priority-class
pvc-bundle la-low 401
mpls experimental other
protect group
class-vc basic-class
```

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at <http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm>.

New Commands

- `mpls experimental`

Modified Commands

- `show mpls forwarding-table`



QoS Features for Voice

This part consists of the following:

- [QoS Features for Voice](#)
- [Voice and Quality of Service Features for ADSL and G.SHDSL on Cisco 1700, Cisco 2600, and Cisco 3600 Series Routers](#)



QoS Features for Voice

Real-time applications such as voice applications have different characteristics and requirements from those of traditional data applications. Because they are real-time-based, voice applications tolerate minimal variation in the amount of delay affecting delivery of their voice packets. Voice traffic is also intolerant of packet loss and jitter, both of which degrade unacceptably the quality of the voice transmission delivered to the recipient end user. To effectively transport voice traffic over IP, mechanisms are required that ensure reliable delivery of packets with low latency. Cisco IOS QoS features collectively embody these techniques, offering the means to provide priority service that meets the stringent requirements of voice packet delivery.

This chapter only provides a high-level overview of Cisco IOS QoS features for voice. For complete conceptual and configuration information for each feature, see the referenced chapters or books.

For a list of related Cisco IOS voice documentation, see the section “[For More Information](#)” in this chapter.

Cisco IOS QoS for Voice Features

Cisco IOS includes a rich set of features that enable you to deploy mechanisms that deliver QoS throughout your network. Following are some of the Cisco IOS features that address the requirements of end-to-end QoS and service differentiation for voice packet delivery:

- **Compressed Real-Time Protocol (CRTP)**—Used in conjunction with RTP, compresses the extensive RTP header, resulting in decreased consumption of available bandwidth for voice traffic. A corresponding reduction in delay is realized. For conceptual information on CRTP, see the chapter [Link Efficiency Mechanisms Overview](#) in this book.
- **Frame Relay Traffic Shaping (FRTS)**—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 conceptual information on FRTS, see the chapter [Policing and Shaping Overview](#) in this book. For information on how to configure FRTS, refer to the *Cisco IOS Wide-Area Networking Configuration Guide*.
- **FRF.12**—Ensures predictability for voice traffic, aiming to provide better throughput on low-speed Frame Relay links by interleaving delay-sensitive voice traffic on one virtual circuit (VC) with fragments of a long frame on another VC utilizing the same interface. For more information about FRF.12, refer to the *Cisco IOS Wide-Area Networking Configuration Guide*.
- **PSTN Fallback**—The Public Switched Telephone Network (PSTN) Fallback feature provides a mechanism to monitor congestion in the IP network and either redirect calls to the PSTN or reject calls based on the network congestion. For more information about PSTN Fallback, refer to the *Cisco IOS Voice, Video, and Fax Configuration Guide*.

- IP RTP Priority and Frame Relay IP RTP Priority—Provides a strict priority queuing scheme that allows delay-sensitive data such as voice to be dequeued and sent before packets in other queues are dequeued. These features are especially useful on slow-speed WAN links, including Frame Relay, Multilink PPP (MLP), and T1 ATM links. It works with WFQ and CBWFQ. For conceptual information on IP RTP Priority and Frame Relay IP RTP Priority, see the chapter [Congestion Management Overview](#) in this book. For information on how to configure IP RTP Priority and Frame Relay IP RTP Priority, see the chapter in this book.
- IP to ATM Class of Service (CoS)—Includes a feature suite that maps QoS characteristics between IP and ATM. Offers differential service classes across the entire WAN, not just the routed portion. Gives mission-critical applications exceptional service during periods of high network usage and congestion. For conceptual information on IP to ATM CoS, see the chapter [IP to ATM Class of Service Overview](#) in this book. For information on how to configure IP to ATM CoS, see the chapter in this book.
- Low latency queuing (LLQ)—Provides strict priority queuing on ATM VCs and serial interfaces. This feature allows you to configure the priority status for a class within CBWFQ, and is not limited to User Datagram Protocol (UDP) port numbers, as is IP RTP Priority. For conceptual information on LLQ, see the chapter [Congestion Management Overview](#) in this book. For information on how to configure LLQ, see the chapter in this book.
- MLP with Link Fragmentation and Interleaving (LFI)—Allows large packets to be multilink-encapsulated and fragmented so that they are small enough to satisfy the delay requirements of real-time traffic. LFI also provides a special transmit queue for the smaller, delay-sensitive packets, enabling them to be sent earlier than other flows. For conceptual information on MLP with LFI, see the chapter [Link Efficiency Mechanisms Overview](#) in this book.
- QoS Policy Propagation via Border Gateway Protocol (BGP)—Leverages BGP to distribute QoS policy to remote routers in your network. It allows you to classify packets and then use other QoS features such as CAR and Weighted Random Early Detection (WRED) to specify and enforce business policies to fit your business model. For conceptual information on Policy Propagation via BGP, see the chapter [Classification Overview](#) in this book. For information on how to configure Policy Propagation via BGP, see the chapter [Configuring QoS Policy Propagation via Border Gateway Protocol](#) in this book.
- Resource Reservation Protocol (RSVP)—Supports the reservation of resources across an IP network, allowing end systems to request QoS guarantees from the network. For networks supporting Voice over IP (VoIP), RSVP—in conjunction with features that provide queuing, traffic shaping, and voice call signalling—can provide call admission control for voice traffic. Cisco also provides RSVP support for LLQ and Frame Relay. For conceptual information on RSVP, see the chapter [Signalling Overview](#) in this book. For information on how to configure RSVP, see the chapter in this book. For information on how to configure RSVP support for LLQ, see the chapter [Configuring RSVP Support for LLQ](#) in this book. For information on how to configure RSVP support for Frame Relay, see the chapter [Configuring RSVP Support for Frame Relay](#) in this book.

Cisco IOS QoS for voice features are best deployed at different points in the network and are designed to be used in conjunction with other QoS features to achieve specific goals such as control over jitter and delay. Not all QoS for voice features are supported on all platforms.

For More Information

For additional information about Cisco IOS QoS for voice, refer to the following publications:

- *Cisco IOS Voice, Video, and Fax Configuration Guide*—This guide shows you how to configure your Cisco router or access server to support voice, video, and broadband transmission.

- *Cisco IOS Voice, Video, and Fax Command Reference*—This publication documents commands used to configure your Cisco router or access server to support voice, video, and broadband transmission.



Voice and Quality of Service Features for ADSL and G.SHDSL on Cisco 1700, Cisco 2600, and Cisco 3600 Series Routers

Feature History

Release	Modification
12.2(2)XQ	Voice and quality of service features for ADSL were introduced for Cisco 1700 series routers.
12.2(2)XK	Voice and quality of service features for ADSL were introduced for Cisco 2600 and Cisco 3600 series routers.
12.2(4)XL	This release further expanded voice and quality of service features for ADSL on Cisco 2600 and Cisco 3600 routers and provided full support for G.SHDSL on Cisco 1700 and Cisco 2600 series routers.
12.2(13)T	The voice and quality of service features for ADSL and G.SHDSL were fully implemented on Cisco 1700 series routers. Note The Tunable Transmission Ring feature is not supported on Cisco 1700 series routers in Cisco IOS Release 12.2(13)T.

This document describes the voice and quality of service (QoS) features for asymmetric digital subscriber lines (ADSL) and for single-pair high-bit-rate digital subscriber lines (G.SHDSL) on Cisco 1700 series, Cisco 2600 series, and Cisco 3600 series routers in Cisco IOS Release 12.2(13)T.

This document includes the following sections:

- [Feature Overview, page 840](#)
- [Supported Platforms, page 850](#)
- [Supported Standards, MIBs, and RFCs, page 851](#)
- [Prerequisites, page 851](#)
- [Configuration Tasks, page 852](#)
- [Configuration Examples, page 856](#)
- [Command Reference, page 871](#)

Feature Overview

Cisco 1700 series, Cisco 2600 series, and Cisco 3600 series routers with ADSL or G.SHDSL WAN interface cards support the integration of voice and data over the same ADSL or G.SHDSL circuit using Voice over IP (VoIP). Cisco 2600 series and Cisco 3600 series routers with ADSL or G.SHDSL WAN interface cards also support the integration of voice and data over the same ADSL or G.SHDSL circuit using Voice over ATM (VoATM).



Note

- To configure these voice and QoS features, you must first install and configure the ADSL or G.SHDSL WAN interface card (WIC) on your Cisco 1700 series, Cisco 2600 series, or Cisco 3600 series router. Refer to the installation and configuration instructions in the following documents:
 - [Configuring an ADSL WAN Interface Card on Cisco 1700 Series Routers](#)
 - [Installing the G.SHDSL WIC on the Cisco 1700 Series Router](#)
 - [1-Port ADSL WAN Interface Card for Cisco 2600 Series and 3600 Series Routers, Release 12.2\(4\)T](#)
 - [1-Port G.SHDSL WAN Interface Card for Cisco 2600 Series and 3600 Series Routers, Release 12.2\(4\)XL](#)

Quality of service (QoS) features make it possible to effectively combine voice and data traffic in the same WAN connection without sacrificing quality and reliability. Service providers can increase revenue by building differentiated service options based on premium, standard, or best-effort service classes.

The following voice and QoS features are supported in the Cisco IOS Release 12.2(13)T.

- [Classification and Marking](#)
 - [Class-Based Packet Marking with Differentiated Services](#)
 - [Committed Access Rate](#)
 - [Dial-Peer DSCP and IP Precedence Marking](#)
 - [IP QoS Map to ATM CoS](#)
 - [Local Policy Routing](#)
 - [Policy-Based Routing](#)
- [Queueing and Scheduling](#)
 - [Class-Based Weighted Fair Queueing](#)
 - [Low Latency Queueing](#)
 - [Per-VC Queueing](#)
- [Congestion Avoidance](#)
 - [Class-Based WRED with DSCP \(egress\)](#)
- [Policing and Traffic Shaping](#)
 - [Class-Based Policing](#)
 - [VC Shaping for VBR-NRT](#)

- [Link Latency](#)
 - [MLP with LFI – Bundling of VCs Across xDSL Interfaces](#)
 - [Tunable Transmission Ring](#) (supported only on Cisco 2600 and Cisco 3600 series routers; not supported on Cisco 1700 series routers until Cisco IOS Release 12.2(8)YN)
- [Other \(IP QoS\)](#)
 - [Access Control Lists](#)
 - [IP QoS Map to ATM CoS](#)
- [Additional Supported Features](#)
 - [F5 OAM CC Segment Functionality](#)
 - [H.323 and Media Gateway Control Protocol](#)
 - [ILMI](#)
 - [Multiple PVC Support](#)
 - [RFC 1483 Routing](#)

Table 14 lists the voice and QoS features for ADSL and G.SHDSL and the releases in which they are available.

Table 14 *Voice and QoS Features and the Releases in Which They Are Available*

Feature	ADSL	G.SHDSL
Access Control Lists	12.2(2)XK, 12.2(4)XL, and 12.2(13)T	12.2(4)XL and 12.2(13)T
Class-Based Packet Marking with DSCP	12.2(2)XK, 12.2(4)XL, and 12.2(13)T	12.2(4)XL and 12.2(13)T
Class-Based Policing	12.2(2)XK, 12.2(4)XL, and 12.2(13)T	12.2(4)XL and 12.2(13)T
Class-Based Weighted Fair Queueing	12.2(2)XK, 12.2(4)XL, and 12.2(13)T	12.2(4)XL and 12.2(13)T
Class-Based WRED with DSCP (egress)	12.2(2)XK, 12.2(4)XL, and 12.2(13)T	12.2(4)XL and 12.2(13)T
Committed Access Rate	12.2(2)XK, 12.2(4)XL, and 12.2(13)T	12.2(4)XL and 12.2(13)T
Dial-Peer DSCP and IP Precedence Marking	12.2(2)XK, 12.2(4)XL, and 12.2(13)T	12.2(4)XL and 12.2(13)T
F5 OAM CC Segment Functionality	12.2(4)XL and 12.2(13)T	12.2(4)XL and 12.2(13)T
H.323 and Media Gateway Control Protocol	12.2(2)XK, 12.2(4)XL, and 12.2(13)T	12.2(4)XL and 12.2(13)T
ILMI	12.2(4)XL and 12.2(13)T	12.2(4)XL and 12.2(13)T
IP QoS Map to ATM CoS	12.2(2)XK, 12.2(4)XL, and 12.2(13)T	12.2(4)XL and 12.2(13)T
Local Policy Routing	12.2(2)XK, 12.2(4)XL, and 12.2(13)T	12.2(4)XL and 12.2(13)T
Low Latency Queueing	12.2(2)XK, 12.2(4)XL, and 12.2(13)T	12.2(4)XL and 12.2(13)T

Table 14 Voice and QoS Features and the Releases in Which They Are Available (continued)

MLP with LFI - Bundling of VCs Across xDSL Interfaces	12.2(2)XK and 12.2(13)T	12.2(4)XL and 12.2(13)T
Multiple PVC Support	12.2(2)XK, 12.2(4)XL, and 12.2(13)T	12.2(4)XL and 12.2(13)T
Per-VC Queuing	12.2(2)XK, 12.2(4)XL, and 12.2(13)T	12.2(4)XL and 12.2(13)T
Policy-Based Routing	12.2(2)XK, 12.2(4)XL, and 12.2(13)T	12.2(4)XL and 12.2(13)T
RFC 1483 Routing	12.2(2)XK, 12.2(4)XL, and 12.2(13)T	12.2(4)XL and 12.2(13)T
Tunable Transmission Ring	12.2(2)XK, 12.2(4)XL, and 12.2(13)T	12.2(4)XL and 12.2(13)T
VC shaping for VBR-NRT	12.2(2)XK, 12.2(4)XL, and 12.2(13)T	12.2(4)XL and 12.2(13)T

Note G.SHDSL WICs are not supported on Cisco 3600 series routers in the Cisco IOS Release 12.2(4)XL

Classification and Marking

The following existing Cisco IOS classification and marking features are supported on ADSL and G.SHDSL WICs:

- [Class-Based Packet Marking with Differentiated Services](#)
- [Committed Access Rate](#)
- [Dial-Peer DSCP and IP Precedence Marking](#)
- [Local Policy Routing](#)
- [Policy-Based Routing](#)

Class-Based Packet Marking with Differentiated Services

For information about class-based packet marking with differentiated services, refer to the following document:

- The chapter “[Quality of Service Overview](#)” in the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2

Committed Access Rate

For information about committed access rate (CAR), refer to the following document:

- The chapter “[Quality of Service Overview](#)” in the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2

Dial-Peer DSCP and IP Precedence Marking

For information about dial-peer differentiated services code points (DSCPs) and IP precedence marking, refer to the following document:

- The chapter [“Quality of Service for Voice over IP”](#) in the *Cisco IOS Quality of Service Solutions*.

Local Policy Routing

For information about local policy routing (LPR), refer to the following documents:

- The chapter [“Configuring IP Routing Protocol—Independent Features”](#) in the *Cisco IOS IP Configuration Guide*, Release 12.2
- The chapter [“Configuring IP Routing Protocols”](#) in the *Router Products Configuration Guide*

Policy-Based Routing

For information about policy-based routing (PBR), refer to the following documents:

- The chapter [“Quality of Service Overview”](#) in the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2
- The chapter [“Configuring Policy-Based Routing”](#) in the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2

Queueing and Scheduling

The following existing Cisco IOS queueing and scheduling features are supported on ADSL WICs and G.SHDSL WICs:

- [Class-Based Weighted Fair Queueing](#)
- [Low Latency Queueing](#)
- [Per-VC Queueing](#)

Class-Based Weighted Fair Queueing

For information about class-based weighted fair queueing (CBWFQ), refer to the following document:

- The chapter [“Quality of Service Overview”](#) in the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2

Low Latency Queueing

For information about low latency queueing (LLQ), refer to the following documents:

- The chapter [“Congestion Management Overview”](#) in the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2
- The chapter [“Quality of Service for Voice over IP”](#) in the *Cisco IOS Quality of Service Solutions* document

**Note**

Low latency queueing works in conjunction with setting the transmission (tx) ring. (For more information about setting the tx ring, see the section “[Tunable Transmission Ring](#).”)

Per-VC Queueing

Per-virtual circuit (per-VC) queueing is supported on ADSL and G.SHDSL interfaces at the driver level, similar to VC-queueing features on other ATM interfaces. This feature underlies many of the Cisco IOS QoS queueing features, such as LLQ.

For more information about per-VC queueing, refer to the following documents:

- [Understanding Weighted Fair Queuing on ATM](#)
- [Per-VC Class-Based, Weighted Fair Queuing \(Per-VC CBWFQ\) on the Cisco 7200, 3600, and 2600 Routers](#)

Congestion Avoidance

The following existing Cisco IOS congestion avoidance feature is supported on ADSL and G.SHDSL WICs:

- [Class-Based WRED with DSCP \(egress\)](#)

Class-Based WRED with DSCP (egress)

For information about class-based weighted random early detection (WRED), refer to the following documents:

- The chapter “[Quality of Service Overview](#)” in the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2
- [Cisco IOS Quality of Service Solutions Command Reference](#), Release 12.2
- [DiffServ Compliant Weighted Random Error Detection](#)

Policing and Traffic Shaping

The following existing Cisco IOS policing and shaping features are now supported on ADSL and G.SHDSL WICs:

- [Class-Based Policing](#)
- [VC Shaping for VBR-NRT](#)

Class-Based Policing

For information about traffic classes and traffic policies, refer to the following document:

- The chapter “[Configuring Traffic Policing](#)” in the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2

VC Shaping for VBR-NRT

For information about VC shaping for variable bit rate-nonreal time (VBR-NRT), refer to the following document:

- [Understanding the VBR-nrt Service Category and Traffic Shaping for ATM VCs](#)

Link Latency

The following link latency features are supported on ADSL and G.SHDSL WICs:

- [MLP with LFI – Bundling of VCs Across xDSL Interfaces](#)
- [Tunable Transmission Ring](#)

MLP with LFI – Bundling of VCs Across xDSL Interfaces

For information about the Multilink PPP Link Fragmentation and Interleaving (MLP and LFI) – Bundling of VCs Across ADSL and G.SHDSL (xDSL) Interfaces feature, refer to the following document:

- The chapter “[Configuring Link Fragmentation and Interleaving for Multilink PPP](#)” in the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2

Tunable Transmission Ring

The tx ring is the first-in, first-out (FIFO) buffer used to hold frames before transmission at the DSL driver level. The tx ring defines the maximum number of packets that can wait for transmission at Layer 2.

The tx ring complements the ability of LLQ to minimize jitter and latency of voice packets. For maximum voice quality, a low tx ring setting should be used. For maximum data throughput, a high tx ring setting should be used.

You can configure the size of the tx ring for each permanent virtual circuit (PVC). The default value is 60. However, the value of the setting can be changed to 2 or 3. (The only permitted values are 2, 3, or 60). A low tx ring setting, such as 2 or 3, is required for latency-critical traffic. For example, when the tx ring limit is configured as 3 and LLQ is configured on the PVC, the worst case delay for a voice packet is the time required to transmit three data packets. When the buffering is reduced by configuring the tx ring limit, the delay experienced by voice packets is reduced by a combination of the tx ring and LLQ mechanism.



Note

- The size of the tx ring buffer is measured in packets, not particles.
 - The Tunable Transmission Ring feature is not supported on Cisco 1700 series routers until Cisco IOS Release 12.2(8)YN.
-

Other (IP QoS)

The following IP QoS features are supported on ADSL and G.SHDSL WICs:

- [Access Control Lists](#)
- [IP QoS Map to ATM CoS](#)

Access Control Lists

For information about configuring access control lists, refer to the following document:

- The chapter “[Configuring IP Services](#)” in the *Cisco IOS IP Configuration Guide*, Release 12.2

IP QoS Map to ATM CoS

For information about IP QoS map to ATM class of service (CoS), refer to the following document:

- The chapter “[Configuring IP to ATM Class of Service](#)” in the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2

Additional Supported Features

The following existing IOS features are supported on ADSL and G.SHDSL WICs:

- [F5 OAM CC Segment Functionality](#)
- [H.323 and Media Gateway Control Protocol](#)
- [ILMI](#)
- [Multiple PVC Support](#)
- [RFC 1483 Routing](#)

F5 OAM CC Segment Functionality

For information about F5 Operation, Administration, and Maintenance Continuity Check (F5 OAM CC) segment functionality, refer to the following documents:

- “[Cisco Product Bulletin No. 1518](#)” about Cisco IOS software Release 12.2(2)XJ
- [Release Notes for the Cisco 1700 Series Routers for Cisco IOS Release 12.2\(XJ\)](#)

H.323 and Media Gateway Control Protocol

For information about Cisco H.323 and Media Gateway Control Protocol (MGCP) features, refer to the following documents:

- The chapter “[Configuring Voice over IP](#)” in the *Cisco IOS Voice, Video, and Fax Configuration Guide*, Release 12.2
- The chapter “[H.323 Applications](#)” in the *Cisco IOS Voice, Video, and Fax Configuration Guide*, Release 12.2

ILMI

For information about Integrated Local Management Interface (ILMI) protocol implementation for Cisco digital subscriber loop access multiplexers (DSLAMs) with N1-2 cards, refer to the following document:

- The chapter “[Configuring ILMI](#)” in the *Configuration Guide for Cisco DSLAMS with N1-2*

Multiple PVC Support

For information about PVCs, refer to the following documents:

- The chapter “[Wide-Area Networking Overview](#)” in the *Cisco IOS Wide-Area Networking Configuration Guide*, Release 12.2
- The chapter “[Configuring ATM](#)” in the *Cisco IOS Wide-Area Networking Configuration Guide*, Release 12.2

RFC 1483 Routing

For information about ATM and ATM adaptation layers (AALs), refer to the following document:

- The chapter “[Wide-Area Networking Overview](#)” in the *Cisco IOS Wide-Area Networking Configuration Guide*, Release 12.2

For information regarding AAL5 Subnetwork Access Protocol (AAL5SNAP) encapsulations, refer to the following document:

- The chapter “[Configuring ATM](#)” in the *Cisco IOS Wide-Area Networking Configuration Guide*, Release 12.2

Benefits

QoS provides improved and more predictable network service for ADSL and G.SHDSL by

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

Restrictions

- G.SHDSL WICs are not supported on Cisco 3600 series routers in the Cisco IOS Release 12.2(4)XL.

**Note**

The G.SHDSL WIC is supported on Cisco 2600 series routers in the Cisco IOS Release 12.2(4)XL.

- Analog and BRI voice on the NM-1V/2V cards are not supported over VoATM in AAL2.

- Refer to the following documents for caveat information for multiple PVCs on Cisco 1700 series, Cisco 2600 series, and Cisco 3600 series routers:
 - [Release Notes for the Cisco 1700 Series Routers for Cisco IOS Release 12.2\(8\)YN](#)
 - [Release Notes for the Cisco 1700 Series Routers for Cisco IOS Release 12.2\(2\)XK](#)
 - [Release Notes for Cisco 2600 Series for Cisco IOS Release 12.2 XK](#)
 - [Release Notes for Cisco 3600 Series for Cisco IOS Release 12.2 XK](#)
 - [Release Notes for Cisco 2600 Series for Cisco IOS Release 12.2 XL](#)
 - [Release Notes for Cisco 3600 Series for Cisco IOS Release 12.2 XL](#)
- F5 OAM CC segment functionality is not currently supported on Cisco DSLAMs.

Related Documents

Table 15 lists related documents about the Voice and QoS for ADSL and G.SHDSL features on Cisco 1700, Cisco 2600, and Cisco 3600 series routers.

Table 15 Related Documents

Related Topic	Document Titles
ADSL WAN interface card	<ul style="list-style-type: none"> • Configuring an ADSL WAN Interface Card on Cisco 1700 Series Routers • 1-Port ADSL WAN Interface Card for Cisco 2600 Series and 3600 Series Routers, Release 12.2(4)T
ATM, configuring	<ul style="list-style-type: none"> • Cisco IOS Wide-Area Networking Configuration Guide, Release 12.2 • Cisco IOS Wide-Area Networking Command Reference, Release 12.2
Caveat information for the Cisco 1700 series, Cisco 2600 series, and Cisco 3600 series routers	<ul style="list-style-type: none"> • Release Notes for the Cisco 1700 Series Routers for Cisco IOS Release 12.2(2)XK • Release Notes for the Cisco 1700 Series Routers for Cisco IOS Release 12.2(8)YN • Release Notes for Cisco 2600 Series for Cisco IOS Release 12.2 XK • Release Notes for Cisco 3600 Series for Cisco IOS Release 12.2 XK • Release Notes for Cisco 2600 Series for Cisco IOS Release 12.2 XL • Release Notes for Cisco 3600 Series for Cisco IOS Release 12.2 XL
Dial-peer DSCPs and IP precedence marking	The chapter “ Quality of Service for Voice over IP ” in the Cisco IOS Quality of Service Solutions

Related Topic	Document Titles
F5 OAM CC segment functionality	<ul style="list-style-type: none"> • “Cisco Product Bulletin No. 1518” about Cisco IOS software Release 12.2(2)XJ • Release Notes for the Cisco 1700 Series Routers for Cisco IOS Release 12.2(XJ)
G.SHDSL WAN interface card	<ul style="list-style-type: none"> • Installing the G.SHDSL WIC on the Cisco 1700 Series Router • 1-Port G.SHDSL WAN Interface Card for Cisco 2600 Series and 3600 Series Routers, Release 12.2(4)XL
ILMI	<ul style="list-style-type: none"> • The chapter “Configuring ILMI” in the Configuration Guide for Cisco DSLAMS with NI-2
IP, configuring	<ul style="list-style-type: none"> • Cisco IOS IP Configuration Guide, Release 12.2 • Cisco IOS IP Command Reference, Release 12.2 (there are three volumes)
Local policy routing	<ul style="list-style-type: none"> • The chapter “Configuring IP Routing Protocols” in Router Products Configuration Guide
Per-VC queueing	<ul style="list-style-type: none"> • Understanding Weighted Fair Queueing on ATM • Per-VC Class-Based, Weighted Fair Queueing (Per-VC CBWFQ) on the Cisco 7200, 3600, and 2600 Routers
QoS, configuring	<ul style="list-style-type: none"> • Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.2 • Cisco IOS Quality of Service Solutions Command Reference, Release 12.2
QoS features on Cisco 1700 series routers	<ul style="list-style-type: none"> • Cisco IOS Software Release 12.2(2)XQ1
VC shaping for VBR-NRT	<ul style="list-style-type: none"> • Understanding the VBR-nrt Service Category and Traffic Shaping for ATM VCs
Voice configuration	<ul style="list-style-type: none"> • Cisco IOS Voice, Video, and Fax Configuration Guide, Release 12.2 • Cisco IOS Voice, Video, and Fax Command Reference, Release 12.2
WRED	<ul style="list-style-type: none"> • DiffServ Compliant Weighted Random Error Detection

Supported Platforms

- Cisco 1720
- Cisco 1721
- Cisco 1751
- Cisco 1751V
- Cisco 1760
- Cisco 1760V
- Cisco 2650
- Cisco 2651
- Cisco 2600XM
- Cisco 3640
- Cisco 3660

Determining Platform Support Through 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 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, refer to the online release notes or, if supported, Cisco Feature Navigator.



Note

As of Cisco IOS Release 12.2(2)XK and Release 12.2(4)XL, Feature Navigator does not support features included in these limited-lifetime releases.

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

No new or modified MIBs 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>

No new or modified RFCs are supported by these features.

RFCs

No new or modified RFCs are supported by this feature.

Prerequisites

To configure the voice and QoS features, you must first install and configure the ADSL or G.SHDSL WIC on your Cisco 1700 series, Cisco 2600 series, or Cisco 3600 series router. Refer to the installation and configuration instructions in the following documents:

- *Configuring an ADSL WAN Interface Card on Cisco 1700 Series Routers*
- *Installing the G.SHDSL ATM WIC on the Cisco 1700 Series Router*
- *1-Port ADSL WAN Interface Card for Cisco 2600 Series and 3600 Series Routers*, Release 12.2(4)T
- *1-Port G.SHDSL WAN Interface Card for Cisco 2600 Series and 3600 Series Routers*, Release 12.2(4)XL

Configuration Tasks

See the following section to configure voice and QoS features over ADSL and G.SHDSL:

- [Configuring the Error Duration for Digital Subscriber Line Access Multiplexers](#) (required)
- [Configuring the Tx Ring Limit](#) (required)
- [Verifying the TX Ring Limit](#)
- [Configuration Examples](#)

Configuring the Error Duration for Digital Subscriber Line Access Multiplexers

To configure the error duration for digital subscriber line access multiplexers (DSLAMs), use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# dsl equipment-type ignore-error duration {seconds}	Configures the DSLAMs. The <i>seconds</i> argument has a valid range from 15 to 30 seconds.

Configuring the Tx Ring Limit

To configure the tx ring limit, use the following commands beginning in global configuration mode:



Note

The Tunable Transmission Ring (tx ring) feature is not supported on Cisco 1700 series routers until Cisco IOS Release 12.2(8)YN.

	Command	Purpose
Step 1	Router (config)# interface atm <i>slot/port</i>	Configures an ATM interface type and enters interface configuration mode. The arguments are as follows: <ul style="list-style-type: none"> • <i>slot</i>—Specifies the backplane slot number on your router. The value ranges from 0 to 4, depending on what router you are configuring. Refer to your router hardware documentation. • <i>/port</i>—ATM port number on a Cisco 2600 or 3600 series router, indicating the T1 link that you are configuring. Enter a value from 0 to 3 or from 0 to 7, depending on whether the network module has four ports or eight ports.

Command	Purpose
Step 2 Router (config-if)# pvc [name] vpi/vci [ces ilmi qsaal smds]	<p>Creates or assigns a name to an ATM permanent virtual circuit (PVC), specifies the encapsulation type on an ATM PVC, and enters interface-ATM-VC configuration mode.</p> <p>The keywords and arguments are as follows:</p> <ul style="list-style-type: none"> • <i>name</i>—(Optional) Specifies the name of the PVC or map. The name can be a maximum of 16 characters. • <i>vpi</i>—Specifies the ATM network virtual path identifier (VPI) for this PVC. The absence of the “/” and a VPI value defaults the VPI value to 0. On Cisco 2600 and 3600 series routers using inverse multiplexing over ATM (IMA), the ranges are 0 to 15, 64 to 79, 128 to 143, and 192 to 207. • <i>vci</i>—Specifies the ATM network virtual channel identifier (VCI) for this PVC. This value ranges from 0 to 1 less than the maximum value set for this interface by the atm vc-per-vp command. Typically, lower values 0 to 31 are reserved for specific traffic (for example, F4 Operation, Administration, and Maintenance [OAM], switched virtual circuit [SVC] signalling, and Integrated Local Management Interface [ILMI] and should not be used). The VCI is a 16-bit field in the header of the ATM cell. The VCI value is unique only on a single link, not throughout the ATM network, because it has local significance only. The <i>vpi</i> and <i>vci</i> arguments cannot both be set to 0; if one is 0, the other cannot be 0. • ces—(Optional) circuit emulation service encapsulation. This keyword is available on the OC-3/STM-1 ATM Circuit Emulation Service network module only. • ilmi—(Optional) Used to set up communication with the ILMI; the associated <i>vpi</i> and <i>vci</i> values ordinarily are 0 and 16, respectively. • qsaal—(Optional) A signalling-type PVC used for setting up or tearing down SVCs; the associated <i>vpi</i> and <i>vci</i> values ordinarily are 0 and 5, respectively. • smds—(Optional) Specifies encapsulation for Switched Multimegabit Data Service (SMDS) networks. If you are configuring an ATM PVC on the ATM Interface Processor (AIP), you must configure AAL3/4SMDS using the atm aal aal3/4 command before specifying SMDS encapsulation. If you are configuring an ATM network processor module (NPM), the atm aal aal3/4 command is not required. SMDS encapsulation is not supported on the ATM port adapter.

Command	Purpose
Step 3 Router (config-if-atm-vc)# tx-ring-limit <i>ring-limit</i>	Limits the number of packets that can be used on a transmission ring on the permanent virtual circuit (PVC). The arguments are as follows: <ul style="list-style-type: none"> • <i>ring-limit</i>—The maximum number of allowable packets that can be placed on the transmission ring. The default value is 60. On Cisco 2600 and Cisco 3600 series routers, the value can be changed to 2 or 3. (The only permitted values are 2, 3, and 60.)

Verifying the TX Ring Limit

The following output examples are for a tx ring limit over ADSL configuration:

The following **show policy-map interface** command output is for a tx ring-limit tuning configuration on a Cisco 2600 router. The **show policy-map interface** command displays the policy-map setup.

```
Router# show policy-map interface atm 0/1.1

ATM0/1.1:VC 11/201 -

Service-policy output:SERVICE-PACK-640

Class-map:VOICE-CLASS (match-all)
 5295 packets, 402420 bytes
 30 second offered rate 30000 bps, drop rate 0 bps
Match:access-group 100
Weighted Fair Queueing
  Strict Priority
  Output Queue:Conversation 72
  Bandwidth 160 (kbps) Burst 4000 (Bytes)
  (pkts matched/bytes matched) 5295/402420
  (total drops/bytes drops) 0/0

Class-map:class-default (match-any)
 42365 packets, 63625280 bytes
 30 second offered rate 4675000 bps, drop rate 4069000 bps
Match:any
```

The following **show interfaces atm** command output is for a tx ring-limit tuning configuration on a Cisco 2600 router. The **show interfaces atm** command displays statistics for the ATM interface.

```
Router# show interfaces atm0/1

ATM0/1 is up, line protocol is up
Hardware is DSL SAR (with Globespan G.SHDSL module)
MTU 4470 bytes, sub MTU 4470, BW 2304 Kbit, DLY 880 usec,
  reliability 255/255, txload 59/255, rxload 1/255
Encapsulation ATM, loopback not set
Encapsulation(s):AAL5, PVC mode
23 maximum active VCs, 256 VCs per VP, 1 current VCCs
VC idle disconnect time:300 seconds
Last input 00:01:35, output 00:00:00, output hang never
Last clearing of "show interface" counters never
queue:0/75/0/0 (size/max/drops/flushes); Total output drops:66321
Queueing strategy:None
second input rate 0 bits/sec, 0 packets/sec
30 second output rate 541000 bits/sec, 93 packets/sec
227 packets input, 5355 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
```

```

15 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
517351 packets output, 133910269 bytes, 0 underruns
93 output errors, 0 collisions, 2 interface resets
0 output buffer failures, 0 output buffers swapped out

```

The following **show queue** command output for a tx ring-limit tuning configuration on a Cisco 2600 router displays the flow of traffic that is currently being transported over the ATM interface:

```

Router# show queue atm 0/1

Interface ATM0/1 VC 11/201
Queueing strategy:weighted fair
Output queue:65/512/64/52265 (size/max total/threshold/drops)
Conversations 2/3/64 (active/max active/max total)
Reserved Conversations 0/0 (allocated/max allocated)
Available Bandwidth 320 kilobits/sec

(depth/weight/total drops/no-buffer drops/interleaves) 1/0/0/0/0
Conversation 72, linktype:ip, length:76
source:10.1.1.204, destination:10.10.11.254, id:0x0000, ttl:59,
TOS:160 prot:17, source port 400, destination port 400

(depth/weight/total drops/no-buffer drops/interleaves) 64/32384/52267/0/0
Conversation 49, linktype:ip, length:1502
source:10.1.1.205, destination:10.10.11.254, id:0x0000, ttl:59,
TOS:0 prot:17, source port 500, destination port 500

```

The following **show atm vc** command outputs are for a tx ring-limit tuning configuration on a Cisco 2600 router. The **show atm vc** command displays all ATM PVCs and SVCs and traffic information.

```

Router# show atm vc

VC not configured on interface ATM0/0

          VCD /
Interface Name  VPI  VCI  Type  Encaps  SC      Kbps  Peak  Avg/Min  Burst
0/1.1          1    11   201  PVC    SNAP   VBR    640   64    0        UP

Router# show atm vc 1

VC 1 doesn't exist on interface ATM0/0
ATM0/1.1:VCD:1, VPI:11, VCI:201
VBR-NRT, PeakRate:640, Average Rate:640, Burst Cells:0
AAL5-LLC/SNAP, etype:0x0, Flags:0x2000020, VCmode:0x0
OAM frequency:10 second(s)
InARP frequency:15 minutes(s)
InPkts:5, OutPkts:14707, InBytes:560, OutBytes:10698804
InPRoc:5, OutPRoc:5
InFast:0, OutFast:3, InAS:0, OutAS:0
InPktDrops:0, OutPktDrops:56701/0/56701 (holdq/outputq/total)
CrcErrors:0, SarTimeOuts:0, OverSizedSDUs:0, LengthViolation:0, CPICErrors:0
OAM cells received:46
OAM cells sent:51
Status:UP

```

Configuration Examples

This section provides the following configuration examples:

- [Differentiated Data Services over ADSL Example](#)
- [VoIP and Data over ADSL Example](#)
- [Tx Ring-Limit Tuning over ADSL Example](#)
- [MLP with LFI over G.SHDSL Example](#)

Differentiated Data Services over ADSL Example

The following is from a Cisco 1751 router. The output displays that the router is configured for differentiated services:

```
access-list 102 permit udp host 16.0.0.4 host 15.0.0.5
access-list 103 permit udp host 16.0.0.4 host 13.0.0.5
ip cef
class-map match-all traffic-INTRA
  match access-group 102
class-map match-all traffic-INTER
  match access-group 103
class-map match-all traffic-dscp1
  match ip dscp 1
class-map match-any traffic-prec3
  match ip dscp 24
  match ip dscp 25
  match ip dscp 26
  match ip dscp 27
policy-map ADSL-out
  class traffic-INTRA
    bandwidth percent 8
  class traffic-dscp1
    set ip dscp 5
  class traffic-prec3
    set ip precedence 2
  class traffic-INTER
    bandwidth percent 8
  class class-default
    fair-queue
!
interface ATM0/0
  no ip address
  no atm ilmi-keepalive
!
interface ATM0/0.1 point-to-point
  description COLLEGAMENTO
  mtu 576
  ip address 1.0.0.1 255.0.0.0
  pvc 99/99
    protocol ip 2.0.0.2 broadcast
    vbr-nrt 142 142 1
    oam-pvc 0
    oam retry 5 5 1
    encapsulation aal5snap
    service-policy out ADSL-out
!
dial-peer voice 201 voip
  destination-pattern 3640200
  session target ipv4:14.0.0.3
```

```

plout-delay maximum 300
ip qos dscp cs4 media
ip qos dscp cs4 signaling

```

The following is from a Cisco 2600 router. The output displays how CBWFQ, CAR, and WRED can be applied in the same configuration to provide differentiated services using QoS:

Building configuration...

```

Current configuration: 2603 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname 2600-1
!
memory-size iomem 20
!
ip subnet-zero
!
ip cef
!
class-map match-all GOLD
  match access-group 1
class-map match-all SILVER
  match access-group 2
!
policy-map GOLD-160-DATA-PACK-640
  class GOLD
    bandwidth 160
    random-detect dscp-based
    random-detect exponential-weighting-constant 3
    random-detect dscp 16 2 4 10
    random-detect dscp 32 4 12 10
  class SILVER
    bandwidth 320
    random-detect dscp-based
    random-detect exponential-weighting-constant 3
    random-detect dscp 0 30 60 10
    set ip dscp 0
!
interface ATM0/0
  no ip address
  load-interval 30
  atm vc-per-vp 256
  no atm ilmi-keepalive
  dsl operating-mode auto
!
interface ATM0/0.1 point-to-point
  ip address 192.168.1.2 255.255.255.0
  pvc 0/201
    protocol ip 192.168.1.1 broadcast
    vbr-nrt 640 640
    tx-ring-limit 3
  service-policy output GOLD-160-DATA-PACK-640
!
interface Ethernet0/0
  ip address 1.3.214.9 255.255.0.0
  half-duplex
!
interface Ethernet0/1

```

```

ip address 10.1.1.1 255.255.255.0
rate-limit input access-group 1 160000 4470 4470 conform-action set-dscp-transmit 32
exceed-action continue
rate-limit input access-group 1 80000 4470 4470 conform-action set-dscp-transmit 16
exceed-action drop
half-duplex
!
ip classless
ip route 10.10.11.200 255.255.255.255 192.168.3.1
ip route 223.255.254.254 255.255.255.255 1.3.0.1
ip http server
!
access-list 1 permit 10.1.1.201
access-list 2 permit 10.1.1.202
access-list 3 permit 10.1.1.203
access-list 100 permit udp any any precedence critical
!
snmp-server manager
call rsvp-sync
alias exec s sh run
alias exec c conf t
!
line con 0
exec-timeout 0 0
privilege level 15
line aux 0
line vty 0 4
login
line vty 5 15
login
!

```

Verifying the Differentiated Data Services over ADSL Configuration

The following **show policy-map interface** command output is for the CBWFQ, CAR, and WRED configuration. The **show policy-map interface** command displays the policy-map setup.

```
Router# show policy-map interface atm0/0.1
```

```
ATM0/0.1:VC 0/201 -
```

```
Service-policy output:GOLD-160-DATA-PACK-640
```

```
Class-map:GOLD (match-all)
22738 packets, 34379856 bytes
30 second offered rate 239000 bps, drop rate 50000 bps
Match:access-group 1
Weighted Fair Queueing
Output Queue:Conversation 73
Bandwidth 160 (kbps)
(pkts matched/bytes matched) 22738/34379856
(depth/total drops/no-buffer drops) 4/4739/0
exponential weight:3
mean queue depth:4
```

dscp	Random drop pkts/bytes	Tail drop pkts/bytes	Minimum threshold	Maximum threshold	Mark probability
af11	0/0	0/0	32	40	1/10
af12	0/0	0/0	28	40	1/10
af13	0/0	0/0	24	40	1/10
af21	0/0	0/0	32	40	1/0
af22	0/0	0/0	28	40	1/10

af23	0/0	0/0	24	40	1/10
af31	0/0	0/0	32	40	1/10
af32	0/0	0/0	28	40	1/10
af33	0/0	0/0	24	40	1/10
af41	0/0	0/0	32	40	1/10
af42	0/0	0/0	28	40	1/10
af43	0/0	0/0	24	40	1/10
cs1	0/0	0/0	22	40	1/10
cs2	332/501984	3907/5907384	2	4	1/10
cs3	0/	0/0	26	40	1/10
cs4	506/765072	0/0	4	12	1/10
cs5	0/0	0/0	30	40	1/10
cs6	0/0	0/0	32	40	1/10
cs7	0/0	0/0	34	40	1/10
ef	0/0	0/0	36	40	1/10
rsvp	0/0	0/0	36	40	1/10
default	0/0	0/0	20	40	1/10

```

Class-map:SILVER (match-all)
  114748 packets, 173498976 bytes
  30 second offered rate 1212000 bps, drop rate 832000 bps
Match:access-group 2
Weighted Fair Queueing
  Output Queue:Conversation 74
  Bandwidth 320 (kbps)
  (pkts matched/bytes matched) 115126/174070512
  (depth/total drops/no-buffer drops) 61/79012/0
  exponential weight:3
  mean queue depth:61

```

dscp	Random drop pkts/bytes	Tail drop pkts/bytes	Minimum threshold	Maximum threshold	Mark probability
af11	0/0	0/0	32	40	1/10
af12	0/0	0/0	28	40	1/10
af13	0/0	0/0	24	40	1/10
af21	0/0	0/0	32	40	1/10
af22	0/0	0/0	28	40	1/10
af23	0/0	0/0	24	40	1/10
af31	0/0	0/0	32	40	1/10
af32	0/0	0/0	28	40	1/10
af33	0/0	0/0	24	40	1/10
af41	0/0	0/0	32	40	1/10
af42	0/0	0/0	28	40	1/10
af43	0/0	0/0	24	40	1/10
cs1	0/0	0/0	22	40	1/10
cs2	0/0	0/0	24	40	1/10
cs3	0/0	0/0	26	40	1/10
cs4	0/0	0/0	28	40	1/10
cs5	0/0	0/0	30	40	1/10
cs6	0/0	0/0	32	40	1/10
cs7	0/0	0/0	34	40	1/10
ef	0/0	0/0	36	40	1/10
rsvp	0/0	0/0	36	60	1/10
default	9096/13753152	70065/105938280	30	60	1/10

```

QoS Set
  ip dscp 0
  Packets marked 115344

```

```

Class-map:class-default (match-any)
  114747 packets, 173497464 bytes
  30 second offered rate 1212000 bps, drop rate 1209000 bps
Match:any

```

The following **show interfaces** command is from a Cisco 2600 router. The **show interfaces** command displays statistics for all interfaces configured on the router.

```
Router# show interfaces e0/1 rate-limit

Ethernet0/1
  Input
    matches:access-group 1
      params: 160000 bps, 4470 limit, 4470 extended limit
      conformed 15673 packets, 23728922 bytes; action:set-dscp-transmit 32
      exceeded 102965 packets, 155889010 bytes; action:continue
      last packet:0ms ago, current burst:4146 bytes
      last cleared 00:19:46 ago, conformed 160000 bps, exceeded 1051000 bps
    matches:access-group 1
      params: 80000 bps, 4470 limit, 4470 extended limit
      conformed 7836 packets, 11863704 bytes; action:set-dscp-transmit 16
      exceeded 95130 packets, 144026820 bytes; action:drop
      last packet:4ms ago, current burst:3708 bytes
      last cleared 00:19:46 ago, conformed 79000 bps, exceeded 971000 bps
```

The following **show interfaces atm** command output is from a Cisco 2600 router. The **show interfaces atm** command displays information about the ATM interface.

```
Router# show interfaces atm 0/0

ATM0/0 is up, line protocol is up
Hardware is DSLSAR (with Alcatel ADSL Module)
MTU 4470 bytes, sub MTU 4470, BW 800 Kbit, DLY 2560 usec,
  reliability 255/255, txload 181/255, rxload 1/255
Encapsulation ATM, loopback not set
Encapsulation(s):AAL5 AAL2, PVC mode
23 maximum active VCs, 256 VCs per VP, 2 current VCCs
VC idle disconnect time:300 seconds
Last input 00:33:22, output 00:00:00, output hang never
Last clearing of "show interface" counters 00:20:09
Input queue:0/75/0/0 (size/max/drops/flushes); Total output drops:208908
Queueing strategy:None
30 second input rate 0 bits/sec, 0 packets/sec
30 second output rate 569000 bits/sec, 48 packets/sec
  0 packets input, 0 bytes, 0 no buffer
  Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  57315 packets output, 86075268 bytes, 0 underruns
  0 output errors, 0 collisions, 0 interface resets
  0 output buffer failures, 0 output buffers swapped out
```

The following **show queue** command output from a Cisco 2600 router displays the flow of traffic that is currently being transported over the ATM interface:

```
Router# show queue atm0/0

Interface ATM0/0 VC 0/201
Queueing strategy:weighted fair
Output queue:130/512/64/214301 (size/max total/threshold/drops)
Conversations 3/3/64 (active/max active/max total)
Reserved Conversations 2/2 (allocated/max allocated)
Available Bandwidth 0 kilobits/sec

(depth/weight/total drops/no-buffer drops/random/tail/interleaves) 5/228/5124/0/0/0/0
Conversation 73, linktype:ip, length:1512
source:10.1.1.201, destination:10.10.11.200, id:0x0000, ttl:59,
TOS:128 prot:17, source port 100, destination port 100
```

```
(depth/weight/total drops/no-buffer drops/random/tail/interleaves)
61/114/85189/0/9843/75346/0
Conversation 74, linktype:ip, length:1512
source:10.1.1.202, destination:10.10.11.200, id:0x0000, ttl:59,
TOS:0 prot:17, source port 200, destination port 200

(depth/weight/total drops/no-buffer drops/interleaves) 64/32384/123990/0/0
Conversation 41, linktype:ip, length:1512
source:10.1.1.203, destination:10.10.11.200, id:0x0000, ttl:59,
TOS:0 prot:17, source port 300, destination port 300
```

VoIP and Data over ADSL Example

The following is sample output from a Cisco 2611 router. In this example, the customer premises equipment (CPE) is restricted to only a single PVC. Voice and data are sent over a single VC. The **tx-ring-limit** command and LLQ are used to give preferential treatment for voice traffic.

Building configuration...

```
Current configuration :1861 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname CPE-2611-1
!
voice-card 1
 dspfarm
!
ip subnet-zero
!
ip cef
!
class-map match-all VOICE
 match access-group 100
!
policy-map VOICE-160-DATA-480-PACK
 class VOICE
  priority 160
 class class-default
  bandwidth 320
!
controller T1 1/0
 framing esf
 linecode b8zs
 ds0-group 0 timeslots 1-24 type e&m-wink-start
!
controller T1 1/1
 framing sf
 linecode ami
!
interface ATM0/0
 no ip address
 load-interval 30
 atm vc-per-vp 256
 no atm ilmi-keepalive
 dsl operating-mode auto
!
interface ATM0/0.1 point-to-point
```

```

ip address 192.168.1.2 255.255.255.0
pvc 0/201
  protocol ip 192.168.1.1 broadcast
  vbr-nrt 640 640
  tx-ring-limit 3
  service-policy output VOICE-160-DATA-480-PACK
!
interface Ethernet0/0
ip address 1.3.214.51 255.255.0.0
half-duplex
!
interface ATM0/1
no ip address
shutdown
atm vc-per-vp 256
no atm ilmi-keepalive
atm voice aal2 aggregate-svc upspeed-number 0
dsl equipment-type CPE
dsl operating-mode GSHDSL symmetric annex A
dsl linerate AUTO
!
interface Ethernet0/1
ip address 10.1.1.1 255.255.255.0
half-duplex
!
ip classless
ip route 223.255.254.254 255.255.255.255 1.3.0.1
ip http server
!
access-list 100 permit udp any any precedence critical
!
snmp-server manager
call rsvp-sync
!
voice-port 1/0:0
!
mgcp profile default
!
dial-peer cor custom
!
dial-peer voice 1 pots
  destination-pattern 7...
  port 1/0:0
!
dial-peer voice 2 voip
  destination-pattern 8...
  session target ipv4:192.168.3.1
  ip qos dscp cs5 media
  ip qos dscp cs5 signaling
  no vad
!
alias exec s sh run
alias exec c conf t
!
line con 0
  exec-timeout 0 0
  privilege level 15
line aux 0
line vty 0 4
  login
line vty 5 15
  login

```

Verifying the VoIP and Data over ADSL Configuration

The following **show policy-map interface** command output from a Cisco 2600 router displays the service-policy setup:

```
Router# show policy-map interface atm0/0

ATM0/0:VC 0/201 -

Service-policy output:VOICE-160-DATA-480

Class-map:class-default (match-any)
  27234 packets, 41109865 bytes
  30 second offered rate 7000 bps, drop rate 3000 bps
Match:any
Weighted Fair Queueing
Queue:Conversation 73
Bandwidth 320 (kbps) Max Threshold 64 (packets)
(pkts matched/bytes matched) 27231/41105329
(depth/total drops/no-buffer drops) 0/14711/0

For
Class-map:class-default (match-any)
  113187 packets, 140760375 bytes
  30 second offered rate 1205000 bps, drop rate 787000 bps
Match:any
```

The following **show queue** command output from a Cisco 2600 router displays the flow of traffic that is currently being transported over an ATM interface:

```
Router# show queue atm 0/0

Interface ATM0/0 VC 0/201
Queueing strategy:weighted fair
Output queue:70/512/64/70462 (size/max total/threshold/drops)
Conversations 3/6/64 (active/max active/max total)
Reserved Conversations 0/0 (allocated/max allocated)
Available Bandwidth 320 kilobits/sec

(depth/weight/total drops/no-buffer drops/interleaves) 3/0/0/0/0
Conversation 72, linktype:ip, length:72
source:192.168.1.2, destination:192.168.1.1, id:0xC77E, ttl:254,
TOS:160 prot:17, source port 19406, destination port 16406

(depth/weight/total drops/no-buffer drops/interleaves) 1/32384/0/0/0
Conversation 23, linktype:ip, length:196
source:192.168.1.2, destination:192.168.1.1, id:0x0000, ttl:255,
TOS:0 prot:17, source port 18653, destination port 18691

(depth/weight/total drops/no-buffer drops/interleaves) 64/32384/65793/0/0
Conversation 59, linktype:ip, length:1502
source:10.1.1.205, destination:10.10.11.200, id:0x0000, ttl:59,
TOS:0 prot:17, source port 500, destination port 500
```

The following **show queueing interface** command output from a Cisco 2600 router displays the queueing configuration of the ATM interface:

```
Router# show queueing interface atm0/0

Interface ATM0/0 VC 0/201
Queueing strategy:weighted fair
Output queue:66/512/64/61642 (size/max total/threshold/drops)
Conversations 2/6/64 (active/max active/max total)
```

```
Reserved Conversations 0/0 (allocated/max allocated)
Available Bandwidth 320 kilobits/sec
```

The following **show interfaces atm** command output from a Cisco 2600 router displays statistics for the ATM interface:

```
Router# show interfaces atm0/0

ATM0/0 is up, line protocol is up
Hardware is DSLSAR (with Alcatel ADSL Module)
Internet address is 192.168.1.2/24
MTU 4470 bytes, sub MTU 4470, BW 800 Kbit, DLY 2560 usec,
  reliability 255/255, txload 166/255, rxload 21/255
Encapsulation ATM, loopback not set
Encapsulation(s):AAL5 AAL2, PVC mode
23 maximum active VCs, 256 VCs per VP, 2 current VCCs
VC idle disconnect time:300 seconds
Last input 00:00:02, output 00:00:00, output hang never
Last clearing of "show interface" counters never
Input queue:0/75/0/0 (size/max/drops/flushes); Total output drops:62360
Queueing strategy:None
30 second input rate 66000 bits/sec, 113 packets/sec
30 second output rate 523000 bits/sec, 229 packets/sec
1603630 packets input, 403845485 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
2680554 packets output, 518308502 bytes, 0 underruns
0 output errors, 0 collisions, 4 interface resets
buffer failures, 0 output buffers swapped out
```

The following **show atm vc** command output from a Cisco 2600 router displays information about the ATM virtual circuit:

```
Router# show atm vc

          VCD /
Interface Name      VPI VCI  Type  Encaps  SC   Kbps  Peak  Avg/Min  Burst
0/0        1          0  201  PVC   SNAP   VBR   640   640    0       UP
VC not configured on interface ATM0/1
```

The following **show atm vc** command output from a Cisco 2600 router displays detailed information about the virtual circuit descriptor (VCD):

```
Router# show atm vc 1

ATM0/0:VCD:1, VPI:0, VCI:201
VBR-NRT, PeakRate:640, Average Rate:640, Burst Cells:0
AAL5-LLC/SNAP, etype:0x0, Flags:0x2000820, VCmode:0x0
OAM frequency:10 second(s)
InARP frequency:15 minutes(s)
InPkts:1606861, OutPkts:2682709, InBytes:404080341, OutBytes:519701804
InPRoc:108900, OutPRoc:109133, Broadcasts:1
InFast:1497961, OutFast:0, InAS:0, OutAS:0
InPktDrops:0, OutPktDrops:64078/0/64078 (holdq/outputq/total)
CrcErrors:0, SarTimeOuts:0, OverSizedSDUs:0, LengthViolation:0, CPIErrors:0
OAM cells received:4765
OAM cells sent:4767
Status:UP
VC 1 doesn't exist on interface ATM0/1
```

Tx Ring-Limit Tuning over ADSL Example

The following is output from a Cisco 2600 router. The output shows that tx ring-limit tuning is configured:



Note

The Tunable Transmission Ring (tx ring) feature is not supported on Cisco 1700 series routers until Cisco IOS Release 12.2(8)YN.

```
Building configuration...

Current configuration :2018 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname CPE-2600-1
!
voice-card 1
  dspfarm
!
ip subnet-zero
!
ip cef
!
class-map match-all VOICE-CLASS
  match access-group 100
!
policy-map SERVICE-PACK-640
  class VOICE-CLASS
    priority 160
!
controller T1 1/0
  framing esf
  linecode b8zs
  ds0-group 0 timeslots 1-24 type e&m-wink-start
!
controller T1 1/1
  framing sf
  linecode ami
!
interface FastEthernet0/0
  ip address 1.3.214.50 255.255.0.0
  duplex auto
  speed auto
!
interface ATM0/1
  no ip address
  load-interval 30
  atm vc-per-vp 256
  no atm ilmi-keepalive
  atm voice aal2 aggregate-svc upspeed-number 0
  dsl equipment-type CPE
  dsl operating-mode GSHDSL symmetric annex A
  dsl linerate AUTO
!
interface ATM0/1.1 point-to-point
  ip address 192.168.1.2 255.255.255.0
  pvc 11/201
```

```

protocol ip 192.168.1.1 broadcast
vbr-nrt 640 640
tx-ring-limit 3
oam-pvc manage
service-policy output SERVICE-PACK-640
!
interface FastEthernet0/1
ip address 10.10.11.1 255.255.255.0
load-interval 30
duplex auto
speed auto
!
ip classless
ip route 10.10.11.254 255.255.255.255 192.168.1.1
ip route 223.255.254.254 255.255.255.255 1.3.0.1
ip http server
ip pim bidir-enable
!
ip director cache time 60
access-list 100 permit udp any any precedence critical
!
snmp-server manager
call rsvp-sync
!
voice-port 1/0:0
!
mgcp profile default
!
dial-peer cor custom
!
dial-peer voice 1 pots
destination-pattern 7...
!
dial-peer voice 2 voip
destination-pattern 8...
session target ipv4:192.168.1.1
ip qos dscp cs5 media
ip qos dscp cs5 signaling
no vad
!
alias exec s sh run
alias exec c conf t
!
line con 0
exec-timeout 0 0
privilege level 15
line aux 0
line vty 0 4
login
line vty 5 15
login
!

```

MLP with LFI over G.SHDSL Example

The following output is from a Cisco 1751 router. The output shows that MLP with LFI is configured for G.SHDSL:

```

class-map match-all VOIP
 match ip dscp 32
class-map CRITICAL
 match access-group 100

```



```

!
policy-map 1751_DSL
  class CRITICAL
    priority 48
  class VOIP
    priority 64
    set ip precedence 6
!
interface ATM0/0
  no ip address
  no atm ilmi-keepalive
!
interface ATM0/0.1 point-to-point
  pvc 0/33
    vbr-rt 150 150 30
    protocol ppp Virtual-Templatel
!
interface Loopback1
  ip address 10.0.0.10 255.255.255.255
interface Virtual-Templatel
  bandwidth 320
  ip unnumbered Loopback1
  service-policy output 1751_DSL
  ppp multilink
  ppp multilink fragment-delay 4
  ppp multilink interleave
!
access-list 100 permit udp any any precedence critical
dial-peer voice 201 voip
  destination-pattern 3640200
  session target ipv4:10.0.0.11
  ip qos dscp cs4 media
  ip qos dscp cs4 signalling

```

The following output is from a Cisco 2600 router. The output shows that MLP with LFI is configured for G.SHDSL:

```

Building configuration...

Current configuration :2107 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname CPE-2600-1
!
memory-size iomem 20
voice-card 1
  dspfarm
!
ip subnet-zero
!
ip cef
!
class-map match-all VOICE-CLASS
  match access-group 100
!
policy-map SERVICE-PACK-640
  class VOICE-CLASS
    priority 160
!
controller T1 1/0

```

```

framing esf
linecode b8zs
ds0-group 0 timeslots 1-24 type e&m-wink-start
!
controller T1 1/1
framing sf
linecode ami
!
interface FastEthernet0/0
ip address 1.3.214.50 255.255.0.0
duplex auto
speed auto
!
interface ATM0/1
no ip address
load-interval 30
atm vc-per-vp 256
no atm ilmi-keepalive
atm voice aal2 aggregate-svc upspeed-number 0
dsl equipment-type CPE
dsl operating-mode GSHDSL symmetric annex A
dsl linerate AUTO
!
interface ATM0/1.1 point-to-point
pvc 11/201
vbr-nrt 640 640
tx-ring-limit 3
oam-pvc manage
protocol ppp Virtual-Template1
!
interface FastEthernet0/1
ip address 10.10.11.1 255.255.255.0
load-interval 30
duplex auto
speed auto
!
interface Virtual-Template1
bandwidth 640
ip address 192.168.1.2 255.255.255.0
load-interval 30
service-policy output SERVICE-PACK-640
ppp multilink
ppp multilink fragment-delay 4
ppp multilink interleave
!
ip classless
ip route 10.10.11.254 255.255.255.255 192.168.1.1
ip route 223.255.254.254 255.255.255.255 1.3.0.1
ip http server
ip pim bidir-enable
!
ip director cache time 60
access-list 100 permit udp any any precedence critical
!
snmp-server manager
call rsvp-sync
!
voice-port 1/0:0
!
mgcp profile default
!
dial-peer cor custom
!
destination-pattern 7...

```

```

!
dial-peer voice 2 voip
 destination-pattern 8...
 session target ipv4:192.168.1.1
 ip qos dscp cs5 media
 ip qos dscp cs5 signaling
 no vad
!
alias exec s sh run
alias exec c conf t
!
line con 0
 exec-timeout 0 0
 privilege level 15
line aux 0
line vty 0 4
 login
line vty 5 15
 login

```

Verifying the MLP with LFI over G.SHDSL Configuration

The following **show users** command output is for the MLP with LFI configuration on a Cisco 2600 router. The **show users** command displays information about the active lines on the router.

```
Router# show users
```

Line	User	Host(s)	Idle	Location
* 0 con 0		idle	00:00:00	

Interface	User	Mode	Idle	Peer Address
Vi1		Virtual PPP (ATM)	-	
Vi2		Virtual PPP (Bundle)	00:14:06	192.168.1.1

The following **show interfaces** command output is for the MLP with LFI configuration on a Cisco 2600 router. The **show interfaces** command displays statistics for all interfaces configured on the router.

```
Router# show interfaces virtual-access 2
```

```

Virtual-Access2 is up, line protocol is up
Hardware is Virtual Access interface
Internet address is 192.168.1.2/24
MTU 1500 bytes, BW 640 Kbit, DLY 100000 usec,
 reliability 255/255, txload 207/255, rxload 1/255
Encapsulation PPP, loopback not set
DTR is pulsed for 5 seconds on reset
LCP Open, multilink Open
Open:IPCP
Last input 00:14:20, output never, output hang never
Last clearing of "show interface" counters 00:26:31
Input queue:0/75/0/0 (size/max/drops/flushes); Total output drops:477969
Queueing strategy:weighted fair
Output queue:64/1000/64/477969 (size/max total/threshold/drops)
Conversations 1/2/256 (active/max active/max total)
Reserved Conversations 0/0 (allocated/max allocated)
Available Bandwidth 320 kilobits/sec
30 second input rate 0 bits/sec, 0 packets/sec
30 second output rate 522000 bits/sec, 91 packets/sec
12 packets input, 911 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
136585 packets output, 97029054 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets

```

```
0 output buffer failures, 0 output buffers swapped out
0 carrier transitions
```

The following **show policy-map interface** command is for the MLP with LFI configuration on a Cisco 2600 route. The **show policy-map interface** command displays the policy-map setup.

```
Router# show policy-map interface virtual-access 2

Virtual-Access2

Service-policy output:SERVICE-PACK-640

Class-map:VOICE-CLASS (match-all)
 30887 packets, 2100316 bytes
 30 second offered rate 27000 bps, drop rate 0 bps
 Match:access-group 100
 Weighted Fair Queueing
   Strict Priority
   Output Queue:Conversation 264
   Bandwidth 160 (kbps) Burst 4000 (Bytes)
   (pkts matched/bytes matched) 30888/2100384
   (total drops/bytes drops) 0/0

Class-map:class-default (match-any)
 247101 packets, 369168894 bytes
 30 second offered rate 4785000 bps, drop rate 4283000 bps
 Match:any
```

The following **show ppp multilink** command output is for the MLP with LFI configuration on a Cisco 2600 router. The **show ppp multilink** command displays bundle information for the Multilink PPP bundles.

```
Router# show ppp multilink

Virtual-Access2, bundle name is green-gateway-3660
Bundle up for 00:26:05
0 lost fragments, 0 reordered, 0 unassigned
0 discarded, 0 lost received, 215/255 load
0xC received sequence, 0x55914 sent sequence
Member links:1 (max not set, min not set)
Virtual-Access1, since 00:26:05, last rcvd seq 00000B 320 weight
```

The following **show interfaces atm** command output is for the MLP with LFI configuration. The **show interfaces atm** command displays information about the ATM interface.

```
Router# show interfaces atm 0/1

ATM0/1 is up, line protocol is up
Hardware is DSLSAR (with Globespan G.SHDSL module)
MTU 4470 bytes, sub MTU 4470, BW 2304 Kbit, DLY 880 usec,
 reliability 255/255, txload 59/255, rxload 1/255
Encapsulation ATM, loopback not set
Encapsulation(s):AAL5 , PVC mode
23 maximum active VCs, 256 VCs per VP, 1 current VCCs
VC idle disconnect time:300 seconds
Last input never, output 00:00:00, output hang never
Last clearing of "show interface" counters never
Input queue:0/75/0/0 (size/max/drops/flushes); Total output drops:75
Queueing strategy:None
30 second input rate 0 bits/sec, 0 packets/sec
30 second output rate 539000 bits/sec, 257 packets/sec
186 packets input, 3409 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
```

```
433482 packets output, 103985075 bytes, 0 underruns
0 output errors, 0 collisions, 2 interface resets
0 output buffer failures, 0 output buffers swapped out
```

Command Reference

The following modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at <http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm>.

Modified

- **dsl equipment-type**
- **tx-ring-limit**



Implementing DiffServ for End-to-End Quality of Service Overview

About Differentiated Services

Differentiated Services (DiffServ) 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 services, Integrated Services (IntServ), and Differentiated Services. For more information about the best-effort services and IntServ, see the chapter [Quality of Service Overview](#) in this book.

Differentiated Services is a multiple service model that can satisfy differing QoS requirements. With Differentiated Services, 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 6-bit differentiated services code point (DSCP) setting 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 queueing.

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

Cisco IOS QoS includes the following features that support Differentiated Services:

- Committed access rate (CAR), which performs packet classification through IP Precedence and QoS group settings. CAR performs metering and class-based policing of traffic, providing bandwidth management.
- Intelligent queueing schemes such as Weighted Random Early Detection (WRED) and weighted fair queueing (WFQ) and their equivalent features on the Versatile Interface Processor (VIP), which are VIP-distributed WRED (DWRED) and VIP-distributed WFQ. These features can be used with CAR for implementing Differentiated Services.
- Modular QoS Command-Line Interface (Modular QoS CLI), which provides a command-line interface (CLI) used to configure class-based QoS features.
- Low latency queueing (LLQ), which brings strict priority queueing (PQ) to class-based WFQ (CBWFQ). Strict PQ allows delay-sensitive data such as voice to be dequeued and sent before packets in other queues are dequeued.

- Generic Traffic Shaping (GTS) shapes traffic by reducing outbound traffic flow to avoid congestion by constraining traffic to a particular bit rate using the token bucket mechanism. GTS applies on a per-interface basis and can use access lists to select the traffic to shape.
- Class-Based Shaping configures GTS on a traffic class, specify average rate or peak rate traffic shaping, and configure CBWFQ inside GTS.

For more information about Cisco IOS QoS features, see the chapter [Quality of Service Overview](#) in this book.

This feature supports the Differentiated Services, Assured Forwarding, and Expedited Forwarding standards.

It also 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*
- RFC 2697, *A Single Rate Three Color Marker*

For more information about the specific components and features related to DiffServ, see the sections “[Differentiated Services Components](#)” and “[Feature Sets](#)” sections later in this chapter.

DS Field Definition

A replacement header field, called the DS field, is defined by Differentiated Services. The DS field supersedes the existing definitions of the IP version 4 (IPv4) type of service (ToS) octet (RFC 791) and the IPv6 traffic class octet. Six bits of the DS field are used as the DSCP to select the Per Hop Behavior (PHB) at each interface. A currently unused (CU) 2-bit field is reserved for explicit congestion notification (ECN). The value of the CU bits is ignored by DS-compliant interfaces when determining the PHB to apply to a received packet.

Per-Hop Behaviors

RFC 2475 defines PHB as the externally observable forwarding behavior applied at a DiffServ-compliant node to a DiffServ Behavior Aggregate (BA).

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

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

The following sections describe the four available standard PHBs:

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

Default PHB

The default PHB essentially specifies that a packet marked with a DSCP value of 000000 (recommended) receives the traditional best-effort service from a DS-compliant node (that is, a network node that complies with all of the core DiffServ requirements). Also, if a packet arrives at a DS-compliant node, and the DSCP value is not mapped to any other PHB, the packet will get mapped to the default PHB.

For more information about default PHB, refer to RFC 2474, *Definition of the Differentiated Services Field in IPv4 and IPv6 Headers*.

Class-Selector PHB

To preserve backward-compatibility with any IP precedence scheme currently in use on the network, DiffServ has defined a DSCP value in the form xxx000, where x is either 0 or 1. These DSCP values are called Class-Selector Code Points. (The DSCP value for a packet with default PHB 000000 is also called the Class-Selector Code Point.)

The PHB associated with a Class-Selector Code Point is a Class-Selector PHB. These Class-Selector PHBs retain most of the forwarding behavior as nodes that implement IP Precedence-based classification and forwarding.

For example, packets with a DSCP value of 11000 (the equivalent of the IP Precedence-based value of 110) have preferential forwarding treatment (for scheduling, queueing, and so on), as compared to packets with a DSCP value of 100000 (the equivalent of the IP Precedence-based value of 100). These Class-Selector PHBs ensure that DS-compliant nodes can coexist with IP Precedence-based nodes.

For more information about Class-Selector PHB, refer to RFC 2474, *Definition of the Differentiated Services Field in IPv4 and IPv6 Headers*.

Assured Forwarding PHB

Assured Forwarding (AF) PHB is nearly equivalent to Controlled Load Service available in the integrated services model. AF_n PHB defines a method by which BAs can be given different forwarding assurances.

For example, network traffic can be divided into the following classes:

- Gold: Traffic in this category is allocated 50 percent of the available bandwidth.
- Silver: Traffic in this category is allocated 30 percent of the available bandwidth.
- Bronze: Traffic in this category is allocated 20 percent of the available bandwidth.

Further, the AF_n PHB defines four AF classes: AF1, AF2, AF3, and AF4. Each class is assigned a specific amount of buffer space and interface bandwidth, according to the SLA with the service provider or policy map.

Within each AF class, you can specify three drop precedence (dP) values: 1, 2, and 3.

Assured Forwarding PHB can be expressed as follows:

AF_ny

In this example, *n* represents the AF class number (1, 2, or 3) and *y* represents the dP value (1, 2, or 3) within the AF_n class.

In instances of network traffic congestion, if packets in a particular AF class (for example, AF1) need to be dropped, packets in the AF1 class will be dropped according to the following guideline:

$dP(AF_ny) \geq dP(AF_nz) \geq dP(AF_nx)$

where $dP(AF_n)$ is the probability that packets of the AF_n class will be dropped. In other words, y denotes the dP within an AF_n class.

In the following example, packets in the AF_{13} class will be dropped before packets in the AF_{12} class, which in turn will be dropped before packets in the AF_{11} class:

$$dP(AF_{13}) \geq dP(AF_{12}) \geq dP(AF_{11})$$

The dP method penalizes traffic flows within a particular BA that exceed the assigned bandwidth. Packets on these offending flows could be re-marked by a policer to a higher drop precedence.

An AF_x class can be denoted by the DSCP value, $xyzab0$, where xyz can be 001, 010, 011, or 100, and ab represents the dP value.

Table 16 lists the DSCP value and corresponding dP value for each AF PHB class.

Table 16 DSCP Values and Corresponding Drop Precedence Values for Each AF PHB Class

Drop Precedence	Class 1	Class 2	Class 3	Class 4
Low drop precedence	001010	010010	011010	100010
Medium drop precedence	001100	010100	011100	100100
High drop precedence	001110	010110	011110	100110

Expedited Forwarding PHB

Resource Reservation Protocol (RSVP), a component of the integrated services model, provides a Guaranteed Bandwidth Service. Applications such as Voice over IP (VoIP), video, and online trading programs require this kind of robust service. The EF PHB, a key ingredient of DiffServ, supplies this kind of robust service by providing low loss, low latency, low jitter, and assured bandwidth service.

EF can be implemented using PQ, along with rate-limiting on the class (or BA). When implemented in a DiffServ network, EF PHB provides a virtual leased line, or premium service. For optimal efficiency, however, EF PHB should be reserved for only the most critical applications because, in instances of traffic congestion, it is not feasible to treat all or most traffic as high priority.

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

The recommended DSCP value for EF PHB is 101110.

For more information about EF PHB, refer to RFC 2598, *An Expedited Forwarding PHB*.

Benefits

Use the Implementing DiffServ for End-to-End Quality of Service feature set to implement the Differentiated Services architecture. The benefits of implementing Differentiated Services include the following:

- Reduces the burden on network devices and easily scales as the network grows
- Allows customers to keep any existing Layer 3 ToS prioritization scheme that may be in use
- Allows customers to mix DiffServ-compliant devices with any existing ToS-enabled equipment in use
- Alleviates bottlenecks through efficient management of current corporate network resources

Differentiated Services Components

The following components make up the foundation of a Cisco Differentiated Services implementation:

- **Traffic conditioning (traffic policing and traffic shaping).** Traffic conditioning is performed at the edges of a DiffServ domain. Traffic conditioners perform traffic shaping and policing functions to ensure that traffic entering the DiffServ domain conforms to the rules specified by the Traffic Conditioning Agreement (TCA), and comply with the service provisioning policy of the domain. Traffic conditioning may range from simple code point re-marking to complex policing and shaping operations.
- **Packet classification.** Packet classification uses a traffic descriptor (for example, the DSCP) to categorize a packet within a specific group in order to define that packet. After the packet has been defined (that is, classified), the packet is then accessible for QoS handling on the network.

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

- **Packet marking.** Packet marking is related to packet classification. Packet marking allows you to classify a packet based on a specific traffic descriptor (such as the DSCP value). This classification can then be used to apply user-defined differentiated services to the packet and to associate a packet with a local QoS group.

Associating a packet with a local QoS group allows users to associate a group ID with a packet. The group ID can be used to classify packets into QoS groups based on prefix, autonomous system, and community string. A user can set up to 64 DSCP values and 100 QoS group markings.

- **Congestion management.** Congestion management (or scheduling) is achieved through traffic scheduling and traffic queueing. When there is network congestion, a scheduling mechanism such as CBWFQ is used to provide guaranteed bandwidth to the different classes of traffic.
- **Congestion avoidance.**

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 WRED.

With WRED and Differentiated Services, you have the option of allowing WRED to use the DSCP value when WRED calculates the drop probability of a packet.

Feature Sets

This section lists the feature sets that correspond to the Differentiated Services components listed earlier. These feature sets provide the necessary functionality that allows you to implement Differentiated Services.

This feature set includes the following features:

- **Modular QoS CLI**—This feature provides a CLI structure that is used to configure class-based QoS features.
- **Class-Based Packet Marking**—This feature provides a user-friendly CLI for efficient packet marking by which users can differentiate packets by designating them different identifying values. For example, this feature allows you to mark packets by setting the IP Precedence bits or the IP DSCP in the ToS byte.

- **Traffic Policing**—This feature allows you to limit the input or output transmission rate of a class of traffic based on user-defined criteria. It also enables the system to mark packets by setting the IP Precedence value, the QoS group, or the DSCP value.
- **Class-Based Shaping**—This feature allows you to configure Generic Traffic Shaping (GTS) on a traffic class, specify average rate or peak rate traffic shaping, and configure CBWFQ inside GTS.
- **CBWFQ**—This feature is a scheduling mechanism used to provide a minimum bandwidth guarantee to traffic classes during times of network congestion at an interface.
- **DiffServ Compliant WRED**—This feature provides support for the DiffServ standard. It enables WRED to use either the DSCP or the IP Precedence value when calculating the drop probability for a packet. This feature should be used in conjunction with CBWFQ.
- **Enhanced `show policy-map interface` Command Enhancements for Class-Based Accounting**—The `show policy-map interface` command now displays information such as the incoming traffic rate, the dropped packet rate, the number of matched packets, and the number of matched bytes for traffic classes that are attached to the specified interface. This feature collects and displays common statistics that are used for billing and accounting purposes. For more information, see the release notes for Cisco IOS Release 12.1(5)T.
- **Multiprotocol Label Switching (MPLS) Class of Service (CoS) Enhancements**—This feature allows the service provider to set the MPLS experimental field instead of overwriting the value in the customer IP Precedence field (the first three bits of the DSCP field in the header of an IP packet). For more information about MPLS Class of Service (CoS), refer to the *Cisco IOS Switching Services Configuration Guide*.

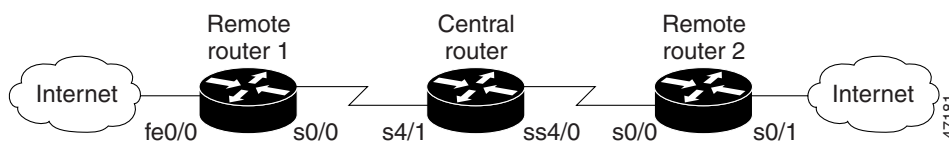
Constructing Services Using DiffServ

The following section provides a sample DiffServ implementation. It includes sample configurations and troubleshooting logs, which can be used for monitoring system performance.

Sample DiffServ Implementation

Figure 47 shows a sample DiffServ implementation with three routers: remote router 1, central router, and remote router 2.

Figure 47 Sample Network Implementing DiffServ



In this example, we want to give end-to-end QoS to several different types of traffic classes using the Cisco IOS Differentiated Services feature set.

Traffic classes along with the SLAs for each traffic class in use on the sample DiffServ implementation are described as follows:

- Voice is considered premium class. The gold class of traffic consists of TACACS sessions, along with traffic marked with DSCP values 12 and 14. The silver traffic class consists of Telnet, Simple Mail Transfer Protocol (SMTP), and FTP sessions. The bronze traffic class consists of web traffic and traffic marked with DSCP values 28 and 30. Anything else is considered as belonging to “best-effort” traffic class.
- The premium class should be forwarded with the lowest delay possible up to a maximum of 500 kbps during periods of congestion. The gold class should be treated preferentially over the silver class, which in turn should be treated preferentially over the bronze class. The gold, silver, and bronze classes should have 35 percent, 25 percent, and 15 percent, respectively, of the interface bandwidth as the minimum bandwidth guarantees. The bronze class should be shaped to 320 kbps, and the best-effort class should be policed to 56 kbps.
- To provision for the various traffic classes, the traffic needs to be classified based on DSCP values in a DiffServ domain. So that traffic can be classified based on DSCP values, the traffic should be premarked with the appropriate DSCP values at the time of entering the network.

In [Figure 47](#), the correct place to do this kind of traffic marking is in the incoming direction of Fast Ethernet interface 0/0 of remote router 1 and the incoming direction of serial interface 0/1 of remote router 2. This marking can be achieved through an input service policy.

[Table 17](#) lists the DSCP values used to mark different classes of traffic entering into the sample network.

Table 17 *DSCP Values for Traffic Classes and Traffic Types*

Traffic Class	Traffic Type	DSCP Value
Premium	Voice	46
Gold	TACACS	10
Silver	Telnet	18
	SMTP	20
	FTP	22
Bronze	HTTP	26

To achieve the marking scheme noted in [Table 17](#), use the following configuration for the policy map called SETDSCP in the input direction of Fast Ethernet interface 0/0 of remote router 1:

```
class-map match-all EF
 match access-group 101

class-map match-all AF1
 match access-group 102

class-map match-all AF21
 match access-group 108
class-map match-all AF22
 match access-group 109
class-map match-all AF23
 match access-group 110

class-map match-all AF3
 match access-group 104

policy-map SETDSCP
 class EF
  set ip dscp 46
 class AF1
  set ip dscp 10
```

```

class AF21
  set ip dscp 18
class AF22
  set ip dscp 20
class AF23
  set ip dscp 22
class AF3
  set ip dscp 26

```

Once the traffic classes are marked with the appropriate DSCP values using the SETDSCP policy map, the different behavior aggregate requirements for each of the traffic classes can be met by using the configuration for the following policy map called VOIP in the output direction:

```

class-map match-all premium
  match ip dscp 46
class-map match-all gold
  match ip dscp 10 12 14
class-map match-all silver
  match ip dscp 18 20 22
class-map match-all bronze
  match ip dscp 26 28 30
class-map best-effort
  match access-group 105

policy-map VOIP
  class premium
    priority 500
  class gold
    bandwidth percent 35
  class silver
    shape average 320000
    bandwidth percent 25
  class bronze
    bandwidth percent 15
  class best-effort
    police 56000 1750 1750 conform-action set-dscp-transmit 0

```

Sample Configurations

This section contains the configurations for each of the routers shown in [Figure 47](#).

The examples demonstrate how marking, shaping, policing, and monitoring is done through the Modular QoS CLI.

Remote Router 1 Configuration

Current configuration:

```

Remotel# show running-config

Building configuration...
!
version 12.1
no service single-slot-reload-enable
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Remotel
!
logging rate-limit console 10 except errors
no logging console

```

```
!
ip subnet-zero
!
ip dhcp smart-relay
!
ip cef
!
class-map match-all gold
  match ip dscp 10 12 14
class-map match-all EF
  match access-group 101
class-map match-all AF21
  match access-group 108
class-map match-all AF23
  match access-group 110
class-map match-all AF22
  match access-group 109
class-map match-all bronze
  match ip dscp 26 28 30
class-map match-all platinum
  match ip dscp 46
class-map match-all silver
  match ip dscp 18 20 22
class-map match-all best-effort
  match access-group 105
class-map match-all AF3
  match access-group 104
class-map match-all AF1
  match access-group 102
!
policy-map VOIP
  class platinum
    priority 500
  class gold
    bandwidth percent 50
  class bronze
    shape average 320000
    bandwidth percent 15
  class silver
    bandwidth percent 35
  class best-effort
    police 56000 1750 1750 conform-action set-dscp-transmit 0 exceed-action drop
  violate-action drop
policy-map SETDSCP
  class EF
    set ip dscp 46
  class AF1
    set ip dscp 10
  class AF3
    set ip dscp 26
  class AF21
    set ip dscp 18
  class AF22
    set ip dscp 20
  class AF23
    set ip dscp 22
!
call rsvp-sync
cns event-service server
!
interface FastEthernet0/0
  ip address 4.1.1.1 255.255.255.0
  load-interval 60
  speed auto
```

```

half-duplex
service-policy input SETDSCP
!
interface Serial0/0
bandwidth 2000
ip address 2.1.1.1 255.255.255.0
load-interval 60
service-policy output VOIP
!
interface Serial0/1
no ip address
shutdown
!
ip classless
ip route 1.1.1.0 255.255.255.0 2.1.1.2
ip route 3.1.1.0 255.255.255.0 2.1.1.2
!
access-list 101 permit udp any any range 16384 32768
access-list 102 permit tcp any any eq tacacs
access-list 104 permit tcp any any eq www
access-list 105 permit ip any any
access-list 108 permit tcp any any eq telnet
access-list 109 permit tcp any any eq smtp
access-list 110 permit tcp any any eq ftp
!
voice-port 1/0/0
!
voice-port 1/0/1
!
dial-peer cor custom
!
dial-peer voice 11 pots
destination-pattern 2220
port 1/0/0
!
dial-peer voice 1 voip
destination-pattern 1110
session target ipv4:1.1.1.2
ip precedence 5
!
line con 0
transport input none
line aux 0
line vty 0 4
login
!
no scheduler allocate
end

```

Central Router Configuration

Current configuration:

```
Central# show running-config
```

```
Building configuration...
```

Current configuration:

```

!
version 12.1
no service single-slot-reload-enable
service timestamps debug uptime
service timestamps log uptime
no service password-encryption

```



```

!
hostname Central
!
logging rate-limit console 10 except errors
no logging console

ip dhcp smart-relay
!
ip cef
!
class-map match-all gold
  match ip dscp 10 12 14
class-map match-all bronze
  match ip dscp 26 28 30
class-map match-all platinum
  match ip dscp 46
class-map match-all silver
  match ip dscp 18 20 22
class-map match-all best-effort
  match ip dscp 0
!
policy-map AVVID
  class silver
    bandwidth percent 35
    random-detect dscp-based
    random-detect dscp 18 20 40 10
    random-detect dscp 20 20 40 30
    random-detect dscp 22 2 3 3
  class gold
    bandwidth percent 50
    random-detect dscp-based
    random-detect dscp 10 20 40 10
    random-detect dscp 12 20 40 15
    random-detect dscp 14 20 40 20
  class bronze
    bandwidth percent 15
    random-detect dscp-based
    random-detect dscp 26 20 40 10
    random-detect dscp 28 20 40 20
    random-detect dscp 30 20 40 30
  class platinum
    priority 500
!
cns event-service server
!
interface Serial4/0
  bandwidth 2000
  ip address 3.1.1.1 255.255.255.0
  no ip mroute-cache
  load-interval 60
  service-policy output AVVID
!
interface Serial4/1
  ip address 2.1.1.2 255.255.255.0
  no ip mroute-cache
  service-policy output AVVID
  clockrate 2015232
!
interface Serial4/2
  no ip address
  no ip mroute-cache
  shutdown
!
interface Serial4/3

```

```

no ip address
no ip mroute-cache
shutdown
!
ip classless
ip route 0.0.0.0 0.0.0.0 10.0.153.1
ip route 1.1.1.0 255.255.255.0 3.1.1.2
ip route 4.1.1.0 255.255.255.0 2.1.1.1
ip http server
!
line con 0
  exec-timeout 0 0
  transport input none
line aux 0
line vty 0 4
line vty 5 15

end

```

Remote Router 2 Configuration

Current configuration:

```

Remote2# show running-config

Building configuration...

Current configuration:
!
version 12.1
no service single-slot-reload-enable
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Remote2
!
logging rate-limit console 10 except errors
no logging console
!
ip dhcp smart-relay
!
ip cef
!
class-map match-all gold
  match ip dscp 10 12 14
class-map match-all EF
  match access-group 101
class-map match-all AF21
  match access-group 108
class-map match-all AF23
  match access-group 110
class-map match-all AF22
  match access-group 109
class-map match-all bronze
  match ip dscp 26 28 30
class-map match-all platinum
  match ip dscp 46
class-map match-all silver
  match ip dscp 18 20 22
class-map match-all best-effort
  match access-group 105
class-map match-all AF3
  match access-group 104

```

```

class-map match-all AF1
  match access-group 102
!
!
policy-map VOIP
  class platinum
    priority 500
  class gold
    bandwidth percent 50
  class bronze
    shape average 320000
    bandwidth percent 15
  class silver
    bandwidth percent 35
  class best-effort
    police 56000 1750 1750 conform-action set-dscp-transmit 0 exceed-action drop
  violate-action drop
policy-map SETDSCP
  class EF
    set ip dscp 46
  class AF1
    set ip dscp 10
  class AF3
    set ip dscp 26
  class AF21
    set ip dscp 18
  class AF22
    set ip dscp 20
  class AF23
    set ip dscp 22
!

interface Serial0/0
  bandwidth 2000
  ip address 3.1.1.2 255.255.255.0
  load-interval 60
  service-policy output VOIP
  clockrate 2000000
!
interface Serial0/1
  ip address 1.1.1.1 255.255.255.0
  load-interval 60
  no keepalive
  service-policy input SETDSCP
  clockrate 2000000
!
ip kerberos source-interface any
ip classless
ip route 2.1.1.0 255.255.255.0 3.1.1.1
ip route 4.1.1.0 255.255.255.0 3.1.1.1
no ip http server
!
access-list 101 permit udp any any range 16384 32768
access-list 102 permit tcp any any eq tacacs
access-list 104 permit tcp any any eq www
access-list 105 permit ip any any
access-list 108 permit tcp any any eq telnet
access-list 109 permit tcp any any eq smtp
access-list 110 permit tcp any any eq ftp
!
voice-port 1/0/0
!
voice-port 1/0/1
!

```

```

dial-peer cor custom
!
dial-peer voice 1 voip
 destination-pattern 2220
 session target ipv4:2.1.1.1
 ip precedence 5
!
dial-peer voice 11 pots
 destination-pattern 1110
 port 1/0/0
!
!
line con 0
 transport input none
line aux 0
line vty 0 4
 login
!
no scheduler allocate
end

```

Troubleshooting Logs

This section contains sample troubleshooting logs for remote router 1 and the central router. These logs can be used for monitoring and maintaining the DiffServ implementation.

Remote Router 1

```
Remotel1# show policy-map SETDSCP
```

```

Policy Map SETDSCP
 Class EF
   set ip dscp 46
 Class AF1
   set ip dscp 10
 Class AF3
   set ip dscp 26
 Class AF21
   set ip dscp 18
 Class AF22
   set ip dscp 20
 Class AF23
   set ip dscp 22

```

```
Remotel1# show policy-map VOIP
```

```

Policy Map VOIP
 Class platinum
   Weighted Fair Queueing
     Strict Priority
     Bandwidth 500 (kbps) Burst 12500 (Bytes)
 Class gold
   Weighted Fair Queueing
     Bandwidth 50 (%) Max Threshold 64 (packets)
 Class bronze
   Traffic Shaping
     Average Rate Traffic Shaping
     CIR 320000 (bps) Max. Buffers Limit 1000 (Packets)
   Weighted Fair Queueing
     Bandwidth 15 (%) Max Threshold 64 (packets)
 Class silver

```

```
Weighted Fair Queueing
  Bandwidth 35 (%) Max Threshold 64 (packets)
Class best-effort
  police 56000 1750 1750 conform-action set-dscp-transmit 0 exceed-action drop
violate-action drop
```

```
Remotel# show policy-map interface f0/0
```

```
FastEthernet0/0
```

```
Service-policy input: SETDSCP (1611)
```

```
Class-map: EF (match-all) (1612/3)
  2154221 packets, 176646532 bytes
  1 minute offered rate 642000 bps, drop rate 0 bps
Match: access-group 101 (1614)
QoS Set
  ip dscp 46
  Packets marked 2154256
```

```
Class-map: AF1 (match-all) (1616/12)
  46351 packets, 69711904 bytes
  1 minute offered rate 254000 bps, drop rate 0 bps
Match: access-group 102 (1618)
QoS Set
  ip dscp 10
  Packets marked 46352
```

```
Class-map: AF3 (match-all) (1620/11)
  81757 packets, 122962528 bytes
  1 minute offered rate 483000 bps, drop rate 0 bps
Match: access-group 104 (1622)
QoS Set
  ip dscp 26
  Packets marked 81951
```

```
Class-map: AF21 (match-all) (1624/4)
  84585 packets, 127215840 bytes
  1 minute offered rate 484000 bps, drop rate 0 bps
Match: access-group 108 (1626)
QoS Set
  ip dscp 18
  Packets marked 84780
```

```
Class-map: AF22 (match-all) (1628/6)
  75440 packets, 113461760 bytes
  1 minute offered rate 423000 bps, drop rate 0 bps
Match: access-group 109 (1630)
QoS Set
  ip dscp 20
  Packets marked 75612
```

```
Class-map: AF23 (match-all) (1632/5)
  66212 packets, 99582848 bytes
  1 minute offered rate 362000 bps, drop rate 0 bps
Match: access-group 110 (1634)
QoS Set
  ip dscp 22
  Packets marked 66428
```

```
Class-map: class-default (match-any) (1636/0)
  2555349 packets, 778812687 bytes
  1 minute offered rate 2896000 bps, drop rate 0 bps
Match: any (1638)
```

```
2555358 packets, 778810855 bytes
1 minute rate 2896000 bps
```

```
Remotel1# show policy-map interface s0/0
```

```
Serial0/0
```

```
Service-policy output: VOIP (1558)
```

```
Class-map: platinum (match-all) (1559/8)
2988402 packets, 215165016 bytes
1 minute offered rate 564000 bps, drop rate 0 bps
Match: ip dscp 46 (1561)
Weighted Fair Queueing
Strict Priority
Output Queue: Conversation 264
Bandwidth 500 (kbps)
(pkts matched/bytes matched) 2988422/215166384
(total drops/bytes drops) 330478/23794416
```

```
Class-map: gold (match-all) (1563/2)
64300 packets, 96064200 bytes
1 minute offered rate 252000 bps, drop rate 0 bps
Match: ip dscp 10 12 14 (1565)
Weighted Fair Queueing
Output Queue: Conversation 265
Bandwidth 50 (%) Max Threshold 64 (packets)
(pkts matched/bytes matched) 64300/96064200
(depth/total drops/no-buffer drops) 0/0/0
```

```
Class-map: bronze (match-all) (1567/7)
115945 packets, 173221830 bytes
1 minute offered rate 479000 bps, drop rate 56000 bps
Match: ip dscp 26 28 30 (1569)
Traffic Shaping
  Target   Byte   Sustain   Excess   Interval   Increment   Adapt
  Rate    Limit  bits/int  bits/int  (ms)       (bytes)     Active
  320000  2000   8000     8000     25         1000        -

  Queue    Packets  Bytes    Packets  Bytes
  Depth
  64       80006   119528964 72784   108739296 yes
Weighted Fair Queueing
Output Queue: Conversation 266
Bandwidth 15 (%) Max Threshold 64 (packets)
(pkts matched/bytes matched) 80006/119528964
(depth/total drops/no-buffer drops) 0/12749/0
```

```
Class-map: silver (match-all) (1572/9)
315979 packets, 472072626 bytes
1 minute offered rate 1258000 bps, drop rate 646000 bps
Match: ip dscp 18 20 22 (1574)
Weighted Fair Queueing
Output Queue: Conversation 267
Bandwidth 35 (%) Max Threshold 64 (packets)
(pkts matched/bytes matched) 316253/472481982
(depth/total drops/no-buffer drops) 0/158914/0
```

```
Class-map: best-effort (match-all) (1576/10)
3548921 packets, 1051813080 bytes
1 minute offered rate 2801000 bps, drop rate 0 bps
Match: access-group 105 (1578)
police:
56000 bps, 1750 limit, 1750 extended limit
```

```

conformed 0 packets, 0 bytes; action: set-dscp-transmit 0
exceeded 0 packets, 0 bytes; action: drop
violated 0 packets, 0 bytes; action: drop

Class-map: class-default (match-any) (1580/0)
 3549281 packets, 1051837716 bytes
 1 minute offered rate 2801000 bps, drop rate 0 bps
Match: any (1582)
 3549281 packets, 1051837644 bytes
 1 minute rate 2801000 bps

Remotel# show queue serial 0/0

Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 631823
Queueing strategy: weighted fair
Output queue: 101/1000/64/593935 (size/max total/threshold/drops)
  Conversations 4/7/256 (active/max active/max total)
  Reserved Conversations 3/3 (allocated/max allocated)
  Available Bandwidth 1000 kilobits/sec

(depth/weight/total drops/no-buffer drops/interleaves) 5/0/346494/0/0
Conversation 264, linktype: ip, length: 72
source: 0.0.0.0, destination: 1.1.1.2, id: 0x0000, ttl: 59,
TOS: 184 prot: 17, source port 0, destination port 16384

(depth/weight/total drops/no-buffer drops/interleaves) 63/45/166791/0/0
Conversation 267, linktype: ip, length: 1494
source: 0.0.0.0, destination: 1.1.1.2, id: 0x0000, ttl: 59,
TOS: 72 prot: 6, source port 0, destination port 23

(depth/weight/total drops/no-buffer drops/interleaves) 35/104/13461/0/0
Conversation 266, linktype: ip, length: 1494
source: 0.0.0.0, destination: 1.1.1.2, id: 0x0000, ttl: 59,
TOS: 104 prot: 6, source port 0, destination port 80

(depth/weight/total drops/no-buffer drops/interleaves) 1/32384/67216/0/0
Conversation 89, linktype: ip, length: 1482
source: 0.0.0.0, destination: 1.1.1.2, id: 0x0000, ttl: 59,
TOS: 0 prot: 17, source port 0, destination port 67

Remotel# show interface serial 0/0

Serial0/0 is up, line protocol is up

Hardware is PowerQUICC Serial
Internet address is 2.1.1.1/24
MTU 1500 bytes, BW 2000 Kbit, DLY 20000 usec,
  reliability 255/255, txload 207/255, rxload 1/255
Encapsulation HDLC, loopback not set
Keepalive set (10 sec)
Last input 00:00:03, output 00:00:00, output hang never
Last clearing of "show interface" counters 00:50:30
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 595699
Queueing strategy: weighted fair
Output queue: 114/1000/64/560199 (size/max total/threshold/drops)
  Conversations 4/7/256 (active/max active/max total)
  Reserved Conversations 3/3 (allocated/max allocated)
  Available Bandwidth 1000 kilobits/sec
  1 minute input rate 0 bits/sec, 0 packets/sec

```

```

1 minute output rate 1624000 bits/sec, 962 packets/sec
 354 packets input, 22827 bytes, 0 no buffer
 Received 354 broadcasts, 0 runts, 0 giants, 0 throttles
 0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
2918044 packets output, 616834104 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

```

Central Router

Central# **show policy-map interface serial 4/0**

Serial4/0

Service-policy output: AVVID (2022)

```

Class-map: silver (match-all) (2023/2)
 251162 packets, 375236028 bytes
 1 minute offered rate 612000 bps, drop rate 0 bps
 Match: ip dscp 18 20 22 (2025)
 Weighted Fair Queueing
  Output Queue: Conversation 265
   Bandwidth 25 (%)
   (pkts matched/bytes matched) 3/4482
   (depth/total drops/no-buffer drops) 0/0/0
 mean queue depth: 0

```

Dscp (Prec)	Random drop pkts/bytes	Tail drop pkts/bytes	Minimum threshold	Maximum threshold	Mark probability
0(0)	0/0	0/0	20	40	1/10
1	0/0	0/0	22	40	1/10
2	0/0	0/0	24	40	1/10
3	0/0	0/0	26	40	1/10
4	0/0	0/0	28	40	1/10

(...up to DSCP 63.....)

61	0/0	0/0	30	40	1/10
62	0/0	0/0	32	40	1/10
63	0/0	0/0	34	40	1/10
rsvp	0/0	0/0	36	40	1/10

```

Class-map: gold (match-all) (2027/3)
 102479 packets, 153103626 bytes
 1 minute offered rate 250000 bps, drop rate 0 bps
 Match: ip dscp 10 12 14 (2029)
 Weighted Fair Queueing
  Output Queue: Conversation 266
   Bandwidth 35 (%)
   (pkts matched/bytes matched) 0/0
   (depth/total drops/no-buffer drops) 0/0/0
 mean queue depth: 0

```

Dscp (Prec)	Random drop pkts/bytes	Tail drop pkts/bytes	Minimum threshold	Maximum threshold	Mark probability
0(0)	0/0	0/0	20	40	1/10
1	0/0	0/0	22	40	1/10
2	0/0	0/0	24	40	1/10
3	0/0	0/0	26	40	1/10

...up to DSCP 63.....)


```

61          0/0          0/0          30          40          1/10
62          0/0          0/0          32          40          1/10
63          0/0          0/0          34          40          1/10
rsvp       0/0          0/0          36          40          1/10
    
```

```

Class-map: bronze (match-all) (2031/4)
  106605 packets, 159267870 bytes
  1 minute offered rate 262000 bps, drop rate 0 bps
  Match: ip dscp 26 28 30 (2033)
  Weighted Fair Queueing
  Output Queue: Conversation 267
  Bandwidth 15 (%)
  (pkts matched/bytes matched) 0/0
  (depth/total drops/no-buffer drops) 0/0/0
  mean queue depth: 0
    
```

Dscp (Prec)	Random drop pkts/bytes	Tail drop pkts/bytes	Minimum threshold	Maximum threshold	Mark probability
0(0)	0/0	0/0	20	40	1/10
1	0/0	0/0	22	40	1/10
2	0/0	0/0	24	40	1/10
3	0/0	0/0	26	40	1/10
4	0/0	0/0	28	40	1/10
5	0/0	0/0	30	40	1/10
6	0/0	0/0	32	40	1/10

(...up to DSCP 63.....)

```

61          0/0          0/0          30          40          1/10
62          0/0          0/0          32          40          1/10
63          0/0          0/0          34          40          1/10
rsvp       0/0          0/0          36          40          1/10
    
```

```

Class-map: platinum (match-all) (2035/5)
  4253851 packets, 306277272 bytes
  1 minute offered rate 499000 bps, drop rate 0 bps
  Match: ip dscp 46 (2037)
  Weighted Fair Queueing
  Strict Priority
  Output Queue: Conversation 264
  Bandwidth 500 (kbps)
  (pkts matched/bytes matched) 4248148/305866656
  (total drops/bytes drops) 5/360
    
```

```

Class-map: class-default (match-any) (2039/0)
  4719109 packets, 1000522466 bytes
  1 minute offered rate 1625000 bps, drop rate 0 bps
  Match: any (2041)
  4719109 packets, 1000522466 bytes
  1 minute rate 1625000 bps
    
```

Central# **show queue serial 4/0**

```

Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 5
Queueing strategy: weighted fair
Output queue: 0/1000/64/5 (size/max total/threshold/drops)
  Conversations 0/2/256 (active/max active/max total)
  Reserved Conversations 3/3 (allocated/max allocated)
  Available Bandwidth 1000 kilobits/sec
    
```

```
Central# show queue serial 4/1

Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: weighted fair
Output queue: 0/1000/64/0 (size/max total/threshold/drops)
  Conversations 0/1/256 (active/max active/max total)
  Reserved Conversations 3/3 (allocated/max allocated)
  Available Bandwidth 1011 kilobits/sec
```

Class-Based Management

The accounting functionality of DiffServ allows you to collect and display service policy statistics on a per-class basis. The **show policy-map interface** command has been enhanced to include additional information related to traffic classes on a particular interface. The **show policy-map interface** command now displays information including the incoming traffic rate, the dropped packet rate, the number of matched packets, and the number of matched bytes, for traffic classes that are attached to the specified interface. These details can be used for billing and accounting purposes, and for managing projects, as appropriate. For more information about the **show policy-map interface** command, refer to the *Cisco IOS Quality of Service Solutions Command Reference*.

What to Do Next

To configure Differentiated Services, use the following Cisco IOS features:

- Modular QoS CLI. For complete conceptual information on the Modular QoS CLI feature, see the chapter [Configuring the Modular Quality of Service Command-Line Interface](#) of this book. For information on how to configure the feature, see the chapter [Configuring the Modular Quality of Service Command-Line Interface](#) of this book.
- Class-Based Packet Marking. For complete conceptual information on the Class-Based Packet Marking feature, see the chapter [Classification Overview](#) of this book.
- Traffic Policing. For complete conceptual information on the Traffic Policing feature, see the chapter [Policing and Shaping Overview](#) of this book. For information on how to configure the feature, see the chapter of this book.
- Class-Based Shaping. For complete conceptual information on the Class-Based Shaping feature, see the chapter [Policing and Shaping Overview](#) of this book.
- DiffServ Compliant WRED. For complete conceptual information on the DiffServ Compliant Weighted Random Early Detection feature, see the chapter [Congestion Avoidance Overview](#) of this book. For information on how to configure the feature, see the chapter of this book.
- MPLS CoS Enhancements. For more information about MPLS Class of Service (CoS), refer to the *Cisco IOS Switching Services Configuration Guide*.



QoS: Classification, Policing, and Marking on LAC

The QoS: Classification, Policing, and Marking on 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.

Feature History for QoS: Classification, Policing, and Marking on LAC

Release	Modification
12.3(8)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- [Prerequisites for QoS: Classification, Policing, and Marking on LAC, page 894](#)
- [Restrictions for QoS: Classification, Policing, and Marking on LAC, page 894](#)
- [Information About QoS: Classification, Policing, and Marking on LAC, page 894](#)
- [How to Configure QoS: Classification, Policing, and Marking on LAC, page 896](#)
- [Configuration Examples for QoS: Classification, Policing, and Marking on LAC, page 896](#)
- [Additional References, page 902](#)
- [Command Reference, page 904](#)
- [Glossary, page 905](#)

Prerequisites for QoS: Classification, Policing, and Marking on LAC

Configure the Routers

You must configure the Client router, the LAC, and the LNS before applying the QoS policy map as described in the [“Configuration Examples for QoS: Classification, Policing, and Marking on LAC”](#) section on page 896.

Verify the State of the Subscriber Service Switch (SSS) Sessions

You must use the **show sss session** command to verify that the user sessions are enabled on the LAC.

Configure the Interface

You must configure the virtual-template interface before applying the policy map to the session.

Restrictions for QoS: Classification, Policing, and Marking on LAC

The following restrictions apply to the QoS: Classification, Policing, and Marking on LAC feature:

- Service-policy on Point-to-Point Protocol over X.25 (PPPoX) interfaces is not supported.
- Class-based queueing and class-based shaping are not supported.
- Layer 2 marking is not supported.
- The QoS Management Information Base (MIB) is not supported.
- The **clear counters** command does not clear the counters of the QoS policy map.
- Multi-hop virtual private dial-up networks (VPDNs) are not supported.

Information About QoS: Classification, Policing, and Marking on LAC

To use the QoS: Classification, Policing, and Marking on LAC feature, you should understand the following concepts:

- [Benefits](#), page 895
- [QoS Policy Map](#), page 895
- [Upstream Traffic](#), page 895
- [Downstream Traffic](#), page 895
- [SSS Session](#), page 895

Benefits

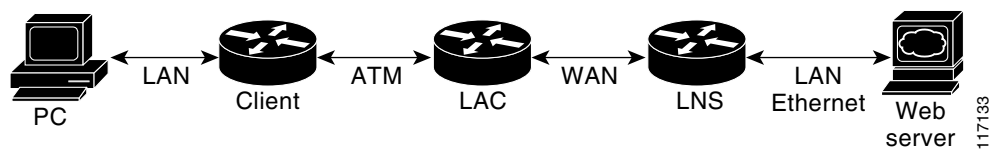
- Provides policing and marking on a per-session basis for the traffic forwarded into L2TP tunnels to the appropriate L2TP Network Server (LNS) and for traffic coming from an L2TP tunnel toward a customer edge router.
- Helps recognize the IP ToS value in the Point-to-Point Protocol over Ethernet (PPPoE) encapsulated traffic in order to classify and police the traffic according to the DSCP value.

QoS Policy Map

QoS policing and marking can be achieved by attaching a QoS policy map to the user interface on the LAC in the input and output directions. By using tunnels, input and output service policies can be attached to interfaces. Policy maps get enforced as the packet enters or leaves the tunnel.

Figure 48 shows the deployment of QoS on PPPoE sessions originating at the client and terminating at the LNS.

Figure 48 Sample Topology for QoS on PPOE Sessions



Note

The LAC is a Cisco Series 7200 router.

Upstream Traffic

Upstream traffic corresponds to packets traversing from the tunnel source to the tunnel destination; in this case, the traffic moves from the LAC to the LNS. The input QoS policy map acts on the upstream traffic before the packet gets encapsulated with the tunnel header.

Downstream Traffic

Downstream traffic corresponds to packets traversing from the tunnel destination to tunnel source; in this case, the traffic going from the LNS to the LAC. The output QoS policy map acts on the downstream traffic after the tunnel encapsulation is removed from the packet header.

SSS Session

The SSS session provides you with the infrastructure to apply QoS features on a per-session basis. The SSS session is preconfigured on the virtual template and you can use this template to provide QoS classification, policing, and marking.

You can verify the statistics of the upstream and downstream traffic from a QoS policy map in an SSS session by using the **show policy-map session** command.

How to Configure QoS: Classification, Policing, and Marking on LAC

This section contains the following task:

- [Verifying a QoS Policy Map, page 896](#) (optional)

Verifying a QoS Policy Map

To enable a service provider to verify the statistics of the upstream and downstream traffic from a QoS policy map in an SSS session, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show policy-map session [uid *uid-number*] [input | output [class *class-name*]]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	show policy-map session [uid <i>uid-number</i>] [input output [class <i>class-name</i>]] Example: Router# show policy-map session uid 401 output	Displays the information about the session identified by the unique ID.

Configuration Examples for QoS: Classification, Policing, and Marking on LAC

This section contains the following configuration examples:

- [Configuring the Routers: Example, page 897](#)
- [Verifying the SSS Session: Example, page 899](#)
- [Applying the QoS Policy Map: Example, page 900](#)
- [Configuring the LAC: Example, page 900](#)
- [Verifying the QoS Policy Map for Downstream Traffic: Example, page 900](#)
- [Applying the QoS Policy Map to the Session: Example, page 901](#)

- [Verifying the QoS Policy Map for Upstream Traffic: Example, page 902](#)

**Note**

The following examples show you how to apply QoS policy maps to upstream and downstream user session traffic to achieve the required Service Level Agreements (SLAs) provided by the service provider.

Configuring the Routers: Example

The following example shows the configuration of the routers before verifying the QoS policy map.

Client Configuration

When you log in to the PC, a PPPoE session is established at the client facing the LAC. This PPPoE session is forwarded through the L2TP tunnel from the LAC to the LNS at which point the PPPoE session terminates.

To apply QoS sessions to the user traffic originating from the PC to the web server and to the traffic originating from the web server to the PC, you should apply a QoS policy map to the user session on the LAC in the input and output directions. The classification will be based on the user traffic originating at the PC and the web traffic originating at the web server.

This topology supports bidirectional traffic, meaning that traffic can flow from the PC to the web server, and from the web server to the PC.

```
username xyz@cisco.com password 0 cisco
username qos4-72a password 0 cisco
username qos4-72b password 0 cisco

aaa authentication ppp default local
aaa session-id common

ip cef
vpdn enable
!
vpdn-group 1
 request-dialin
  protocol pppoe
!
pppoe-forwarding

interface ATM5/0
 no ip address
 no ip redirects
 no ip proxy-arp
 no ip mroute-cache
 load-interval 30
 no atm ilmi-keepalive
!
interface ATM5/0.1 point-to-point
 pvc 0/100
  encapsulation aal5snap
  pppoe max-sessions 100
  pppoe-client dial-pool-number 1
!
!interface Dialer1
 mtu 1492
 ip address negotiated
 encapsulation ppp
 dialer pool 1
```

```

no peer default ip address
no cdp enable
ppp authentication chap callin
ppp chap hostname xyz@cisco.com
ppp chap password 0 cisco
ppp ipcp dns request
!
```

LAC Configuration

The following example shows that the interfaces between the client and the LAC are ATM5/0 interfaces.

```

username xyz@cisco.com password 0 cisco
username qos4-72a password 0 cisco
username qos4-72b password 0 cisco

aaa new-model
!
!
aaa authentication ppp default local
aaa session-id common

ip cef
vpdn enable
!
vpdn-group 1
 accept-dialin
  protocol pppoe
  virtual-template 1
!
vpdn-group 2
 request-dialin
  protocol l2tp
  domain cisco.com
  initiate-to ip 156.1.1.3
  local name lac
  no l2tp tunnel authentication
  ip tos reflect
!
pppoe-forwarding

interface Serial3/6
 bandwidth 2015
 ip address 10.10.100.1 255.255.255.0
 no ip redirects
 no ip proxy-arp
 load-interval 30
 no keepalive
 no cdp enable
!

interface ATM5/0
 no ip address
 no ip redirects
 no ip proxy-arp
 load-interval 30
 no atm ilmi-keepalive
!
interface ATM5/0.1 point-to-point
 pvc 0/100
  encapsulation aal5snap
  pppoe max-sessions 100
  protocol ppp Virtual-Template1
  protocol pppoe
```



```

!
!
interface Virtual-Template1
  mtu 1492
  no ip address
  no peer default ip address
  ppp authentication chap
!

```

LNS Configuration

The following example shows that the interfaces between the LAC and the LNS are Serial3/6 interfaces.

```

username xyz@cisco.com password 0 cisco
username qos4-72b password 0 cisco
username qos4-72a password 0 cisco
aaa new-model
!
!
aaa authentication ppp default local
aaa session-id common

ip cef
vpdn enable
!
vpdn-group 1
  accept-dialin
  protocol any
  virtual-template 1
  terminate-from hostname lac
  local name lns
  lcp renegotiation always
  no l2tp tunnel authentication
  ip tos reflect
!

interface Serial3/6
  bandwidth 2015
  ip address 10.10.100.1 255.255.255.0
  no ip redirects
  no ip proxy-arp
  no ip mroute-cache
  load-interval 30
  no keepalive
  no cdp enable
!

```

Verifying the SSS Session: Example

The following example from the **show sss session** command shows that a user session is enabled on the LAC:

```
Router# show sss session
```

```

Current SSS Information: Total sessions 1
Uniq ID Type      State      Service   Identifier      Last Chg
401    PPPoE/PPP  connected Forwarded  xyz@cisco.com  00:02:06

```

Applying the QoS Policy Map: Example

The following output shows a QoS policy map to be applied to the user session in the output direction, which is the downstream traffic coming into the PC from the web server. The first subclass of traffic within the session is marked with dscp af11, the second subclass is policed, and the third subclass is dropped.

```
class-map match-any customer1234
  match ip dscp cs1 cs2 cs3 cs4
class-map match-any customer56
  match ip dscp cs5 cs6
class-map match-any customer7
  match ip dscp cs7

policy-map downstream-policy
  class customer1234
    set ip dscp af11
  class customer56
    police cir 20000 bc 10000 pir 40000 be 10000
      conform-action set-dscp-transmit af21
      exceed-action set-dscp-transmit af22
      violate-action set-dscp-transmit af23
  class customer7
    drop
```

Configuring the LAC: Example

The following example from the **interface virtual-template** command shows a QoS policy map being applied to the user session on the LAC:

```
Router# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface virtual-template1

Router(config-if)# service-policy output downstream-policy

Router(config-if)# end
```

Verifying the QoS Policy Map for Downstream Traffic: Example

In the following example from the **show policy-map session** command, the QoS policy map is applied for traffic in the downstream direction.



Note

The session ID, 401, is obtained from the output of the **show sss session** command in the [“Verifying the SSS Session: Example”](#) section on page 899.

```
Router# show policy-map session uid 401 output

SSS session identifier 401 -

  Service-policy output: downstream-policy

    Class-map: customer1234 (match-any)
      4464 packets, 249984 bytes
```

```

5 minute offered rate 17000 bps, drop rate 0 bps
Match: ip dscp cs1 cs2 cs3 cs4
    4464 packets, 249984 bytes
    5 minute rate 17000 bps
QoS Set
    dscp af11
    Packets marked 4464

Class-map: customer56 (match-any)
2232 packets, 124992 bytes
5 minute offered rate 8000 bps, drop rate 0 bps
Match: ip dscp cs5 cs6
    2232 packets, 124992 bytes
    5 minute rate 8000 bps
police:
    cir 20000 bps, bc 10000 bytes
    pir 40000 bps, be 10000 bytes
conformed 2232 packets, 124992 bytes; actions:
    set-dscp-transmit af21
exceeded 0 packets, 0 bytes; actions:
    set-dscp-transmit af22
violated 0 packets, 0 bytes; actions:
    set-dscp-transmit af23
conformed 8000 bps, exceed 0 bps, violate 0 bps

Class-map: customer7 (match-any)
1116 packets, 62496 bytes
5 minute offered rate 4000 bps, drop rate 4000 bps
Match: ip dscp cs7
    1116 packets, 62496 bytes
    5 minute rate 4000 bps
drop

Class-map: class-default (match-any)
1236 packets, 68272 bytes
5 minute offered rate 4000 bps, drop rate 0 bps
Match: any

```

Applying the QoS Policy Map to the Session: Example

In the following example, the service provider applies a QoS policy map to the user session in order to limit the amount of bandwidth that the user session is permitted to consume in the upstream direction from the PC to the web server.

```

policy-map upstream-policy
class class-default
    police cir 8000 bc 1500 be 1500
    conform-action transmit
    exceed-action drop

```

This QoS policy map is then applied to the user session as follows:

```
Router# configure terminal
```

Enter configuration commands, one per line. End with CNTL/Z.

```
Router(config)# interface virtual-templated
```

```
Router(config-if)# service-policy input upstream-policy
```

```
Router(config-if)# end
```

Verifying the QoS Policy Map for Upstream Traffic: Example

In the following example from the **show policy-map session** command, the QoS policy map is applied for traffic in the upstream direction.



Note

The session ID, 401, is obtained from the output of the **show sss session** command in the [“Verifying the SSS Session: Example”](#) section on page 899.

```
Router# show policy-map session uid 401 input
SSS session identifier 401 -

Service-policy input: upstream-policy

Class-map: class-default (match-any)
 1920 packets, 111264 bytes
 5 minute offered rate 7000 bps, drop rate 5000 bps
Match: any
police:
  cir 8000 bps, bc 1500 bytes
  conformed 488 packets, 29452 bytes; actions:
    transmit
  exceeded 1432 packets, 81812 bytes; actions:
    drop
  conformed 7000 bps, exceed 5000 bps
```

Additional References

The following sections provide references related to the QoS: Classification, Policing, and Marking on LAC 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 , Release 12.3 T
Information about attaching policy maps to interfaces	Cisco IOS Quality of Service Solutions Configuration Guide , Release 12.3
WAN configuration	Cisco IOS Wide-Area Networking Configuration Guide , Release 12.3
L2TPv3	Layer 2 Tunnel Protocol Version 3 feature module
DSCP	“Implementing DiffServ for End-to-End Quality of Service Overview” chapter of the Cisco IOS Quality of Service Configuration Guide , Release 12.3

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
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
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 new command is pertinent to this feature. To see the command pages for this command and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at <http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm>.

- **show policy-map session**

Glossary

DSCP—differentiated services code point. A marker in the header of an IP packet that indicates the per-hop behavior given to the packet within the service provider network.

LAC—Layer 2 Tunneling Protocol (L2TP) access concentrator. A node that acts as one side of an L2TP tunnel endpoint and 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.

L2TP—Layer 2 Tunneling Protocol. An Internet Engineering Task Force (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 virtual private dialup network (VPDN).

LNS—L2TP Network Server. A node that acts as one side of an L2TP tunnel endpoint and 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.

PPoE—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.

QoS—quality of service. A measure of performance for a transmission system that reflects its transmission quality and service availability.

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.

ToS—type of service. An 8-bit field carried in the header of an Internet Protocol Version 4 (IPv4) header that can be used to identify packets designated to receive preferential treatment on a class of service (CoS) basis.

**Note**

Refer to [Internetworking Terms and Acronyms](#) for terms not included in this glossary.



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 quality of service (QoS) targets for networked applications.

Feature History for QoS Bandwidth Estimation

Release	Modification
12.3(14)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- [Prerequisites for QoS Bandwidth Estimation, page 907](#)
- [Restrictions for QoS Bandwidth Estimation, page 908](#)
- [Information About QoS Bandwidth Estimation, page 908](#)
- [How to Configure QoS Bandwidth Estimation, page 912](#)
- [Configuration Examples for QoS Bandwidth Estimation, page 916](#)
- [Additional References, page 918](#)
- [Command Reference, page 920](#)
- [Glossary, page 921](#)

Prerequisites for QoS Bandwidth Estimation

- Before using this feature, configure a class map and a policy map using the modular quality of service (QoS) command-line interface (CLI) (MQC), and specify the appropriate match criteria.
- This feature requires the purchase of a Cisco IOS software feature license. The right to use this feature is not included in the base Cisco IOS software license for the software image.

Restrictions for QoS Bandwidth Estimation

This feature supports policy maps that are attached to interfaces in an output direction only.

Information About QoS Bandwidth Estimation

To use the QoS Bandwidth Estimation feature, you need to understand the following concepts:

- [Feature Overview of QoS Bandwidth Estimation, page 908](#)
- [Benefits of QoS Bandwidth Estimation, page 910](#)

Feature Overview of QoS Bandwidth Estimation

Allocating adequate bandwidth is key to assuring the network performance required for applications. However, allocating too much bandwidth can be costly. The QoS Bandwidth Estimation feature in Cisco IOS software uses Corvil Bandwidth technology to allow you as a network manager to determine the bandwidth requirements to achieve user-specified quality of service (QoS) targets for networked applications.

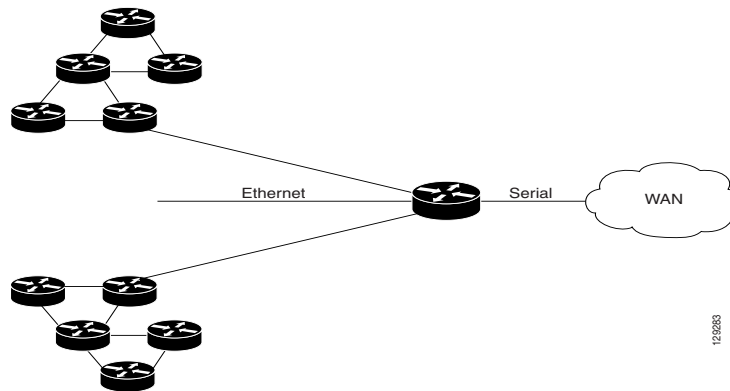
Corvil Bandwidth can determine the minimum bandwidth required to deliver traffic within customer-specified QoS targets with statistical reliability. From a network management perspective, an application's QoS requirements are characterized with respect to its sensitivity to packet loss and delay. Corvil Bandwidth provides a way to specify limits for delay and packet loss, and get a tight estimate of the minimum bandwidth essential to achieve desired application performance.

Corvil Bandwidth achieves its results by taking very short timescale (8- millisecond) snapshots of traffic and summarizing them in traffic descriptors that place very low overhead on the router because each descriptor has fewer than 300 bytes. These traffic descriptors record the exceptional events (bursts) and are input to the Corvil Bandwidth algorithm to calculate the minimum bandwidth required to deliver the user-specified QoS target for the observed traffic. (The QoS target is specified in terms of sensitivity to traffic delay and packet loss. For example, voice over IP (VoIP) traffic is very sensitive to both, whereas e-mail file transfer is sensitive to neither.)

As a result, turning on Corvil Bandwidth in the router allows you to obtain bandwidth values that can be used directly to configure the existing Cisco IOS QoS mechanisms on the router to achieve the required application performance as efficiently as possible.

For example, in [Figure 49](#), Corvil Bandwidth is enabled on the router so that the serial interface can deliver the WAN traffic within the customer-specified QoS targets with statistical reliability.

Figure 49 Sample Topology Using QoS Bandwidth Estimation



Applying Corvil Bandwidth

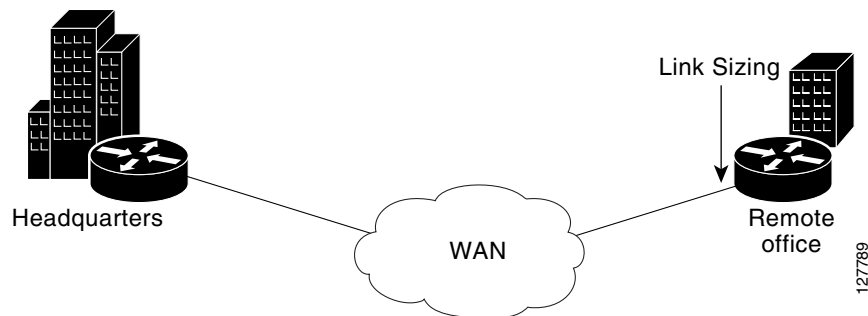
The following sections describe how Corvil Bandwidth can be implemented:

- [Link Sizing, page 909](#)
- [Bandwidth Allocations by Traffic Class, page 909](#)

Link Sizing

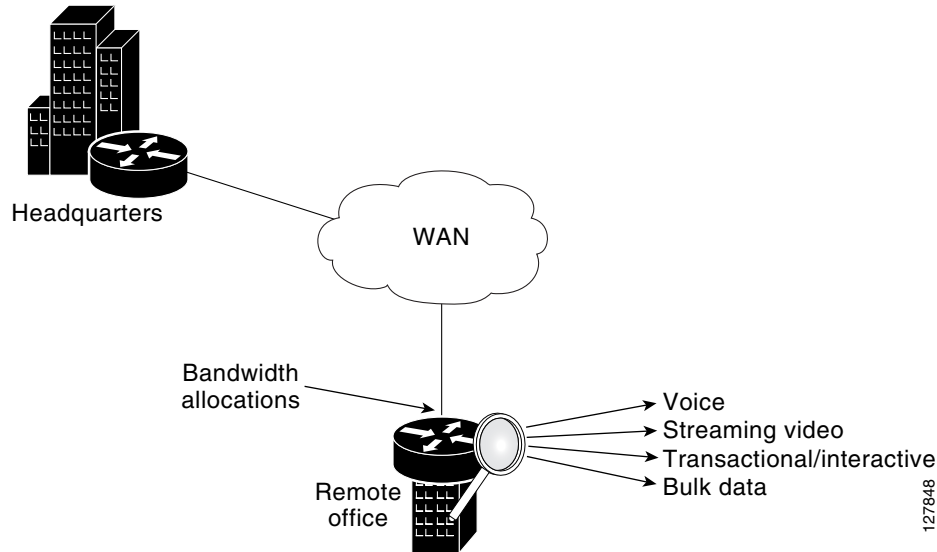
To use Corvil Bandwidth to establish the overall bandwidth requirement for a link, you start with QoS targets appropriate for the speed of the link and for the applications being carried on the link ([Figure 50](#)). The QoS targets are achieved as long as the link capacity is greater than or equal to the computed Corvil Bandwidth value.

Figure 50 Link Sizing



Bandwidth Allocations by Traffic Class

Corvil Bandwidth can be used to size bandwidth allocations for individual traffic classes defined via the MQC ([Figure 51](#)). You specify the QoS target for a traffic class, and Corvil Bandwidth reports the minimum amount of bandwidth that must be allocated to meet that target. The Corvil Bandwidth value can be used directly in the corresponding MQC policy. (The bandwidth allocation is not changed automatically.)

Figure 51 Bandwidth Allocations

Benefits of QoS Bandwidth Estimation

Table 18 shows the features and benefits of QoS Bandwidth Estimation using Corvil Bandwidth technology.

Table 18 QoS Bandwidth Estimation

Feature	Benefits
User-specified packet loss and delay targets	<ul style="list-style-type: none"> Establishment of service-level objectives for the desired performance of networked applications Elimination of operational overhead and guesswork in bandwidth provisioning and QoS configuration Potentially significant bandwidth cost savings while meeting QoS requirements Increased capability and flexibility to offer bandwidth-on-demand types of services
Frequent fine-grain traffic measurements	<ul style="list-style-type: none"> More accurate calculation of bandwidth requirements Greater ability to meet more stringent QoS targets
Support for multiple traffic classes on an interface	<ul style="list-style-type: none"> Individually specified QoS targets for each traffic class (class map) to calculate Corvil Bandwidth values
Corvil Bandwidth integrated with MQC	<ul style="list-style-type: none"> Results available by traffic class Bandwidth adjustment enabled in the corresponding MQC-based policy
Corvil Bandwidth results reported in kbps	<ul style="list-style-type: none"> Results directly applied via Cisco IOS MQC bandwidth command and to link-rate sizing

Table 18 *QoS Bandwidth Estimation (continued)*

Feature	Benefits
Corvil Bandwidth results available in class-based QoS MIB	<ul style="list-style-type: none">• Integrated with Simple Network Management Protocol (SNMP)-based performance management tools
Low resource consumption on router	<ul style="list-style-type: none">• Efficient to use, adding little additional processing or memory requirements
Available on any router interface	<ul style="list-style-type: none">• Applicable to serial, T1/E1, FastEthernet, and other interfaces as well as ATM virtual circuits (VCs), Frame Relay permanent virtual circuits (PVCs), multilink bundle interfaces, and virtual LAN (VLAN) subinterfaces

How to Configure QoS Bandwidth Estimation

This section contains the following procedures:

- [Generating a Bandwidth Estimate, page 912](#) (required)
- [Attaching the Policy Map to an Interface, page 914](#) (required)
- [Verifying the Configuration, page 915](#) (optional)

Generating a Bandwidth Estimate

Perform the following task to generate a bandwidth estimate.

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. **estimate bandwidth drop-one-in** *n* **delay-one-in** *n* **milliseconds** *n*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	policy-map <i>policy-map-name</i> Example: Router(config)# policy-map my-policy	Specifies the name of the policy map to be created. Enters policy-map configuration mode. <ul style="list-style-type: none">• Enter the policy-map name.
Step 4	class [<i>class-name</i> class-default] Example: Router(config-pmap)# class my-class	Specifies the class so that you can configure or modify its policy. Enters policy-map class configuration mode. <ul style="list-style-type: none">• Enter the class name or use the class-default keyword.

	Command or Action	Purpose
Step 5	<p>bandwidth [<i>bandwidth-kbps</i> remaining percent <i>percentage</i> percent <i>percentage</i>]</p> <p>Example: Router(config-pmap-c)# bandwidth percent 20</p>	<p>Specifies or modifies the bandwidth allocated for a class belonging to a policy map.</p> <ul style="list-style-type: none"> Enter the bandwidth to be set or modified.
Step 6	<p>estimate bandwidth [drop-one-in <i>n</i>] [delay-one-in <i>n</i> milliseconds <i>n</i>]</p> <p>Example: Router(config-pmap-c)# estimate bandwidth drop-one-in 100 delay-one-in 100 milliseconds 50</p>	<p>(Optional) Estimates the bandwidth needed per traffic class for given quality of service (QoS) targets based on traffic data.</p> <ul style="list-style-type: none"> Enter values for the packet loss target, the delay target, and the delay threshold.
Step 7	<p>end</p> <p>Example: Router(config-pmap-c)# end</p>	<p>(Optional) Exits policy-map class configuration mode.</p>

Attaching the Policy Map to an Interface

Perform the following task to attach the policy map to an interface.

Restrictions

This feature supports policy maps attached to an interface in the output direction only.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
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. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> [<i>name-tag</i>] Example: Router(config)# interface f0/1	Configures the interface type specified and enters interface configuration mode. <ul style="list-style-type: none">Enter interface type.
Step 4	service-policy { input output } <i>policy-map-name</i> Example: Router(config-if)# service-policy output my-policy	Specifies the name of the policy map to be attached to the input direction of the interface. Note You can configure policy maps on ingress or egress routers and attach them 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 to your network configuration. For this feature, only the output direction is supported. <ul style="list-style-type: none">Enter the output keyword followed by the policy map name.
Step 5	end Example: Router(config-if)# end	(Optional) Exits interface configuration mode.

Verifying the Configuration

Perform the following task to verify that bandwidth estimates have been generated.

SUMMARY STEPS

1. **enable**
2. **show policy-map interface** *interface-name* [**vc** [*vpi/* *vci*]][**dlci** *dlci*] [**input** | **output**]
3. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	show policy-map interface <i>interface-name</i> [vc [<i>vpi/</i> <i>vci</i>]][dlci <i>dlci</i>] [input output] Example: Router# show policy-map interface f0/1	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. <ul style="list-style-type: none"> • Enter the interface name.
Step 3	end Example: Router# end	(Optional) Exits privileged EXEC mode.

Configuration Examples for QoS Bandwidth Estimation

This section contains the following configuration examples:

- [Generating Bandwidth Estimates for QoS Targets: Example, page 916](#)
- [Attaching the Policy Map to an Interface: Example, page 916](#)
- [Verifying the Configuration: Example, page 916](#)

Generating Bandwidth Estimates for QoS Targets: Example

In the following example, a policy map and a traffic class are configured. Then bandwidth estimates for QoS targets including packet loss rate, delay time and probability, and timeframe in milliseconds are configured.

```
Router# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Router(config)# policy-map my-policy
Router(config-pmap)# class my-class
Router(config-pmap-c)# bandwidth percent 20
Router(config-pmap-c)# estimate bandwidth drop-one-in 100 delay-one-in 100 milliseconds 50
Router(config-pmap-c)# end
```

Attaching the Policy Map to an Interface: Example

The following example shows the policy map named my-policy being attached to the FastEthernet 0/1 interface in the output direction:

```
Router# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Router(config)# interface f0/1
Router(config-if)# service-policy output my-policy
Router(config-if)# end
```

Verifying the Configuration: Example

The following example from the **show policy-map interface** command verifies that the policy map named my-policy is attached to the FastEthernet 0/1 interface in the output direction and bandwidth estimates have been created:

```
Router# show policy-map interface f0/1
FastEthernet0/1

Service-policy output: my-policy

Class-map: icmp (match-all)
  199 packets, 22686 bytes
  30 second offered rate 0 bps, drop rate 0 bps
Match: access-group 101
Bandwidth Estimation:
  Quality-of-Service targets:
    drop no more than one packet in 1000 (Packet loss < 0.10%)
```

```
    delay no more than one packet in 100 by 40 (or more) milliseconds
      (Confidence: 99.0000%)
    Corvil Bandwidth: 1 kbits/sec

Class-map: class-default (match-any)
  112 packets, 14227 bytes
  30 second offered rate 0 bps, drop rate 0 bps
  Match: any
  Bandwidth Estimation:
    Quality-of-Service targets:
      <none specified, falling back to drop no more than one packet in 500
    Corvil Bandwidth: 1 kbits/sec
```

Additional References

The following sections provide references related to the QoS Bandwidth Estimation feature.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	<i>Cisco IOS Quality of Service Solutions Command Reference</i> , Release 12.3 T
MQC	<i>Cisco IOS Quality of Service Solutions Configuration Guide</i> , Release 12.3

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
<ul style="list-style-type: none"> 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
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 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

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at <http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm>.

New Commands

- `estimate bandwidth`

Modified Commands

- `show policy-map interface`

Glossary

Corvil Bandwidth—The optimum bandwidth that delivers predictability in QoS targets while maximizing the efficiency of the network.

CTD—Corvil traffic descriptor. A compact encoding of the distribution of bit and packet rates in a traffic aggregate over any given time window. CTDs summarize observed traffic and are input for the Corvil algorithm that calculates the minimum bandwidth required to deliver the user-specified QoS target for the observed traffic.

delay—The time taken from point-to-point in a network. Delay can be measured in either one-way or round-trip delay. See also latency

latency—The delay on a router between the time a device receives a packet and the time that packet is forwarded out the destination port.

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.

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.

traffic class—Three elements used to classify traffic. They include: 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 the **match** commands.

**Note**

Refer to [Internetworking Terms and Acronyms](#) for terms not included in this glossary.



Part 8: Modular Quality of Service Command-Line Interface





Modular Quality of Service Command-Line Interface Overview

This chapter provides a high-level overview of the Modular Quality of Service (QoS) Command-Line Interface (CLI), a feature that allows users to specify a traffic class independently of QoS policies.

For information on how to configure the Modular QoS CLI, see the chapter [Configuring the Modular Quality of Service Command-Line Interface](#) in this book.

About the Modular QoS CLI

The Modular QoS CLI is a CLI structure that allows users to create traffic polices and attach these polices to interfaces. A traffic policy contains a traffic class and one or more QoS features. A traffic class is used to classify traffic, while the QoS features in the traffic policy determine how to treat the classified traffic.

Modular QoS CLI configuration includes contains the following three steps, which are detailed more thoroughly in the [Configuring the Modular Quality of Service Command-Line Interface](#) of this book:

-
- Step 1** Define a traffic class with the **class-map** command.
 - Step 2** Create a traffic policy by associating the traffic class with one or more QoS features (using the **policy-map** command).
 - Step 3** Attach the traffic policy to the interface with the **service-policy** command.
-

The **class-map** command is used to define a traffic class. The purpose of a traffic class is to classify traffic.

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. 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. The default traffic class is detailed more thoroughly in the [Configuring the Modular Quality of Service Command-Line Interface](#) chapter of this book.

The instruction on how to evaluate these **match** commands needs to be specified if more than one match criterion exists in the traffic class. The evaluation instruction is specified with one of the following two options: **class-map match-any** or **class-map match-all**. If **match-any** is specified as the evaluation instruction, the traffic being evaluated by the traffic class must match one of the specified criteria. If **match-all** is specified as the evaluation instruction, the traffic being evaluated by the traffic class must match all of the specified criteria. The functionality of these options is detailed more thoroughly in the [Configuring the Modular Quality of Service Command-Line Interface](#) chapter of this book.

The **policy-map** command is used to create a traffic policy. The purpose of a traffic policy is to configure the QoS features that should be associated with the traffic that has been classified in a user-specified traffic class or classes. A traffic policy contains three elements: a name, a traffic class (specified with the **class-map** command), and the QoS policies (which are detailed in the [Configuring the Modular Quality of Service Command-Line Interface](#) chapter of this book). The name of a traffic policy is specified in the **policy-map** CLI (for example, issuing the **policy-map class1** command would create a traffic policy named class1). The traffic class that is used to classify traffic to the specified traffic policy is defined in policy map configuration mode, which is the automatic mode after naming the traffic policy. After choosing the traffic class that is used to classify traffic to the traffic policy, the user can enter the QoS features to apply to the classified traffic. This is done in policy-map class configuration mode. The QoS feature options are detailed more thoroughly in the [Configuring the Modular Quality of Service Command-Line Interface](#) chapter of this book.

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

Similarly, the Modular QoS CLI 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. An example of a nested traffic class configuration using both **match-all** and **match-any** is provided in the [Configuring the Modular Quality of Service Command-Line Interface](#) chapter of this book.

**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.

The **service-policy** command is used to attach the traffic policy, as specified with the **policy-map** command, to an interface. Because the elements of the traffic policy can be applied to packets entering and leaving the interface, users are required to specify whether the traffic policy characteristics should be applied to incoming or outgoing packets. For instance, the **service-policy output class1** command would attach all the characteristics of the traffic policy named class1 to the specified interface. All packets leaving the specified interface are evaluated according to the criteria specified in the traffic policy named class1. For information on using the **service-policy** command, see the [Configuring the Modular Quality of Service Command-Line Interface](#) chapter of this book.

Supported MIB

The Class-Based Quality of Service Management Information Base (Class-Based QoS MIB) provides read access to QoS configurations. This MIB also provides QoS statistics information based on the Modular QoS CLI, including information regarding class map and policy map parameters.

This Class-Based QoS MIB is actually two MIBs: CISCO-CLASS-BASED-QOS-MIB and CISCO-CLASS-BASED-QOS-CAPABILITY-MIB.

Use the Cisco Network Management Toolkit for MIBs tool on Cisco.com to locate MIBs.



Configuring the Modular Quality of Service Command-Line Interface

This section describes the tasks for configuring QoS functionality with the Modular Quality of Service (QoS) Command-Line Interface (CLI).

For complete conceptual information, see the chapter [Modular Quality of Service Command-Line Interface Overview](#) in this book.

For a complete description of the QoS commands in this chapter, refer to 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.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the “[Finding Additional Feature Support Information](#)” section on page lxix in the [Using Cisco IOS Software for Release 12.4](#) chapter in this book.

Modular QoS CLI Configuration Task List

To configure and enable class-based QoS features, perform the tasks described in the following sections. The tasks in the first three sections are required; the task in the remaining section is optional.

- [Creating a Traffic Class](#) (Required)
- [Creating a Traffic Policy](#) (Required)
- [Attaching a Traffic Policy to an Interface](#) (Required)
- [Verifying the Configuration](#) (Optional)

See the end of this chapter for the section “[Modular QoS CLI Configuration Examples](#).”

Creating a Traffic Class

The **class-map** global configuration command is used to create a traffic class. The syntax of the **class-map** command is as follows:

```
class-map [match-any | match-all] class-name  
no class-map [match-any | match-all] class-name
```

The **match all** and **match any** options need to be specified only if more than one match criterion is configured in the traffic class. The **class-map match-all** command is used when all of the match criteria in the traffic class must be met in order for a packet to match the specified traffic class. The **class-map match-any** command is used when only one of the match criterion in the traffic class must be met in order for a packet to match the specified traffic class. If neither the **match-all** nor **match-any** keyword is specified, the traffic class will behave in a manner consistent with **class-map match-all** command.

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.

For additional information on using the **match-any** and **match-all** options, see the “[Configuring the Modular Quality of Service Command-Line Interface](#)” chapter of this book.

To create a traffic class containing match criteria, use the **class-map** global configuration command to specify the traffic class name, and then use the following **match** commands in class-map configuration mode, as needed.

**Note**

This chapter lists the match commands available as of Release 12.2. For the most current information about match commands and Cisco IOS software, see the New Feature Documentation index for your particular Cisco IOS software release on Cisco.com.

Command	Purpose
Router(config)# class-map <i>class-map-name</i>	Specifies the user-defined name of the traffic class. Names can be a maximum of 40 alphanumeric characters. If match-all or match-any are not specified, traffic must match all the match criterion to be classified as part of the traffic class.
Router(config)# class-map match-all <i>class-map-name</i>	Specifies that all match criterion must be met for traffic entering the traffic class to be classified as part of the traffic class.
Router(config)# class-map match-any <i>class-map-name</i>	Specifies that one of the match criterion must be met for traffic entering the traffic class to be classified as part of the traffic class.
Router(config-cmap)# match access-group <i>access-group</i>	Specifies the numbered access list against whose contents packets are checked to determine if they belong to the class. Note Access lists configured with the optional log keyword of the access-list command are not supported when configuring a traffic class. For more information about the access-list command, refer to the <i>Cisco IOS IP Command Reference, Volume 1 of 4: Addressing and Services</i> , Release 12.3 T.
Router(config-cmap)# match any	Specifies that all packets will be matched.
Router config-cmap)# match class-map <i>class-name</i>	Specifies the name of a traffic class to be used as a matching criterion (for nesting traffic class [nested class maps] within one another).
Router(config-cmap)# match cos <i>cos-number</i>	Specifies the CoS value against whose contents packets are checked to determine if they belong to the class.
Router(config-cmap)# match destination-address mac <i>address</i>	Specifies the name of the destination MAC address used as a match criterion against which packets are checked to determine if they belong to the class.
Router(config-cmap)# match input-interface <i>interface-name</i>	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.

Command	Purpose
Router (config-cmap)# match ip dscp <i>ip-dscp-value</i>	Specifies up to eight differentiated services code point (DSCP) values used as match criteria. The value of each service code point is from 0 to 63.
Router (config-cmap)# match ip precedence <i>ip-precedence-value</i>	Specifies up to eight IP Precedence values used as match criteria.
Router (config-cmap)# match ip rtp <i>starting-port-number</i> <i>port-range</i>	Specifies the Real-Time Protocol (RTP) port as the match criterion.
Router (config-cmap)# match mpls experimental <i>mpls-values</i>	Specifies the Multiprotocol Label Switching (MPLS) values to use as match criterion against which packets are checked to determine if they belong to the class.
Router (config-cmap)# match not <i>match-criteria</i>	Specifies a match criterion value that prevents packets from being classified as members of a specified traffic class. All other values of that particular match criterion belong to the class.
Router (config-cmap)# match protocol <i>protocol</i>	Specifies the name of the protocol used as a match criterion against which packets are checked to determine if they belong to the class.
Router (config-cmap)# match qos-group <i>qos-group-value</i>	Specifies the number of the QoS group index used as a match criterion against which packets are checked to determine if they belong to the class.
Router (config-cmap)# match source-address <i>mac</i> <i>address-destination</i>	Specifies the name of the source MAC address used as a match criterion against which packets are checked to determine if they belong to the class.

Creating a Traffic Policy

To configure a traffic policy, use the **policy-map** global configuration command to specify the traffic policy name, and then use the following configuration commands to associate a traffic class, which was configured with the **class-map** command, with one or more QoS features. The traffic class is associated with the traffic policy when the **class** command is used. The **class** command must be issued after entering policy-map configuration mode. After entering the **class** command, you are automatically in policy-map class configuration mode, which is where the QoS policies for the traffic policy are defined.

The QoS policies that can be applied in the traffic policy in policy-map class configuration mode are detailed in the following example:

The syntax of the **policy-map** command is:

```
policy-map policy-name
no policy-map policy-name
```

The syntax of the **class** command is:

```
class class-name
no class class-name
```

All traffic that fails to meet the matching criteria belongs to the default traffic class. The default traffic class is user-configurable, but the default traffic class cannot be deleted.

To configure a traffic policy, use the **policy-map** global configuration command to specify the traffic policy name, and then use the following configuration commands to associate a traffic class, which was configured with the **class-map** command, with one or more QoS policies. The traffic class is associated with the traffic policy when the **class** command is used. The **class** command must be issued immediately

after entering policy-map configuration mode. After entering the **class** command, you are automatically in policy-map class configuration mode, which is where the QoS policies for the traffic policy are defined.

To create a traffic policy, use the following commands beginning in global configuration mode, as needed.

**Note**

This chapter lists some of the command options for the policy-map configuration mode. These command options are not limited to Release 12.2 and can vary among platforms and Cisco IOS releases. Because software is updated frequently, this list of commands might not represent the most updated software command options. For the most current command options for your Cisco IOS software, see the New Feature Documentation index for your particular Cisco IOS software release on Cisco.com.

Command	Purpose
Router (config)# policy-map <i>policy-name</i>	Specifies the name of the traffic policy to configure. Names can be a maximum of 40 alphanumeric characters.
Router (config-pmap)# class <i>class-name</i>	Specifies the name of a predefined traffic class, which was configured with the class-map command, used to classify traffic to the traffic policy.
Router (config-pmap)# class class-default	Specifies the default class to be created as part of the traffic policy.
Router (config-pmap-c)# bandwidth { <i>bandwidth-kbps</i> percent percent }	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.
Router (config-pmap-c)# default <i>command</i>	Sets a command to its default value.
Router (config-pmap-c)# fair-queue <i>number-of-queues</i>	Specifies the number of queues to be reserved for a traffic class.
Router (config-pmap-c)# police <i>bps burst-normal burst-max conform-action action exceed-action action violate-action action</i>	Specifies a maximum bandwidth usage by a traffic class through the use of a token bucket algorithm.
Router (config-pmap-c)# priority { <i>kbps</i> percent percent } [<i>bytes</i>]	Specifies the guaranteed allowed bandwidth, in kbps or percentage, for priority (time-sensitive) traffic. The optional <i>bytes</i> argument controls the size of the burst allowed to pass through the system without being considered in excess of the configured kbps rate.
Router (config-pmap-c)# queue-limit <i>packets</i>	Specifies the maximum number of packets queued for a traffic class (in the absence of the random-detect command).
Router (config-pmap-c)# random-detect	Enables a Weighted Random Early Detection (WRED) drop policy for a traffic class that has a bandwidth guarantee.
Router (config-pmap-c)# set atm-clp	Sets the ATM cell loss priority (CLP) bit to 1.

Command	Purpose
Router (config-pmap-c)# set cos <i>cos-value</i>	Specifies a Class of Service (CoS) value or values to associate with the packet. The number is in the range from 0 to 7.
Router (config-pmap-c)# set ip dscp <i>ip-dscp-value</i>	Specifies the IP DSCP of packets within a traffic class. The IP DSCP value can be any value from 0 to 63.
Router (config-pmap-c)# set ip precedence <i>ip-precedence-value</i>	Specifies the IP Precedence value of packets within a traffic class. The IP Precedence value can be any value from 0 to 7.
Router (config-pmap-c)# set mpls experimental <i>value</i>	Designates the value to which the MPLS bits are set if the packets match the specified policy map.
Router (config-pmap-c)# set qos-group <i>qos-group-value</i>	Specifies a QoS group value to associate with the packet. The QoS group value can be any value from 0 to 99.
Router (config-pmap-c)# service-policy <i>policy-map-name</i>	Specifies the name of a traffic policy to be used as a matching criterion (for nesting traffic policies [hierarchical traffic policies] within one another).
Router (config-pmap-c)# shape { average peak } <i>mean-rate</i> [<i>burst-size</i> [<i>excess-burst-size</i>]]	Shapes traffic to the indicated bit rate according to the algorithm specified.

Attaching a Traffic Policy to an Interface

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

Use the **no** form of the command to detach a traffic policy from an interface. The **service-policy** command syntax is as follows:

```
service-policy {input | output} policy-map-name
no service-policy {input | output} policy-map-name
```



Note

Depending on the platform and Cisco IOS release, a traffic policy can be attached to an ATM permanent virtual circuit (PVC) subinterface, Frame Relay data-link connection identifier (DLCI), or other type of interface.

To attach a traffic policy to an interface, use the following commands in interface configuration mode, as needed:

Command	Purpose
Router(config-if)# service-policy output <i>policy-map-name</i>	Specifies the name of the traffic policy to be attached to the output direction of an interface. The traffic policy evaluates all traffic leaving that interface.
Router(config-if)# service-policy input <i>policy-map-name</i>	Specifies the name of the traffic policy to be attached to the input direction of an interface. The traffic policy evaluates all traffic entering that interface.

**Note**

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.

Verifying the Configuration

To display the information relating to a traffic class or traffic policy, use one of the following commands in EXEC mode, as needed. To display the configuration of a traffic policy and its associated traffic class, use the **show policy-map** EXEC command.

Command	Purpose
Router# show class-map	Displays all traffic class information.
Router# show class-map <i>class-name</i>	Displays the traffic class information for the user-specified traffic class.
Router# show policy-map	Displays all configured traffic policies.
Router# show policy-map <i>policy-map-name</i>	Displays the user-specified traffic policy.
Router# show policy-map <i>interface</i>	Displays configurations and statistics of all input and output policies attached to an interface.
Router# show policy-map interface <i>interface-spec</i>	Displays configuration and statistics of the input and output policies attached to a particular interface.
Router# show policy-map interface <i>interface-spec</i> <i>input</i>	Displays configuration and statistics of the input policy attached to an interface.
Router# show policy-map interface <i>interface-spec</i> <i>output</i>	Displays configuration and statistics of the output policy attached to an interface.
Router# show policy-map [interface [<i>interface-spec</i> [<i>input</i> <i>output</i>] [class <i>class-name</i>]]]	Displays the configuration and statistics of the class name configured in the policy.

Modular QoS CLI Configuration Examples

This section provides the Modular QoS CLI configuration examples:

- [Traffic Classes Defined Example](#)
- [Traffic Policy Created Example](#)
- [Traffic Policy Attached 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 \(Nested Class Maps\) Example](#)
- [Traffic Policy as a QoS Policy \(Hierarchical Traffic Policies\) Example](#)

For information on how to configure the QoS functionality with the Modular QoS CLI, see the section [“Modular QoS CLI Configuration Task List”](#) in this chapter.

Traffic Classes Defined 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
```

Traffic Policy Created Example

In the following example, a traffic policy called policy1 is defined to contain policy specifications for the two classes—class1 and class2. The match criteria for these classes were defined in the traffic classes (see the section [“Creating a Traffic Class”](#) in this chapter).

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)# exit

Router(config-pmap)# class class2
Router(config-pmap-c)# bandwidth 2000
Router(config-pmap)# exit
```

Traffic Policy Attached to an Interface Example

The following example shows how to attach an existing traffic policy (which was created in the preceding [“Traffic Policy Created Example”](#) section) to an interface. After you define a traffic policy with the **policy-map** command, you can attach it to one or more interfaces to specify the traffic policy for those 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 at the input and only one traffic policy attached at the output.

```
Router(config)# interface e1/1
Router(config-if)# service-policy output policy1
Router(config-if)# exit

Router(config)# interface fa1/0/0
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 the user does not configure a default class, packets are still treated as members of the default class. However, by default, the default class has no enabled features. Therefore, packets belonging to a default class with no configured features have no QoS functionality. These packets are then placed into a FIFO queue and forwarded at a rate determined by the available underlying link bandwidth. This FIFO queue is 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 section illustrates the difference between the **class-map match-any** command and the **class-map match-all** command. The **match-any** and **match-all** options 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 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 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 matches 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 traffic class called cisco2, the match criteria are evaluated consecutively until a successful match criterion is located. The packet is first evaluated to determine whether IP protocol can be used as a match criterion. If IP protocol can be used as a match criterion, the packet is matched to traffic class george. If IP protocol is not a successful match criterion, then QoS group 4 is evaluated as a match criterion. Each matching 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 traffic class.

Note that the **class-map match-all** command requires that all of the match criteria must be met in order for the packet to be considered a member of the specified traffic class (a logical AND operator). In the example, protocol IP AND QoS group 4 AND access group 101 have to be successful match criteria. However, only one match criterion must be met for the packet in the **class-map match-any** command to be classified as a member of the traffic class (a logical OR operator). In the example, protocol IP OR QoS group 4 OR access group 101 have to be successful match criteria.

Traffic Class as a Match Criterion (Nested Class Maps) Example

There are two reasons to use the **match class-map** command. One reason is maintenance; if a long traffic class currently exists, using the Traffic Class match criterion is simply easier than retyping the same traffic class configuration.

The more prominent 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 criteria 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 criteria type.

A concept example: 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 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, a user 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 the user 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 1.1.1
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 of traffic class montana 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 1.1.1, 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 1.1.1
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)# exit
```

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 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, 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.11 and FRF.12 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.11 or FRF.12. The value used with the **shape** command is provisioned from the committed information rate (CIR) value from the service provider. For additional information on FRF.11 and FRF.12 PVCs, refer to the *Cisco IOS Voice, Video, and Fax Configuration Guide*.

For additional information on hierarchical traffic policies, refer to the *Cisco IOS Voice, Video, and Fax Configuration Guide*. For information about the **service-policy** command, refer to the *Cisco IOS Quality of Service Solutions Command Reference*.



Part 9: 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>.



Part 10: AutoQoS





AutoQoS — VoIP

The AutoQoS — VoIP feature allows you to automate the delivery of quality of service (QoS) on your network and provides a means for simplifying the implementation and provisioning of QoS for Voice over IP (VoIP) traffic.

Feature Specifications for AutoQoS — VoIP

Feature History

Release	Modification
12.2(15)T	This feature was introduced.

Supported Platforms

Cisco 2600 series, Cisco 2600 XM series, Cisco 2691, Cisco 3620, Cisco 3640, Cisco 3660, Cisco 3725, Cisco 3745, Cisco 7200 series

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- [Prerequisites for AutoQoS — VoIP, page 948](#)
- [Restrictions for AutoQoS — VoIP, page 948](#)
- [Information About AutoQoS — VoIP, page 949](#)
- [How to Configure the AutoQoS — VoIP Feature, page 951](#)
- [Configuration Examples for AutoQoS — VoIP, page 956](#)
- [Additional References, page 961](#)
- [Command Reference, page 963](#)

Prerequisites for AutoQoS — VoIP

- Ensure that no QoS policies (service policies) are attached to the interface. This feature cannot be configured if a QoS policy (service policy) is attached to the interface.
- To include Simple Network Protocol (SNMP) traps (monitored events), the SNMP server must be enabled.

Restrictions for AutoQoS — VoIP

General Restrictions

- The AutoQoS — VoIP feature is supported on the following interfaces, data-link connection identifiers (DLCIs), and permanent virtual circuits (PVCs) only:
 - Serial interfaces with PPP or High-Level Data Link Control (HDLC)
 - Frame Relay DLCIs in point-to-point subinterfaces only
 - ATM PVCs

The AutoQoS — VoIP feature is supported on low-speed ATM PVCs in point-to-point subinterfaces only. The AutoQoS — VoIP feature is supported on high-speed ATM PVCs in any type of subinterface.



Note An ATM PVC is classified as low-speed if its bandwidth is less than or equal to 768 kbps; an ATM PVC is classified as high-speed if its bandwidth is greater than 768 kbps

- Frame Relay-to-ATM Interworking links

Serial Interface Restrictions

- For a serial interface with a low-speed link, Multilink PPP (MLP) is configured automatically. The serial interface must have an IP address. When MLP is configured, this IP address is removed and put on the MLP bundle. To ensure that the traffic goes through the low-speed link, the following conditions must be met:
 - The AutoQoS - VoIP feature must be configured at the *both* ends of the link.
 - The amount of bandwidth configured must be the same on *both* ends of the link.

Frame Relay DLCI Restrictions

- The AutoQoS — VoIP feature cannot be configured on a Frame Relay DLCI if a map class is attached to the DLCI.
- If a Frame Relay DLCI is already assigned to one subinterface, the AutoQoS — VoIP feature cannot be configured from a different subinterface.
- For low-speed Frame Relay DLCIs configured for use on Frame Relay-to-ATM networks, MLP over Frame Relay (MLPoFR) is configured automatically. The subinterface must have an IP address. When MLPoFR is configured, this IP address is removed and put on the MLP bundle. The AutoQoS — VoIP feature must also be configured on the ATM side of the network.
- For low-speed Frame Relay DLCIs with Frame Relay-to-ATM Interworking, the AutoQoS — VoIP feature cannot be configured if a virtual template is already configured for the DLCI.

ATM PVC Restrictions

- For a low-speed ATM PVC, the AutoQoS — VoIP feature cannot be configured if a virtual template is already configured for the ATM PVC.
- For low-speed ATM PVCs, MLP over ATM (MLPoATM) is configured automatically. The subinterface must have an IP address.

When MLPoATM is configured, this IP address is removed and put on the MLP bundle. The AutoQoS — VoIP feature must also be configured on the ATM side of the network.

Information About AutoQoS — VoIP

To configure the AutoQoS — VoIP feature, you need to understand the following concepts:

- [Benefits of AutoQoS — VoIP, page 949](#)
- [Design Considerations, page 950](#)
- [Configurations for the Interface Configurations, Policy Maps, Class Maps, and ACLs, page 951](#)

Benefits of AutoQoS — VoIP

The key benefits of the AutoQoS -VoIP feature include the following:

- Customers can implement the QoS features required for VoIP traffic without an in-depth knowledge of the following underlying technologies:
 - PPP
 - Frame Relay
 - ATM
 - Service policies
 - Link efficiency mechanisms (LEM), such as Link Fragmentation and Interleaving (LFI)
- The AutoQoS — VoIP feature simplifies QoS implementation and speeds up the provisioning of QoS technology over a Cisco network. It reduces human error and lowers training costs. With the AutoQoS — VoIP feature, one command (the **auto qos** command) enables QoS for VoIP traffic across every Cisco router and switch.
- Customers can also use existing Cisco IOS commands to modify the configurations automatically generated by the AutoQoS — VoIP feature as needed to meet specific requirements.
- The Cisco product, CiscoWorks QoS Policy Manager (QPM), can be used in conjunction with the AutoQoS — VoIP feature to provide a centralized, web-based tool to cost effectively manage and monitor network-wide QoS policies. The AutoQoS — VoIP feature together with CiscoWorks QPM, eases QoS implementation, provisioning, and management.

Design Considerations

General QoS Requirements

- Recommended methods and values are configured to meet the QoS requirements for voice traffic.
- The AutoQoS — VoIP feature takes the interface type and bandwidth into consideration when implementing the following QoS features:
 - Classification

Classification is used to differentiate the voice packets from the data packets and handle the voice packets appropriately.
 - Low latency queueing (LLQ) - Priority Queueing (PQ)

The LLQ (specifically, PQ) is applied to the voice packets to meet the latency requirements.
 - Compressed Real-Time Protocol (cRTP)

With cRTP, the 40-byte IP header of the voice packet is reduced from 2 to 4 bytes, thereby reducing voice bandwidth requirements. cRTP must be applied at both ends of a network link.
 - LFI

LFI is used to reduce the jitter of voice packets by preventing voice packets from getting delayed behind large data packets in a queue. LFI must be applied at both ends of a network link.

Bandwidth Implications

- The bandwidth of the serial interface determines the speed of the link. The speed of the link in turn determines the configurations generated by the AutoQoS — VoIP feature.



Note Changing the bandwidth before configuring the AutoQoS — VoIP feature is not recommended.

The AutoQoS — VoIP feature uses the bandwidth at the time the feature is configured. AutoQoS — VoIP does not respond to changes made to bandwidth after the feature is configured.

For example, if the **auto qos voip** command is used to configure the AutoQoS — VoIP feature on an interface with 1000 Kbps, the AutoQoS — VoIP feature generates configurations for high-speed interfaces. However, if the bandwidth is later changed to 500 Kbps, the AutoQoS — VoIP feature will not use the lower bandwidth. The AutoQoS — VoIP feature retains the higher bandwidth and continues to use the generated configurations for high-speed interfaces.

To force the AutoQoS — VoIP feature to use the lower bandwidth (and thus generate configurations for the low-speed interfaces), use the **no auto qos voip** command to remove the AutoQoS — VoIP feature and then reconfigure the feature.

Fragmentation for Frame Relay Networks

- For Frame Relay networks, fragmentation is configured using a delay of 10 milliseconds (ms) and a minimum fragment size of 60 bytes. This ensures that the VoIP packets are not fragmented. However, when the G.711 coder-decoder (codec) is used on low-speed links, the fragment size configured by the AutoQoS — VoIP feature could be smaller than the size of the G.711 VoIP packet.

To solve this potential problem, choose one of the following:

- Change the fragment size to the required value.
- Change the size of the G.711 VoIP packet to a smaller value.

For example, if the AutoQoS — VoIP feature is configured on a Frame Relay DLCI with 128 Kbps, the fragment size configured by the AutoQoS — VoIP feature will be 160 bytes. The size of the G.711 VoIP packet will be 160 bytes, minus the bytes in the packet headers for the layers. The workaround is to either change the fragment size from 160 bytes to 220 bytes or change the size of the G.711 VoIP packet from 160 bytes to 80 bytes.

signalling Protocols

- The AutoQoS — VoIP feature currently identifies the following signalling protocols:
 - H.323
 - H.225 (Unicast only)
 - Session Initiation Protocol (SIP)
 - “Skinny” gateway protocol
 - Media Gateway Control Protocol (MGCP)



Note

Access control lists (ACLs) can be configured to identify any additional signalling protocols that may be needed.

Configurations for the Interface Configurations, Policy Maps, Class Maps, and ACLs

The AutoQoS — VoIP feature automatically creates configurations that are then used for the interface configurations, policy maps, class maps, and ACLs. The interface configurations, policy maps, class maps, and ACLs are created to classify VoIP packets and to provide the appropriate QoS treatment for the network traffic.

This feature also creates interface (or PVC)-specific configurations. These interface (or PVC)-specific configurations are created according to the interface type and the link speed.



Note

Links with bandwidths lower than or equal to 768 kbps are considered low-speed links; links with bandwidths higher than 768 kbps are considered high-speed links.

How to Configure the AutoQoS — VoIP Feature

This section contains the following tasks. Each task is identified as either required or optional.

- [Enabling the AutoQoS — VoIP Feature, page 951](#) (required)
- [Verifying the Configuration, page 955](#) (optional)

Enabling the AutoQoS — VoIP Feature

The only required step for enabling the AutoQoS — VoIP feature is to use the **auto qos voip** command. This command automatically creates configurations for interface configurations, policy maps, class maps, and ACLs. These interface configurations, policy maps, class maps, and ACLs are used to classify VoIP packets and to provide the appropriate QoS for the network traffic.

Prerequisites for Using the auto qos Command

Before using the **auto qos** command at an interface or an ATM PVC, ensure that the following prerequisites have been met:

- Cisco Express Forwarding (CEF) must be enabled at the interface or ATM PVC.
- If the interface or subinterface has a link speed of 768 kbps or lower, configure the primary or secondary IP address of the interface by using the **ip address** command.
- For all interfaces or subinterfaces, configure the amount of bandwidth by using the **bandwidth** command. The amount of bandwidth allocated should be based on the link speed of the interface.
- For an ATM PVC, configure the variable bit rate (VBR) by using either the **vbr-nrt** command or the **vbr-rt** command or configure the constant bit rate (CBR) by using the **cbr** command.

Restrictions for Using the auto qos Command

- The **auto qos voip** command is not supported on subinterfaces.
- Do not change the bandwidth of the interface before using the **auto qos** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **bandwidth** *kilobits*
5. **vbr-nrt** *output-pcr output-scr output-mbs [input-pcr] [input-scr] [input-mbs]*
6. **vbr-rt** *peak-rate average-rate burst*
7. **cbr** *rate*
8. **pvc** [*name*] *vpilvci [ces | ilmi | qsaal | smds]*
9. **ip address** *ip-address mask [secondary]*
10. **frame-relay interface-dlci** *dlci [ietf | cisco] [voice-cir cir] [ppp virtual-template-name]*
11. **auto qos voip** [**trust**] [**fr-atm**]
12. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	<p>interface <i>type number</i></p> <p>Example: Router(config-if)# interface s4/0</p>	<p>Configures an interface (or subinterface) type and enters interface configuration mode.</p> <ul style="list-style-type: none"> Enter the interface type number.
Step 4	<p>bandwidth <i>kilobits</i></p> <p>Example: Router(config-if)# bandwidth 1540</p>	<p>(Optional) Sets a bandwidth value for an interface.</p> <ul style="list-style-type: none"> Enter the bandwidth value in kbps. <p>Note This step applies only to interfaces and subinterfaces. It is not required for ATM PVCs.</p>
Step 5	<p>vbr-nrt <i>output-pcr output-scr output-mbs</i> <i>[input-pcr] [input-scr] [input-mbs]</i></p> <p>Example: Router(config-if)# vbr-nrt 10000 5000 32 20000 10000 64</p>	<p>(Optional) Configures the variable bit rate-nonreal time (VBR-NRT) QoS and specifies the output peak cell rate (PCR), output sustainable cell rate (SCR), and output maximum burst cell size (MBS) for an ATM PVC, PVC range, switched virtual circuit (SVC), virtual circuit (VC) class, or VC bundle member.</p> <ul style="list-style-type: none"> Enter the output PCR, SCR, and MBS. <p>Note This step applies only to ATM PVCs. It is not required for interfaces or subinterfaces.</p>
Step 6	<p>vbr-rt <i>peak-rate average-rate burst</i></p> <p>Example: Router(config-if)# vbr-rt 640 56 80</p>	<p>(Optional) Configures the real-time VBR for Voice over ATM connections.</p> <ul style="list-style-type: none"> Enter the peak information rate (PIR), the average information rate (AIR), and the burst size. <p>Note This step applies only to ATM PVCs. It is not required for interfaces or subinterfaces.</p>
Step 7	<p>cbr <i>rate</i></p> <p>Example: Router(config-if-atm-vc)# cbr 56</p>	<p>(Optional) Configures the CBR for the ATM circuit emulation service (CES) for an ATM PVC.</p> <p>This command can be used in different modes, including ATM-VC configuration mode (for ATM PVCs and SVCs), ATM PVC range configuration mode (for an ATM PVC range), or ATM PVC-in-range configuration mode (for an individual PVC within a PVC range).</p> <ul style="list-style-type: none"> Enter the CBR. <p>Note This step applies only to ATM PVCs. It is not required for interfaces or subinterfaces.</p>
Step 8	<p>pvc [<i>name</i>] <i>vpi/vci</i> [<i>ces</i> <i>ilmi</i> <i>qsaal</i> <i>smds</i>]</p> <p>Example: Router(config-if)# pvc 1/32</p>	<p>(Optional) Creates or assigns a name to an ATM PVC and specifies the encapsulation type on an ATM PVC.</p> <ul style="list-style-type: none"> Enter the ATM network virtual path identifier (VPI) and the ATM network virtual channel identifier (VCI) for the ATM PVC. <p>Note This step applies only to ATM PVCs. It is not required for interfaces or subinterfaces.</p>

	Command or Action	Purpose
Step 9	<p>ip address <i>ip-address mask</i> [secondary]</p> <p>Example: Router(config-if)# ip address 10.10.100.1 255.255.255.0</p>	<p>(Optional) Sets a primary or secondary IP address for an interface.</p> <p>Note Applies only to low-speed interfaces (that is, interfaces with link speeds of 768 kbps or lower.)</p>
Step 10	<p>frame-relay interface-dlci <i>dlci</i> [ietf cisco] [voice-cir <i>cir</i>] [ppp <i>virtual-template-name</i>]</p> <p>Example: Router(config-if)# frame-relay interface-dlci 100</p>	<p>(Optional) Assigns a DLCI to a specified Frame Relay subinterface on the router or access server, or assigns a specific PVC to a DLCI, or applies a virtual template configuration for a PPP session.</p> <ul style="list-style-type: none"> Enter the DLCI number. <p>Note This step applies only to Frame Relay interfaces (either low-speed or high-speed).</p>
Step 11	<p>auto qos voip [trust] [fr-atm]</p> <p>Example: Router(config-if)# auto qos voip or Router(config-fr-dlci)# auto qos voip</p>	<p>Configures the AutoQoS — VoIP feature.</p> <p>Note For low-speed Frame Relay DLCIs interconnected with ATM PVCs in the same network, the fr-atm keyword must be explicitly configured in the auto qos voip command to configure the AutoQoS - VoIP feature properly. That is, the command must be configured as auto qos voip fr-atm.</p>
Step 12	<p>exit</p> <p>Example: Router(config-if)# exit</p>	<p>(Optional) Returns to interface configuration mode.</p>

FAQs and Troubleshooting Tips

Below are answers to frequently asked questions (FAQs) and tips for troubleshooting situations that you may encounter when configuring or using the AutoQoS — VoIP feature.

Why can't I configure the AutoQoS — VoIP feature?

- To configure the feature, CEF must be is enabled. Verify that CEF is enabled on your network.
- Also, the feature cannot be enabled if a service policy is already attached to the interface. Determine whether there is a service policy attached to the interface. If so, remove the service policy from the interface.

Why isn't the AutoQoS — VoIP feature supported on my router?

- The AutoQoS — VoIP feature is supported only on the IP Plus image for lower-end platforms. Verify that you have the IP Plus image installed on your router.

Why are some of my QoS configurations still present after I disable the AutoQoS — VoIP feature?

- You have to manually disable any QoS configurations that were modified by the AutoQoS — VoIP feature.

Why did my low-speed network link go down when I enabled the AutoQoS — VoIP feature?

- Ensure that AutoQoS — VoIP is enabled on *both* sides of the network link.

Why can't I establish an end-to-end connection on the Frame Relay link?

- Check the bandwidth on both sides of the Frame Relay link. The bandwidth on both sides of the link *must be the same*, otherwise a fragmentation size mismatch occurs and a connection cannot be established.

**Note**

For more help, see the [“Technical Assistance”](#) section.

What to Do Next

If the interface configurations, policy maps, class maps, and ACLs created (on the basis of the configurations created by the AutoQoS - VoIP feature) do not meet the needs of your network, the interface configurations, policy maps, class maps, and ACLs can be modified using the appropriate Cisco IOS commands.

**Note**

While you can modify the interface configurations, policy maps, class maps, and ACLs, they may not be removed properly when the AutoQoS — VoIP feature is disabled using the **no auto qos** command. You may need to manually remove any modified interface configurations, policy maps, class maps, and ACLs. For more information about the **no auto qos** command, see the [“Command Reference”](#) section.

Verifying the Configuration

The AutoQoS — VoIP feature automatically generates configurations that, in turn, are used to create interface configurations, policy maps, class maps, and ACLs. These interface configurations, policy maps, class maps, and ACLs configure the QoS features on your network.

To verify the configuration (that is, the contents of the interface configurations, policy maps, class maps, and ACLs), use the following commands:

SUMMARY STEPS

1. **enable**
2. **show auto qos [interface [interface type]]**
3. **show policy-map interface [interface type]**
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	show auto qos [interface [<i>interface type</i>]] Example: Router# show auto qos interface s4/0	(Optional) Displays the interface configurations, policy maps, class maps, and ACLs created on the basis of automatically generated configurations. <ul style="list-style-type: none"> The interface configurations, policy maps, class maps, and ACLs can be displayed for a specific interface or all interfaces.
Step 3	show policy-map interface [<i>interface type</i>] Example: Router# show policy-map interface s4/0	(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. <ul style="list-style-type: none"> The packet statistics can be displayed for a specific interface, subinterface, PVC, or all interfaces, subinterfaces, or PVCs.
Step 4	exit Example: Router# exit	(Optional) Exits privileged EXEC mode.

Configuration Examples for AutoQoS — VoIP

This section provides the following configuration examples:

- [Configuring the AutoQoS — VoIP Feature Examples, page 956](#)
- [Verifying the AutoQoS — VoIP Feature Configuration Examples, page 958](#)

Configuring the AutoQoS — VoIP Feature Examples

When the **auto qos voip** command is used to configure the AutoQoS — VoIP feature, configurations are generated. These configurations are then used to create interface configurations, policy maps, class maps, and aCLs.

This section contains the following examples of configuring the AutoQoS — VoIP feature on the interfaces, PVCs, and links.

Configuring the AutoQoS — VoIP Feature on a High-Speed Serial Interface Example

In this example, the AutoQoS — VoIP feature is configured on the high-speed serial interface s1/2.

```
Router> enable
Router# configure terminal
Router(config)# interface s1/2
Router(config-if)# bandwidth 1540
Router(config-if)# auto qos voip
Router(config-if)# exit
```

Configuring the AutoQoS — VoIP Feature on a Low-Speed Serial Interface Example

In this example, the AutoQoS — VoIP feature is configured on the low-speed serial interface s1/3.

```
Router# configure terminal
Router(config)# interface s1/3
Router(config-if)# bandwidth 512
Router(config-if)# ip address 10.10.100.1 255.255.255.0
Router(config-if)# auto qos voip
Router(config-if)# exit
```

Configuring the AutoQoS — VoIP Feature on High-Speed Frame Relay Interfaces Example

In this example, the AutoQoS — VoIP feature is configured on the high-speed serial point-to-point Frame Relay subinterface s4/1.2.

```
Router> enable
Router# configure terminal
Router(config)# interface s4/1.2 point-to-point
Router(config-if)# bandwidth 1540
Router(config-if)# frame-relay interface-dlci 100
Router(config-fr-dlci)# auto qos voip
Router(config-if)# exit
```

Configuring the AutoQoS — VoIP Feature on Low-Speed Frame Relay Interfaces Example

In this example, the AutoQoS — VoIP feature is configured on the low-speed point-to-point Frame Relay subinterface s4/2.1.

```
Router# configure terminal
Router(config)# interface s4/2.1 point-to-point
Router(config-if)# bandwidth 512
Router(config-if)# ip address 10.10.100.1 255.255.255.0
Router(config-if)# frame-relay interface-dlci 100
Router(config-fr-dlci)# auto qos voip
Router(config-if)# exit
```

Configuring the AutoQoS — VoIP Feature on a High-Speed ATM PVC Example

In this example, the AutoQoS — VoIP feature is configured on the high-speed point-to-point ATM PVC ATM5/0.1.

```
Router# configure terminal
Router(config)# interface ATM5/0.1 point-to-point
Router(config-if)# pvc 1/32
Router(config-if)# vbr-nrt 1540 1540
Router(config-if)# auto qos voip
Router(config-if)# exit
```

Configuring the AutoQoS — VoIP Feature on a Low-Speed ATM PVC Example

In this example, the AutoQoS — VoIP feature is configured on a low-speed point-to-point ATM PVC ATM5/0.2.

```
Router# configure terminal
Router(config)# interface ATM5/0.2 point-to-point
Router(config-if)# ip address 10.10.100.1 255.255.255.0
Router(config-if)# pvc 1/32
Router(config-if)# vbr-nrt 512 512
Router(config-if)# auto qos voip
Router(config-if)# exit
```

Configuring the AutoQoS — VoIP Feature for Frame Relay-to-ATM Interworking Example

In this example, the AutoQoS — VoIP feature is configured for Frame Relay-to-ATM Interworking. The AutoQoS — VoIP feature is configured on the serial point-to-point subinterface s1/3.1.

```
Router# configure terminal
Router(config)# interface s1/3.1 point-to-point
Router(config-if)# bandwidth 512
Router(config-if)# ip address 10.10.100.1 255.255.255.0
Router(config-if)# frame-relay interface-dlci 100
Router(config-if)# auto qos voip fr-atm
Router(config-if)# exit
```

In this configuration, the optional **fr-atm** keyword is used to enable the AutoQoS — VoIP feature for the Frame Relay-to-ATM Interworking.



Note

The ATM-to-ATM side of the network needs no special configuration to distinguish it from the ATM-to-Frame Relay side of the network.

Verifying the AutoQoS — VoIP Feature Configuration Examples

When the **auto qos voip** command is used to configure the AutoQoS — VoIP feature, configurations are generated. These configurations are then used to create interface configurations, policy maps, class maps, and ACLs. The **show auto qos interface** command can be used to verify the contents of the interface configurations, policy maps, class maps, and ACLs.

This section contains the following sample output of the **show auto qos interface** command for interfaces, PVCs, and links.



Note

The **show auto qos interface** command output displays only those configurations created by the AutoQoS - VoIP feature.

Sample show auto qos interface Command Output for a High-Speed Serial Interface

The following is sample output of the **show auto qos** command for a high-speed serial interface:

```
Router# show auto qos interface s6/0

Serial6/0 -
!
interface Serial6/0
  service-policy output AutoQoS-Policy-UnTrust
```

Sample show auto qos interface Command Output for a Low-Speed Serial Interface

The following is sample output of the **show auto qos interface** command for a low-speed serial interface:

```
Router# show auto qos interface s6/0

Serial6/0 -
!
interface Serial6/0
  no ip address
  encapsulation ppp
  no fair-queue
  ppp multilink
  multilink-group 2001100126
!
interface Multilink2001100126
  bandwidth 512
  ip address 10.10.100.1 255.255.255.0
  service-policy output AutoQoS-Policy-UnTrust
  ppp multilink
  ppp multilink fragment-delay 10
  ppp multilink interleave
  ip rtp header-compression iphc-format
```

Sample show auto qos Interface Command Output for a High-Speed Frame Relay Interface

The following is sample output of the **show auto qos** command for a high-speed Frame Relay interface:

```
Router# show auto qos interface s6/1.1

Serial6/1.1: DLCI 100 -
!
interface Serial6/1
  frame-relay traffic-shaping
!
interface Serial6/1.1 point-to-point
  frame-relay interface-dlci 100
  class AutoQoS-VoIP-FR-Serial6/1-100
!
map-class frame-relay AutoQoS-VoIP-FR-Serial6/1-100
  frame-relay cir 1540000
  frame-relay bc 15400
  frame-relay be 0
  frame-relay mincir 1540000
  service-policy output AutoQoS-Policy-UnTrust
```



Note

The output of the **show autoqos interface** command for high-speed Frame Relay interfaces is similar to the output for low-speed Frame Relay interfaces. The only difference is that Frame Relay Fragmentation and cRTP are not configured for high-speed Frame Relay interfaces.

Sample show auto qos interface Command Output for a Low-Speed Frame Relay Interface

The following is sample output of the **show auto qos** command for a low-speed Frame Relay interface:

```
Router# show auto qos interface s6/1.1

Serial6/1.1: DLCI 100 -
!
interface Serial6/1
  frame-relay traffic-shaping
!
```

```

interface Serial6/1.1 point-to-point
 frame-relay interface-dlci 100
   class AutoQoS-VoIP-FR-Serial6/1-100
 frame-relay ip rtp header-compression
!
map-class frame-relay AutoQoS-VoIP-FR-Serial6/1-100
 frame-relay cir 512000
 frame-relay bc 5120
 frame-relay be 0
 frame-relay mincir 512000
 service-policy output AutoQoS-Policy-UnTrust
 frame-relay fragment 640

```

Sample show auto qos interface Command Output for a High-Speed ATM PVC

The following is sample output of the **show auto qos** command for a high-speed ATM PVC:

```

Router# show auto qos interface a2/0.1

ATM2/0.1: PVC 1/100 -
!
interface ATM2/0.1 point-to-point
 pvc 1/100
   tx-ring-limit 3
   service-policy output AutoQoS-Policy-UnTrust

```

Sample show auto qos interface Command Output for a Low-Speed ATM PVC

The following is sample output of the **show auto qos** command for a low-speed ATM PVC:

```

Router# show auto qos interface a2/0.1

ATM2/0.1: PVC 1/100 -
!
interface ATM2/0.1 point-to-point
 pvc 1/100
   tx-ring-limit 3
   encapsulation aal5mux ppp Virtual-Template200
!
interface Virtual-Template200
 bandwidth 512
 ip address 10.10.100.1 255.255.255.0
 service-policy output AutoQoS-Policy-UnTrust
 ppp multilink
 ppp multilink fragment-delay 10
 ppp multilink interleave

```

Sample show auto qos interface Command Output for Frame Relay-to-ATM Interworking Links

The following is sample output of the **show auto qos** command for Frame Relay-to-ATM Interworking links:

```

Router# show auto qos interface s6/1.1

Serial6/1.1: DLCI 100 -
!
interface Serial6/1
 frame-relay traffic-shaping
!
interface Serial6/1.1 point-to-point
 frame-relay interface-dlci 100 ppp Virtual-Template200
   class AutoQoS-VoIP-FR-Serial6/1-100
!

```

```

interface Virtual-Template200
  bandwidth 512
  ip address 10.10.100.1 255.255.0.0
  service-policy output AutoQoS-Policy-UnTrust
  ppp multilink
  ppp multilink fragment-delay 10
  ppp multilink interleave
!
map-class frame-relay AutoQoS-VoIP-FR-Serial6/1-100
  frame-relay cir 512000
  frame-relay bc 5120
  frame-relay be 0
  frame-relay mincir 512000

```

Additional References

For additional information related to AutoQoS — VoIP, refer to the following references:

- [Related Documents, page 961](#)
- [Standards, page 962](#)
- [MIBs, page 962](#)
- [RFCs, page 962](#)
- [Technical Assistance, page 963](#)

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	<i>Cisco IOS Quality of Service Solutions Command Reference</i> , Release 12.2 T
QoS concepts and features applicable to VoIP; high-level examples for configuring these features in different network environments	<i>Quality of Service for Voice over IP</i> , Cisco Integrated Networking Solutions document
LFI and cRTP	“Link Efficiency Mechanisms” section of the <i>Cisco IOS Quality of Service Solutions Configuration Guide</i> , Release 12.2
Packet classification	“Classification” section of the <i>Cisco IOS Quality of Service Solutions Configuration Guide</i> , Release 12.2
LLQ	“Congestion Management” section of the <i>Cisco IOS Quality of Service Solutions Configuration Guide</i> , Release 12.2
Service policies	“Modular Quality of Service Command-Line Interface” section of the <i>Cisco IOS Quality of Service Solutions Configuration Guide</i> , Release 12.2
Frame Relay and ATM commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	<i>Cisco IOS Wide-Area Networking Command Reference</i> , Release 12.2 T
Frame-Relay DLCIs, ATM PVCs, Frame Relay-to-ATM Interworking, MLPoFR, and other information about Frame Relay networks	<i>Cisco IOS Wide-Area Networking Configuration Guide</i> , Release 12.2

Related Topic	Document Title
MLP	<i>Cisco IOS Dial Technologies Configuration Guide</i> , Release 12.2
CEF	<i>Cisco IOS Switching Services Configuration Guide</i> , Release 12.2
SNMP	<i>Cisco IOS Configuration Fundamentals Configuration Guide</i> , Release 12.2
CiscoWorks QPM	Product information available online at Cisco.com

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 standards has not been modified by this feature.	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

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

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

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 new commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at <http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm>.

New Commands

- **auto qos voip**
- **show auto qos**



AutoQoS for the Enterprise

The AutoQoS for the Enterprise feature automates the deployment of quality of service (QoS) policies in a general business environment, particularly for midsize companies and branch offices of larger companies. Existing QoS policies may be present during the first configuration phase of this feature, that is, during the Auto-Discovery (data collection) phase. However, any existing QoS policies must be removed before the AutoQoS-generated policies are applied during the second configuration phase of this feature.

Feature History for the AutoQoS for the Enterprise Feature

Feature History

Release	Modification
12.3(7)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- [Prerequisites for the AutoQoS for the Enterprise Feature, page 965](#)
- [Restrictions for the AutoQoS for the Enterprise Feature, page 966](#)
- [Information About the AutoQoS for the Enterprise Feature, page 967](#)
- [How to Configure the AutoQoS for the Enterprise Feature, page 975](#)
- [Configuration Examples for the AutoQoS for the Enterprise Feature, page 983](#)
- [Additional References, page 988](#)
- [Command Reference, page 990](#)

Prerequisites for the AutoQoS for the Enterprise Feature

- Ensure that no QoS policies (service policies) are attached to the interface. This feature cannot be configured if a QoS policy is attached to the interface.

- To include Simple Network Management Protocol (SNMP) traps (monitored events), the SNMP server must be enabled.

Restrictions for the AutoQoS for the Enterprise Feature

General Restrictions

- The AutoQoS for the Enterprise feature is supported on the following interfaces, data-link connection identifiers (DLCIs), and permanent virtual circuits (PVCs) only:
 - Serial interfaces with PPP or High-Level Data Link Control (HDLC)
 - Frame Relay DLCIs in point-to-point subinterfaces only
 - ATM PVCs

The AutoQoS for the Enterprise feature is supported on low-speed and high-speed ATM PVCs in point-to-point subinterfaces.



Note An ATM PVC is classified as low speed if its bandwidth is less than or equal to 768 Kbps; an ATM PVC is classified as high speed if its bandwidth is greater than 768 Kbps.

- Frame Relay-to-ATM Interworking links

Serial Interface Restrictions

- For a serial interface with a low-speed link, Multilink PPP (MLP) is configured automatically. The serial interface must have an IP address. When MLP is configured, this IP address is removed and put on the MLP bundle. To ensure that the traffic goes through the low-speed link, the following conditions must be met:
 - The AutoQoS for the Enterprise feature must be configured at the *both* ends of the link.
 - The amount of bandwidth configured must be the same on *both* ends of the link.

Frame Relay DLCI Restrictions

- The AutoQoS for the Enterprise feature cannot be configured on a Frame Relay DLCI if a map class is attached to the DLCI.
- If a Frame Relay DLCI is already assigned to one subinterface, the AutoQoS for the Enterprise feature cannot be configured from a different subinterface.
- For low-speed Frame Relay DLCIs configured for use on Frame Relay-to-ATM networks, MLP over Frame Relay (MLPoFR) is configured automatically. The subinterface must have an IP address. When MLPoFR is configured, this IP address is removed and put on the MLP bundle. The AutoQoS for the Enterprise feature must also be configured on the ATM side of the network.
- For low-speed Frame Relay DLCIs with Frame Relay-to-ATM Interworking, the AutoQoS for the Enterprise feature cannot be configured if a virtual template is already configured for the DLCI.

ATM PVC Restrictions

- For a low-speed ATM PVC, the AutoQoS for the Enterprise feature cannot be configured if a virtual template is already configured for the ATM PVC.

- For low-speed ATM PVCs, MLP over ATM (MLPoATM) is configured automatically. The subinterface must have an IP address.

When MLPoATM is configured, this IP address is removed and put on the MLP bundle. The AutoQoS for the Enterprise feature must also be configured on the ATM side of the network.

Information About the AutoQoS for the Enterprise Feature

To configure the AutoQoS for the Enterprise feature, you should understand the following concepts:

- [Benefits of the AutoQoS for the Enterprise Feature, page 967](#)
- [Design Considerations, page 968](#)
- [Configuration Phases, page 969](#)

Benefits of the AutoQoS for the Enterprise Feature

The key benefits of the AutoQoS for the Enterprise feature include the following:

- Customers can implement the QoS features required for voice, video, and data traffic without an in-depth knowledge of the following underlying technologies:
 - PPP
 - Frame Relay
 - ATM
 - Service policies
 - Link efficiency mechanisms (LEM), such as Link Fragmentation and Interleaving (LFI)
- The AutoQoS for the Enterprise feature simplifies QoS implementation and speeds up the provisioning of QoS technology over a Cisco network. It reduces human error and lowers training costs.
- The AutoQoS for the Enterprise feature creates class maps and policy maps on the basis of Cisco experience and “best practices” methodology.
- Customers can also use existing Cisco IOS commands to modify the configurations, automatically generated by the AutoQoS for the Enterprise feature, as needed to meet specific requirements.

Design Considerations

General QoS Requirements

- Recommended methods and values are configured to meet the QoS requirements for voice traffic.
- The AutoQoS for the Enterprise feature takes the interface type and bandwidth into consideration when implementing the following QoS features:
 - Low latency queueing (LLQ) — Priority Queueing (PQ)

The LLQ (specifically, PQ) is applied to the voice packets to meet the latency requirements.
 - Compressed Real-Time Protocol (cRTP)

With cRTP, the 40-byte IP header of the voice packet is reduced from 2 to 4 bytes, thereby reducing voice bandwidth requirements. cRTP must be applied at both ends of a network link.
 - LFI

LFI is used to reduce the jitter of voice packets by preventing voice packets from getting delayed behind large data packets in a queue. LFI must be applied at both ends of a network link.

Bandwidth Implications

- The bandwidth of the serial interface determines the speed of the link. The speed of the link, in turn, determines the configurations generated by the AutoQoS for the Enterprise feature.



Note Changing the bandwidth during configuring the AutoQoS for the Enterprise feature is not recommended.

The AutoQoS for the Enterprise feature uses the bandwidth that is allocated at the time the feature is configured. The AutoQoS for the Enterprise feature does not respond to changes made to bandwidth after the feature is configured.

For example, if the **auto qos** command is used to configure the AutoQoS for the Enterprise feature on an interface with 1000 Kbps, the AutoQoS for the Enterprise feature generates configurations for high-speed interfaces. However, if the bandwidth is later changed to 500 Kbps, the AutoQoS for the Enterprise feature will not use the lower bandwidth. The AutoQoS for the Enterprise feature retains the higher bandwidth and continues to use the generated configurations for high-speed interfaces.

To force the AutoQoS for the Enterprise feature to generate configurations for the low-speed interfaces, perform the following tasks:

5. Use the **no auto qos** command to remove the AutoQoS for the Enterprise feature.
6. Use the **no auto discovery qos** command to stop the Auto-Discovery (data collection) configuration phase.
7. Use the **auto discovery qos** command to resume the Auto-Discovery (data collection) phase.
8. Use the **auto qos** command to begin the AutoQoS template generation and installation configuration phase.

Fragmentation for Frame Relay Networks

- For Frame Relay networks, fragmentation is configured using a delay of 10 milliseconds (ms) and a minimum fragment size of 60 bytes. This configuration ensures that the VoIP packets are not fragmented. However, when the G.711 coder-decoder (codec) is used on low-speed links, the fragment size configured by the AutoQoS for the Enterprise feature could be smaller than the size of the G.711 Voice over IP (VoIP) packet.

To solve this potential problem, choose one of the following:

- Change the fragment size to the required value.
- Change the size of the G.711 VoIP packet to a smaller value.

For example, if the AutoQoS for the Enterprise feature is configured on a Frame Relay DLCI with 128 Kbps, the fragment size configured by the AutoQoS for the Enterprise feature will be 160 bytes. The size of the G.711 VoIP packet will be 160 bytes, minus the bytes in the packet headers for the layers. The workaround is to either change the fragment size from 160 bytes to 220 bytes or change the size of the G.711 VoIP packet from 160 bytes to 80 bytes.

Configuration Phases

The AutoQoS for the Enterprise feature consists of two configuration phases, completed in the following order:

- Auto-Discovery (data collection)

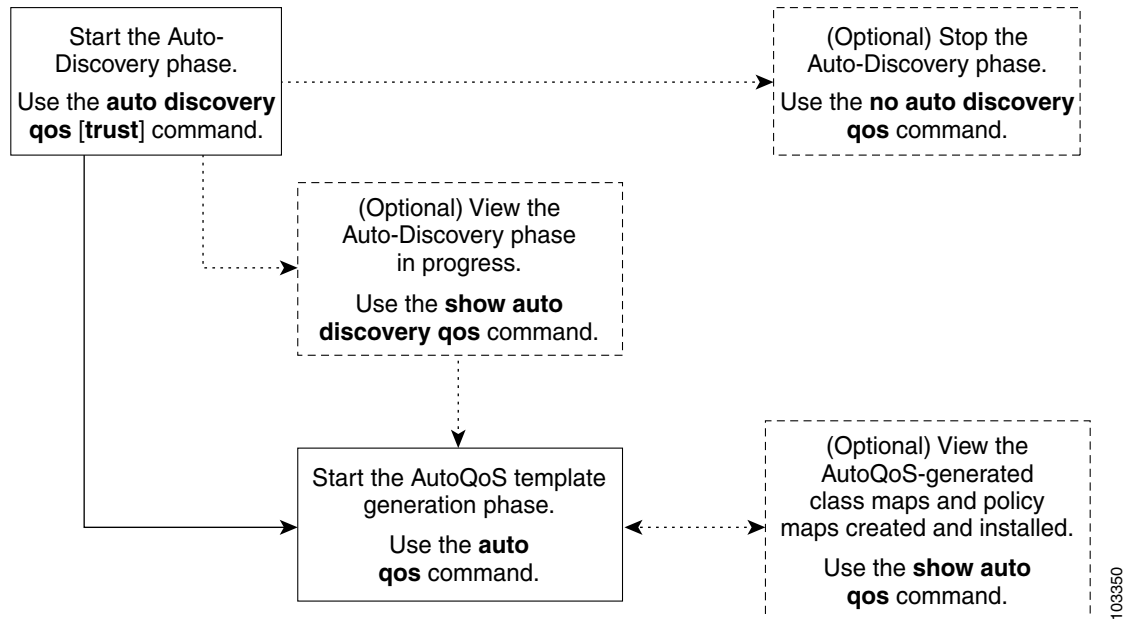
The Auto-Discovery phase uses network-based application recognition (NBAR)-based protocol discovery to detect the applications on the network and performs statistical analysis on the network traffic.

- AutoQoS template generation and installation

This phase generates templates from the data collected during the Auto-Discovery phase and installs the templates on the interface. Then these templates are used as the basis for creating the class maps and policy maps for your network. After the class maps and policy maps are created, they are then installed on the interface.

Figure 52 below illustrates the top-level processes for configuring the AutoQoS for Enterprise feature. The dotted lines indicate optional processes.

Figure 52 Top-Level Processes for Configuring the AutoQoS for the Enterprise Feature



First, start the Auto-Discovery (data collection) phase by using the **auto discovery qos** command. Note the following points about the Auto-Discovery phase:

- If you want to stop the Auto-Discovery phase, use the **no auto discovery qos** command. This command stops data collection and removes any data collection reports that have been generated.
- If you want to view the Auto-Discovery phase in progress, use the **show auto discovery qos** command. This command displays the results of the data collected during the Auto-Discovery phase.

Second, start the AutoQoS template generation phase by using the **auto qos** command. This phase generates templates from the data collected during the Auto-Discovery phase. It then uses those templates as the basis for creating and installing the class maps and policy maps for your network.



Note After the **auto qos** command has finished creating and installing the templates, creating the class maps and policy maps, and installing the class maps and policy maps on the interface, you can view the class maps and policy maps by using the **show auto qos** command.

Detailed information about the Auto-Discovery phase and the AutoQoS template generation phase is provided below.

Auto-Discovery (Data Collection) Phase

The Auto-Discovery (data collection) phase uses NBAR to detect network applications as they arrive at an interface, collect data from the offered traffic, and perform statistical analysis.

The data collected should be a representative sampling of the volume and type of voice, video, and data on your network. Therefore, the amount of time devoted to data collection varies from network to network. Run the Auto-Discovery phase for as long as necessary. The length of time needed can vary, depending on the volume and nature of traffic on your network.

AutoQoS Template Generation and Installation Phase

This phase generates templates from the data collected during the Auto-Discovery phase and installs the templates on the interface. Then these templates are used as the basis for creating the class maps and policy maps for your network. After the class maps and policy maps are created, they are then installed on the interface.

During this phase, the AutoQoS for the Enterprise also assigns the appropriate bandwidth amounts and sets the appropriate scheduling parameters for the network traffic.

Class-Map Templates

The AutoQoS for the Enterprise feature creates a number of class-map templates, used for the following purposes:

- To classify applications and map them to classes for DiffServ per-hop behavior (PHB) mapping.
- To define the class-based QoS policy templates

AutoQoS Classes

The AutoQoS for the Enterprise feature defines 10 AutoQoS classes, designed to accommodate various enterprise applications. [Table 19](#) lists the AutoQoS class name, the type of traffic defined for the class, and the differentiated services code point (DSCP) value for the type of traffic, if applicable.

Table 19 *AutoQoS for the Enterprise Feature Class Definitions*

AutoQoS Class Name	Traffic Type	DSCP Value
IP Routing	Network control traffic, such as routing protocols	CS6
Interactive Voice	Inactive voice-bearer traffic	EF
Interactive Video	Interactive video data traffic	AF41
Streaming Video	Streaming media traffic	CS4
Telephony signalling	Telephony signalling and control traffic	CS3
Transactional/Interactive	Database applications transactional in nature	AF21
Network Management	Network management traffic	CS2
Bulk Data	Bulk data transfers; web traffic; general data service	AF11

Table 19 *AutoQoS for the Enterprise Feature Class Definitions (continued)*

AutoQoS Class Name	Traffic Type	DSCP Value
Scavenger	Casual entertainment; rogue traffic; traffic in this category is given less-than-best-effort treatment	CS1
Best Effort	Default class; all non-critical traffic; HTTP; all miscellaneous traffic	0

These classes are used with the modular quality of service (QoS) command-line interface (MQC) to configure class maps, once the classification (match) criteria are determined. The match criteria can be configured using the appropriate **match protocol** commands.

These classes are also chosen to meet the scheduling requirement in compliance with the DiffServ recommendations. Each class will be associated with an egress (output) queue. The applications mapped to a class will be put into the same queue and receive the same (weighted) queueing scheduling.

**Note**

The actual number of queues created corresponds to the number of applications (and then classes) discovered during AutoQoS-Discovery.

AutoQoS Classification Using NBAR

NBAR is the classification mechanism for the AutoQoS for the Enterprise feature. NBAR is a Cisco product that classifies network traffic using information about the application such as protocol type, URL, and dynamically assigned ports.

All the NBAR-supported applications are mapped to the AutoQoS classes described in the “[AutoQoS Classes](#)” section.

The AutoQoS for the Enterprise feature provides static default mapping rules used to build the AutoQoS class-map templates. [Table 20](#) lists each AutoQoS class, the application to which it is mapped, and the Cisco IOS **match protocol** command used in a policy map to establish the mapping.

Table 20 *AutoQoS Classes, Applications, and match protocol Command*

AutoQoS Class	Application	match protocol Command
Interactive Voice	VoIP bearer	match protocol rtp voice
Interactive Video	Video conference	match protocol rtp video
Telephony signalling	Voice and video signalling and control	match protocol rtcp match protocol h323
Streaming Video	Streaming video	match protocol cuseeme match protocol netshow match protocol realaudio match protocol streamwork match protocol vdlive

Table 20 *AutoQoS Classes, Applications, and match protocol Command (continued)*

AutoQoS Class	Application	match protocol Command
Transactional/Interactive	Database	match protocol sap match protocol sqlnet match protocol sqlserver match protocol citrix match protocol notes
	Interactive sessions	match protocol telnet match protocol secure-telnet match protocol xwindows match protocol ssh match protocol finger
	Other enterprise applications	match protocol novadigm match protocol pcan anywhere
Bulk Data	File transfer	match protocol ftp match protocol secure-ftp match protocol nntp match protocol secure-ntp match protocol printer
	Email and groupware	match protocol exchange match protocol smtp match protocol pop3 match protocol secure-pop3
Scavenger	Peer-to-peer file transfer	match protocol napster match protocol fastrack match protocol gnutella

[Table 21](#) lists the best-effort AutoQoS class (Best Effort), the application category for this class, and the NBAR protocols associated with this class.

Table 21 *Best Effort Class, Application Categories, and Associated NBAR Protocols*

AutoQoS Class	Application Category	NBAR Protocols
Best Effort	Known	http, secure-http, gopher, nfs, sunrpc, ntp, rcmd
Note The class class-default does not need a match statement in the policy map.	Unknown	All applications not identified by NBAR

**Note**

NBAR allows new applications to be defined and added to the network by using different tools such as a Packet Description Language Module (PDLM). The AutoQoS class mapping can not be predetermined for these applications. Therefore, these new applications will be viewed as unknown and put into the AutoQoS default (that is, Best Effort) class.

Table 22 lists the AutoQoS network routing protocol class (IP Routing), the application category for this class, and the NBAR protocols associated with this class.

Table 22 IP Routing Class, Application Categories, and Associated NBAR Protocols

AutoQoS Class	Application Category	NBAR Protocols
IP Routing Note The Type of Service (ToS) byte is always marked as 0x11000000.	Network routing and signalling	All supported network routing and signalling protocols. The list of NBAR supported protocols includes bgp, eigrp, rip, rsvp.

Table 23 lists each AutoQoS management class (Network Management), the application to which it is mapped, and the Cisco IOS **match protocol** command used in a policy map to establish the mapping.

Table 23 Network Management Class, Application Categories, and match protocol Command

AutoQoS Class	Application Category	match protocol Command
Network Management	Network Management	match protocol snmp match protocol syslog match protocol dhcp match protocol dns match protocol ldap match protocol secure-ldap match protocol socks match protocol imap match protocol secure-imap match protocol kerberos

These AutoQoS classes and mapping scheme are used as the basic building blocks for packet classification. If these classes and this mapping scheme are not correct for your particular network, you can change them using the standard Cisco IOS commands and the MQC.

Trusted Boundary

A trusted boundary is the location in the network where the QoS marking is established. AutoQoS can be enabled with the **trust** keyword of the **auto discovery qos** command when the data collection phase is enabled.

The AutoQoS classification for trusted marking will use DSCP match statements specified in Table 24.

When a marking is trusted, the following DSCP values are used in the match statements in the policy maps.

Table 24 *DSCP Values in Match Statements for Trusted Boundaries*

AutoQoS Class	DSCP Values in Match Statements
IP Routing	match ip dscp cs6
Interactive Voice	match ip dscp ef
Interactive Video	match ip dscp af41
Streaming Video	match ip dscp cs4
Telephony signalling	match ip dscp cs3
Transactional/Interactive	match ip dscp af21
Network Management	match ip dscp cs2
Bulk Data	match ip dscp af11
Scavenger	match ip dscp cs1

Policy-Map Templates

The policy-map templates created by the AutoQoS for the Enterprise feature are used to define the following three components:

- Queues scheduling
- Minimum guaranteed bandwidth
- Default Weighted Random Early Detection (WRED) for the applicable classes

These components are designed according to “best practice” recommendations and include QoS features for specific link types, such as low- and high-speed Frame Relay DLCIs.

How to Configure the AutoQoS for the Enterprise Feature

This section contains the following tasks. Each task is identified as either required or optional.

- [Enabling the Auto-Discovery Phase, page 975](#) (required)
- [Enabling the AutoQoS Template Generation and Installation Phase, page 978](#) (required)
- [Verifying the Configuration, page 981](#) (optional)

Enabling the Auto-Discovery Phase

The Auto-Discovery phase uses NBAR to detect network applications and protocols as they leave an interface, collect data from the offered traffic, and perform statistical analysis. The information collected will be used to build the AutoQoS templates. These templates are then used to create the appropriate class maps and policy maps described in the “[AutoQoS Template Generation and Installation Phase](#)” section.

To enable the Auto-Discovery phase, use the **auto discovery qos** command.

Prerequisites for Using the auto discovery qos Command

Before using the **auto discovery qos** command at an interface or an ATM PVC, ensure that the following prerequisites have been met:

- Cisco Express Forwarding (CEF) must be enabled.
- If the interface or subinterface has a link speed of 768 kbps or lower, configure the primary or secondary IP address of the interface by using the **ip address** command.
- For all interfaces or subinterfaces, configure the amount of bandwidth by using the **bandwidth** command. The amount of bandwidth allocated should be based on the link speed of the interface.
- For an ATM PVC, configure the variable bit rate (VBR) by using either the **vbr-nrt** command or the **vbr-rt** command or configure the constant bit rate (CBR) by using the **cbr** command.

Restrictions for Using the auto discovery qos Command

- The **auto discovery qos** command is not supported on subinterfaces.
- Do not change the bandwidth of the interface when using the **auto discovery qos** command.
- All previously attached policies must be removed from the interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **bandwidth** *kilobits*
5. **vbr-nrt** *output-pcr output-scr output-mbs [input-pcr] [input-scr] [input-mbs]*
6. **vbr-rt** *peak-rate average-rate burst*
7. **cbr** *rate*
8. **pvc** [*name*] *vpilvci [ces | ilmi | qsaal | smds]*
9. **ip address** *ip-address mask [secondary]*
10. **frame-relay interface-dlci** *dlci [ietf | cisco] [voice-cir cir] [ppp virtual-template-name]*
11. **auto discovery qos** [*trust*]
12. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>enable</p> <p>Example: Router> enable</p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	<p>configure terminal</p> <p>Example: Router# configure terminal</p>	<p>Enters global configuration mode.</p>
Step 3	<p>interface <i>type number</i></p> <p>Example: Router(config)# interface s4/0</p>	<p>Configures an interface (or subinterface) type and enters interface configuration mode.</p> <ul style="list-style-type: none"> Enter the interface type number.
Step 4	<p>bandwidth <i>kilobits</i></p> <p>Example: Router(config-if)# bandwidth 1540</p>	<p>(Optional) Sets a bandwidth value for an interface.</p> <ul style="list-style-type: none"> Enter the bandwidth value in Kbps. <p>Note This step applies only to interfaces and subinterfaces. It is not required for ATM PVCs.</p>
Step 5	<p>vbr-nrt <i>output-pcr output-scr output-mbs</i> <i>[input-pcr] [input-scr] [input-mbs]</i></p> <p>Example: Router(config-if)# vbr-nrt 10000 5000 32 20000 10000 64</p>	<p>(Optional) Configures the variable bit rate-nonreal time (VBR-NRT) QoS and specifies the output peak cell rate (PCR), output sustainable cell rate (SCR), and output maximum burst cell size (MBS) for an ATM PVC, PVC range, switched virtual circuit (SVC), virtual circuit (VC) class, or VC bundle member.</p> <ul style="list-style-type: none"> Enter the output PCR, SCR, and MBS. <p>Note This step applies only to ATM PVCs. It is not required for interfaces or subinterfaces.</p>
Step 6	<p>vbr-rt <i>peak-rate average-rate burst</i></p> <p>Example: Router(config-if)# vbr-rt 640 56 80</p>	<p>(Optional) Configures the real-time VBR for Voice over ATM connections.</p> <ul style="list-style-type: none"> Enter the peak information rate (PIR), the average information rate (AIR), and the burst size. <p>Note This step applies only to ATM PVCs. It is not required for interfaces or subinterfaces.</p>
Step 7	<p>cbr <i>rate</i></p> <p>Example: Router(config-if-atm-vc)# cbr 56</p>	<p>(Optional) Configures the CBR for the ATM circuit emulation service (CES) for an ATM PVC.</p> <p>This command can be used in different modes, including ATM-VC configuration mode (for ATM PVCs and SVCs), ATM PVC range configuration mode (for an ATM PVC range), or ATM PVC-in-range configuration mode (for an individual PVC within a PVC range).</p> <ul style="list-style-type: none"> Enter the CBR. <p>Note This step applies only to ATM PVCs. It is not required for interfaces or subinterfaces.</p>

	Command or Action	Purpose
Step 8	<p>pvc [<i>name</i>] <i>vpi/vci</i> [<i>ces</i> <i>ilmi</i> <i>qsaal</i> <i>smds</i>]</p> <p>Example: Router(config-if)# pvc 1/32</p>	<p>(Optional) Creates or assigns a name to an ATM PVC and specifies the encapsulation type on an ATM PVC.</p> <ul style="list-style-type: none"> Enter the ATM network virtual path identifier (VPI) and the ATM network virtual channel identifier (VCI) for the ATM PVC. <p>Note This step applies only to ATM PVCs. It is not required for interfaces or subinterfaces.</p>
Step 9	<p>ip address <i>ip-address mask</i> [<i>secondary</i>]</p> <p>Example: Router(config-if)# ip address 10.10.100.1 255.255.255.0</p>	<p>(Optional) Sets a primary or secondary IP address for an interface.</p> <p>Note Applies only to low-speed interfaces (that is, interfaces with link speeds of 768 Kbps or lower.)</p>
Step 10	<p>frame-relay interface-dlci <i>dlci</i> [<i>ietf</i> <i>cisco</i>] [<i>voice-cir cir</i>] [<i>ppp virtual-template-name</i>]</p> <p>Example: Router(config-if)# frame-relay interface-dlci 100</p>	<p>(Optional) Assigns a DLCI to a specified Frame Relay subinterface on the router or access server, or assigns a specific PVC to a DLCI, or applies a virtual template configuration for a PPP session.</p> <ul style="list-style-type: none"> Enter the DLCI number. <p>Note This step applies only to Frame Relay interfaces (either low-speed or high-speed).</p>
Step 11	<p>auto discovery qos [<i>trust</i>]</p> <p>Example: Router(config-if)# auto discovery qos</p>	<ul style="list-style-type: none"> Configures the data discovery phase of the AutoQoS for the Enterprise feature. <p>Note The optional trust keyword indicates that the DSCP markings of the packet are trust (that is, relied on) for classification of the voice, video, and data traffic. For more information, see the “Trusted Boundary” section on page 974.</p>
Step 12	<p>exit</p> <p>Example: Router(config-if)# exit</p>	<p>(Optional) Returns to interface configuration mode.</p>

What to Do Next

Use the **auto qos** command to generate and install the AutoQoS templates. These templates are generated on the basis of the data collected in the Auto-Discovery phase, and will be used to create and install the corresponding class maps and policy maps.

Enabling the AutoQoS Template Generation and Installation Phase

This phase generates templates on the basis of the data collected during the Auto-Discovery phase and then installs the templates on the interface. These templates are then used to create class maps and policy maps for use on your network. After they are created, the class maps and policy maps are also installed on the interface.

To enable the AutoQoS template generation and installation phase, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **bandwidth** *kilobits*
5. **vbr-nrt** *output-pcr output-scr output-mbs [input-pcr] [input-scr] [input-mbs]*
6. **vbr-rt** *peak-rate average-rate burst*
7. **cbr rate**
8. **pvc** [*name*] *vpi/vci [ces | ilmi | qsaal | smds]*
9. **ip address** *ip-address mask [secondary]*
10. **frame-relay interface-dlci** *dlci [ietf | cisco] [voice-cir cir] [ppp virtual-template-name]*
11. **auto qos**
12. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface s4/0	Configures an interface (or subinterface) type and enters interface configuration mode. <ul style="list-style-type: none">Enter the interface type number.
Step 4	bandwidth <i>kilobits</i> Example: Router(config-if)# bandwidth 1540	(Optional) Sets a bandwidth value for an interface. <ul style="list-style-type: none">Enter the bandwidth value in Kbps. Note This step applies only to interfaces and subinterfaces. It is not required for ATM PVCs.
Step 5	vbr-nrt <i>output-pcr output-scr output-mbs [input-pcr] [input-scr] [input-mbs]</i> Example: Router(config-if)# vbr-nrt 10000 5000 32 20000 10000 64	(Optional) Configures the VBR-NRT and specifies the output PCR, output SCR, and output MBS for an ATM PVC, PVC range, SVC, VC class, or VC bundle member. <ul style="list-style-type: none">Enter the output PCR, SCR, and MBS. Note This step applies only to ATM PVCs. It is not required for interfaces or subinterfaces.

	Command or Action	Purpose
Step 6	<p>vbr-rt <i>peak-rate average-rate burst</i></p> <p>Example: Router(config-if)# vbr-rt 640 56 80</p>	<p>(Optional) Configures the real-time VBR for Voice over ATM connections.</p> <ul style="list-style-type: none"> Enter the PIR, the AIR, and the burst size. <p>Note This step applies only to ATM PVCs. It is not required for interfaces or subinterfaces.</p>
Step 7	<p>cbr <i>rate</i></p> <p>Example: Router(config-if-atm-vc)# cbr 56</p>	<p>(Optional) Configures the CBR for the ATM CES for an ATM PVC.</p> <p>This command can be used in different modes, including ATM-VC configuration mode (for ATM PVCs and SVCs), ATM PVC range configuration mode (for an ATM PVC range), or ATM PVC-in-range configuration mode (for an individual PVC within a PVC range).</p> <ul style="list-style-type: none"> Enter the CBR. <p>Note This step applies only to ATM PVCs. It is not required for interfaces or subinterfaces.</p>
Step 8	<p>pvc [<i>name</i>] <i>vpi/vci</i> [ces ilmi qsaal smds]</p> <p>Example: Router(config-if)# pvc 1/32</p>	<p>(Optional) Creates or assigns a name to an ATM PVC and specifies the encapsulation type on an ATM PVC.</p> <ul style="list-style-type: none"> Enter the ATM network VPI and the ATM network VCI for the ATM PVC. <p>Note This step applies only to ATM PVCs. It is not required for interfaces or subinterfaces.</p>
Step 9	<p>ip address <i>ip-address mask</i> [secondary]</p> <p>Example: Router(config-if)# ip address 10.10.100.1 255.255.255.0</p>	<p>(Optional) Sets a primary or secondary IP address for an interface.</p> <p>Note Applies only to low-speed interfaces (that is, interfaces with link speeds of 768 Kbps or lower.)</p>
Step 10	<p>frame-relay interface-dlci <i>dlci</i> [ietf cisco] [voice-cir <i>cir</i>] [ppp <i>virtual-template-name</i>]</p> <p>Example: Router(config-if)# frame-relay interface-dlci 100</p>	<p>(Optional) Assigns a DLCI to a specified Frame Relay subinterface on the router or access server, or assigns a specific PVC to a DLCI, or applies a virtual template configuration for a PPP session.</p> <ul style="list-style-type: none"> Enter the DLCI number. <p>Note This step applies only to Frame Relay interfaces (either low-speed or high-speed).</p>
Step 11	<p>auto qos</p> <p>Example: Router(config-if)# auto qos</p>	<ul style="list-style-type: none"> Configures the Auto-Discovery (data discovery) phase of the AutoQoS for the Enterprise feature.
Step 12	<p>exit</p> <p>Example: Router(config-if)# exit</p>	<p>(Optional) Returns to interface configuration mode.</p>

FAQs and Troubleshooting Tips

Below are answers to frequently asked questions (FAQs) and tips for troubleshooting situations that you may encounter when configuring or using the AutoQoS for the Enterprise feature.

Why can't I configure the AutoQoS for the Enterprise feature?

- To configure the feature, CEF must be enabled. Verify that CEF is enabled on your network.

Why isn't the AutoQoS for the Enterprise feature supported on my router?

- The AutoQoS for the Enterprise feature is supported only on the IP Plus image for low-end platforms. Verify that you have the IP Plus image installed on your router.

Why are some of my QoS configurations still present after I disable the AutoQoS for the Enterprise feature?

- You have to manually disable any QoS configurations that were modified by the AutoQoS for the Enterprise feature.

Why did my low-speed network link go down when I enabled the AutoQoS for the Enterprise feature?

- Ensure that the AutoQoS for the Enterprise feature is enabled on *both* sides of the network link.

Why can't I establish an end-to-end connection on the Frame Relay link?

- Check the bandwidth on both sides of the Frame Relay link. The bandwidth on both sides of the link *must be the same*; otherwise a fragmentation size mismatch occurs, and a connection cannot be established.

**Note**

For more help, see the [“Technical Assistance”](#) section.

What to Do Next

If the policy maps and class maps created (on the basis of the templates generated by the AutoQoS for the Enterprise feature) do not meet the needs of your network, the policy maps and class maps can be modified using the appropriate Cisco IOS commands.

**Note**

Although you can modify the policy maps and class maps, they may not be removed properly when the AutoQoS for the Enterprise feature is disabled using the **no auto qos** command. You may have to manually remove any modified policy maps and class maps. For more information about the **no auto qos** command, see the [“Command Reference”](#) section.

Verifying the Configuration

The AutoQoS template generation phase of the AutoQoS for the Enterprise feature automatically generates templates that are, in turn, used to create policy maps and class maps. These policy maps and class maps configure the QoS features on your network.

To verify the configuration (that is, the policy maps and class maps), perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show auto qos [interface [interface type]]**
and/or
3. **show auto discovery qos [interface [interface type]]**
and/or
4. **show policy-map interface [interface type]**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	show auto qos [interface [interface type]] Example: Router# show auto qos interface s4/0 and/or	(Optional) Displays the AutoQoS templates created for a specific interface or all interfaces.
Step 3	show auto discovery qos [interface [interface type]] Example: Router# show auto discovery qos interface s4/0 and/or	(Optional) Displays the results of the data collected during the Auto-Discovery phase for a specific interface or all interfaces.
Step 4	show policy-map interface [interface type] Example: Router# show policy-map interface s4/0	(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. <ul style="list-style-type: none"> • The packet statistics can be displayed for a specific interface, subinterface, PVC, or all interfaces, subinterfaces, or PVCs.
Step 5	exit Example: Router# exit	(Optional) Exits privileged EXEC mode.

Configuration Examples for the AutoQoS for the Enterprise Feature

This section provides the following configuration examples:

- [Enabling the Auto-Discovery Phase: Example, page 983](#)
- [Enabling the AutoQoS Template Generation Phase: Example, page 983](#)
- [Verifying the AutoQoS for the Enterprise Configuration: Example, page 983](#)

Enabling the Auto-Discovery Phase: Example

In the following example, the Auto-Discovery phase of the AutoQoS for the Enterprise feature has been enabled on serial interface s4/0 by using the **auto discovery qos** command. In this example, the bandwidth has been specified, although this is optional. With this configuration, data about the network traffic will be collected using NBAR-based protocol discovery and the traffic on the network will be analyzed.

```
Router> enable
Router# configure terminal
Router(config)# interface s4/0
Router(config-if)# bandwidth 1540
Router(config-if)# auto discovery qos
Router(config-if)# exit
```

Enabling the AutoQoS Template Generation Phase: Example

In the following example, the template generation phase of the AutoQoS for the Enterprise feature has been enabled on serial interface s4/0 by using the **auto qos** command. In the template generation phase, class maps and policy maps are created (and installed) on the basis of the information collected during the Auto-Discovery phase conducted earlier.

```
Router> enable
Router# configure terminal
Router(config)# interface s4/0
Router(config-if)# auto qos
Router(config-if)# exit
```

Verifying the AutoQoS for the Enterprise Configuration: Example

The AutoQoS template generation phase of the AutoQoS for the Enterprise feature automatically generates templates that are, in turn, used to create policy maps and class maps. These policy maps and class maps configure the QoS features on your network.

The output of the **show auto discovery qos** command, the **show auto qos** command, and the **show policy-map interface** command can be used to verify the contents of the policy maps and class maps created by this AutoQoS for the Enterprise feature. The following section contains sample output for each of these commands.

The following is sample output from the **show auto discovery qos** command. This example displays the data collected during the Auto-Discovery (data discovery) phase.

Router# **show auto discovery qos**

Serial2/1.1

AutoQoS Discovery enabled for applications

Discovery up time: 55 minutes, 52 seconds

AutoQoS Class information:

Class VoIP:

Recommended Minimum Bandwidth: 517 Kbps/50% (PeakRate).

Detected applications and data:

Application/ Protocol	AverageRate (kbps/%)	PeakRate (kbps/%)	Total (bytes)
-----	-----	-----	-----
rtp audio	2/<1	517/50	703104

Class Interactive Video:

Recommended Minimum Bandwidth: 24 Kbps/2% (AverageRate).

Detected applications and data:

Application/ Protocol	AverageRate (kbps/%)	PeakRate (kbps/%)	Total (bytes)
-----	-----	-----	-----
rtp video	24/2	5337/52	704574

Class Control:

Recommended Minimum Bandwidth: 0 Kbps/0% (AverageRate).

Detected applications and data:

Application/ Protocol	AverageRate (kbps/%)	PeakRate (kbps/%)	Total (bytes)
-----	-----	-----	-----
h323	0/0	74/7	30212
rtcp	0/0	7/<1	1540

Class Streaming Video:

Recommended Minimum Bandwidth: 3 Kbps/<1% (AverageRate).

Detected applications and data:

Application/ Protocol	AverageRate (kbps/%)	PeakRate (kbps/%)	Total (bytes)
-----	-----	-----	-----
cuseeme	3/<1	6148/60	99038

Class Transactional:

Recommended Minimum Bandwidth: 1 Kbps/<1% (AverageRate).

Detected applications and data:

Application/ Protocol	AverageRate (kbps/%)	PeakRate (kbps/%)	Total (bytes)
-----	-----	-----	-----
sqlnet	1/<1	1706/16	40187

Class Bulk:

Recommended Minimum Bandwidth: 0 Kbps/0% (AverageRate).

Detected applications and data:

Application/ Protocol	AverageRate (kbps/%)	PeakRate (kbps/%)	Total (bytes)
-----	-----	-----	-----
ftp	0/0	313/30	74480

Class Scavenger:

Recommended Minimum Bandwidth: 1 Kbps (AverageRate)/0% (fixed).

Detected applications and data:

Application/ Protocol	AverageRate (kbps/%)	PeakRate (kbps/%)	Total (bytes)
-----	-----	-----	-----
napster	1/<1	1429/13	33941

```

Class Management:
Recommended Minimum Bandwidth: 0 Kbps/0% (AverageRate).
Detected applications and data:
Application/           AverageRate           PeakRate              Total
Protocol              (kbps/%)             (kbps/%)             (bytes)
-----
dhcp                  0/0                  84/8                 114480
ldap                  0/0                  169/16              55364
Class Routing:
Recommended Minimum Bandwidth: 0 Kbps/0% (AverageRate).
Detected applications and data:
Application/           AverageRate           PeakRate              Total
Protocol              (kbps/%)             (kbps/%)             (bytes)
-----
icmp                  0/0                  2/<1                 300
Class Best Effort:
Current Bandwidth Estimation: 350 Kbps/34% (AverageRate).
Detected applications and data:
Application/           AverageRate           PeakRate              Total
Protocol              (kbps/%)             (kbps/%)             (bytes)
-----
unknowns             336/32              99457/97            949276
http                 14/1                15607/15            41945

```

The following is sample output from the **show auto qos** command. This example displays the templates created on the basis of the data collected during the data collection phase.

```

Router# show auto qos
!
policy-map AutoQoS-Policy-Se2/1.1
  class AutoQoS-Voice-Se2/1.1
    priority percent 50
    set dscp ef
  class AutoQoS-Inter-Video-Se2/1.1
    bandwidth remaining percent 10
    set dscp af41
  class AutoQoS-Stream-Video-Se2/1.1
    bandwidth remaining percent 1
    set dscp cs4
  class AutoQoS-Transactional-Se2/1.1
    bandwidth remaining percent 1
    set dscp af21
  class AutoQoS-Scavenger-Se2/1.1
    bandwidth remaining percent 1
    set dscp cs1
  class class-default
    fair-queue
!
policy-map AutoQoS-Policy-Se2/1.1-Parent
  class class-default
    shape average 1024000
    service-policy AutoQoS-Policy-Se2/1.1
!
class-map match-any AutoQoS-Stream-Video-Se2/1.1
  match protocol cuseeme
!
class-map match-any AutoQoS-Transactional-Se2/1.1
  match protocol sqlnet
!
class-map match-any AutoQoS-Voice-Se2/1.1
  match protocol rtp audio
!
class-map match-any AutoQoS-Scavenger-Se2/1.1
  match protocol napster

```

```

!
class-map match-any AutoQoS-Inter-Video-Se2/1.1
  match protocol rtp video
!
rmon event 33333 log trap AutoQoS description "AutoQoS SNMP traps for Voice Drops" owner
AutoQoS

Serial2/1.1: DLCI 58 -
!
interface Serial2/1.1 point-to-point
  frame-relay interface-dlci 58
    class AutoQoS-FR-Serial2/1-58
!
map-class frame-relay AutoQoS-FR-Serial2/1-58
  frame-relay cir 1024000
  frame-relay bc 10240
  frame-relay be 0
  frame-relay mincir 1024000
  service-policy output AutoQoS-Policy-Se2/1.1-Parent

```

The following sample output from the **show policy-map interface** command displays the packet statistics of the classes (for all service policies) configured by the AutoQoS for the Enterprise feature on the serial2/1/1 subinterface.

```
Router# show policy-map interface
```

```

Serial2/1.1: DLCI 58 -

Service-policy output: AutoQoS-Policy-Se2/1.1-Parent

Class-map: class-default (match-any)
 725797 packets, 224584146 bytes
 5 minute offered rate 3468000 bps, drop rate 2605000 bps
Match: any
Traffic Shaping
  Target/Average   Byte   Sustain   Excess   Interval   Increment
  Rate             Limit  bits/int  bits/int  (ms)       (bytes)
  1024000/1024000  6400   25600    25600    25         3200

Adapt Queue      Packets  Bytes    Packets  Bytes    Shaping
Active Depth
-      1000      268047   48786251 268032   48777309 yes

Service-policy : AutoQoS-Policy-Se2/1.1

Class-map: AutoQoS-Voice-Se2/1.1 (match-any)
 80596 packets, 5158144 bytes
 5 minute offered rate 105000 bps, drop rate 14000 bps
Match: protocol rtp audio
 80596 packets, 5158144 bytes
 5 minute rate 105000 bps
Queueing
  Strict Priority
  Output Queue: Conversation 72
  Bandwidth 70 (%)
  Bandwidth 716 (kbps) Burst 17900 (Bytes)
  (pkts matched/bytes matched) 82010/5248640
  (total drops/bytes drops) 12501/800064
QoS Set
  dscp ef
  Packets marked 82010

Class-map: AutoQoS-Inter-Video-Se2/1.1 (match-any)
 50669 packets, 42473594 bytes

```



```
5 minute offered rate 692000 bps, drop rate 513000 bps
Match: protocol rtp video
  50669 packets, 42473594 bytes
  5 minute rate 692000 bps
Queueing
  Output Queue: Conversation 73
  Bandwidth remaining 10 (%) Max Threshold 64 (packets)
  (pkts matched/bytes matched) 51558/43218807
(depth/total drops/no-buffer drops) 9/37454/7588
QoS Set
  dscp af41
  Packets marked 52193

Class-map: AutoQoS-Stream-Video-Se2/1.1 (match-any)
79843 packets, 30678725 bytes
5 minute offered rate 511000 bps, drop rate 428000 bps
Match: protocol cuseeme
  79843 packets, 30678725 bytes
  5 minute rate 511000 bps
Queueing
  Output Queue: Conversation 74
  Bandwidth remaining 1 (%) Max Threshold 64 (packets)
  (pkts matched/bytes matched) 82381/31658370
(depth/total drops/no-buffer drops) 0/63889/7245
QoS Set
  dscp cs4
  Packets marked 82395

Class-map: AutoQoS-Transactional-Se2/1.1 (match-any)
77805 packets, 8511468 bytes
5 minute offered rate 157000 bps, drop rate 102000 bps
Match: protocol sqlnet
  77805 packets, 8511468 bytes
  5 minute rate 157000 bps
Queueing
  Output Queue: Conversation 75
  Bandwidth remaining 1 (%) Max Threshold 64 (packets)
  (pkts matched/bytes matched) 80635/8820988
(depth/total drops/no-buffer drops) 64/50967/3296
QoS Set
  dscp af21
  Packets marked 80655

Class-map: AutoQoS-Scavenger-Se2/1.1 (match-any)
30723 packets, 7127736 bytes
5 minute offered rate 136000 bps, drop rate 84000 bps
Match: protocol napster
  30723 packets, 7127736 bytes
  5 minute rate 136000 bps
Queueing
  Output Queue: Conversation 76
  Bandwidth remaining 1 (%) Max Threshold 64 (packets)
  (pkts matched/bytes matched) 31785/7373950
(depth/total drops/no-buffer drops) 0/16381/6160
QoS Set
  dscp cs1
  Packets marked 31955
```

```

Class-map: class-default (match-any)
  406161 packets, 130634479 bytes
  5 minute offered rate 2033000 bps, drop rate 1703000 bps
  Match: any
  Queueing
    Flow Based Fair Queueing
    Maximum Number of Hashed Queues 64
  (total queued/total drops/no-buffer drops) 806/291482/13603

```

Additional References

The following sections provide references related to the AutoQoS for the Enterprise 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, Release 12.3 T
NBAR	Network-Based Application Recognition and Distributed Network-Based Application Recognition , Cisco IOS Release 12.3(4)T feature module
AutoQoS for voice over IP (VoIP)	AutoQoS — VoIP , Cisco IOS Release 12.2(15)T feature module
QoS concepts and features applicable to VoIP; high-level examples for configuring these features in different network environments	Quality of Service for Voice over IP , Cisco Integrated Networking Solutions document
LFI and cRTP	Cisco IOS Quality of Service Solutions Configuration Guide
Packet classification	Cisco IOS Quality of Service Solutions Configuration Guide
LLQ	Cisco IOS Quality of Service Solutions Configuration Guide
Service policies (policy maps)	Cisco IOS Quality of Service Solutions Configuration Guide
Frame Relay and ATM commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	Cisco IOS Wide-Area Networking Command Reference, Release 12.3 T
Frame-Relay DLCIs, ATM PVCs, Frame Relay-to-ATM Interworking, MLPoFR, and other information about Frame Relay networks	Cisco IOS Wide-Area Networking Configuration Guide
MLP	Cisco IOS Dial Technologies Configuration Guide
CEF	Cisco IOS Switching Services Configuration Guide
SNMP	Cisco IOS Configuration Fundamentals and Network Management Configuration Guide
CiscoWorks QoS Policy Manager (QPM)	Product information available online at Cisco.com

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
<ul style="list-style-type: none"> CISCO-CLASS-BASED-QOS-MIB CISCO-CLASS-BASED-QOS-CAPABILITY-MIB 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

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

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 new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at <http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm>.

New Commands

- **auto qos**
- **auto discovery qos**
- **show auto discovery qos**

Modified Commands

- **show auto qos**